

Eggplant Functional Release Notes

The notes below provide descriptions of the new features and changes introduced with each release of eggPlant. You are strongly encouraged to read about the relevant changes whenever you upgrade from an earlier version.

Potential compatibility issues are highlighted in red.

Compatibility

- **Mac OS X 10.9, 10.10, 10.11, 10.12, 10.13**
- **Linux 64-bit Debian and RedHat based on glib 2.11 and above**
- **Windows 64-bit Versions: 7, 10, Server 2008, Server 2012**

NOTE:

32-bit distributions are no longer provided

Release 18.0.1.1 (4-January-2018)

- Corrected a problem with the Save Image Panel
-

Release 18.0.1 (2-January-2018)

Bug Fixes / Tweaks:

- Fixes a problem running the Team License Server batch file
- Fixes a problem setting up the Team License Server as a service
- Launches the local license server when invoked from command-line
- Fixes a problem with the Swipe command

- Now properly lists all open suites in the Save Image Panel
 - Fixes a problem bringing up the Release Notes from the Help Menu
-

Release 18.0.0 (20-December-2017)

Highlights:

- **Selenium WebDriver Support** – Selenium WebDriver calls can now be incorporated into Eggplant scripts.
- **Roaming Licenses** – Administrators can now check out licenses to testers working remotely or traveling between offices.
- **Expanded Language Browser** – The language browser in the script editor now includes more basic elements of SenseTalk, and new topics such as Selenium WebDriver commands and functions.
- **New Sample Suites** – Added new sample suites that demonstrate SenseTalk scripts, Gherkin features, Scriptless (Table) tests, and API tests, respectively. You can open these suites from the Welcome screen. (*Help* menu → *Welcome to Eggplant*)
- **New Installer (Windows)** – The new installer offers faster installation in a smaller download.
- **Updated icons and design**

Selenium WebDriver Support:

Eggplant Functional can now interact with a Selenium Server using the WebDriver API. This requires that you have installed and started a WebDriver Server on your System-Under-Test (SUT).

- **Connections** - You must first establish a WebDriver connection to your SUT. Web Driver connections can be made ***in addition*** to your active VNC or RDP connection. You can establish these connections in three ways
 - **Connection List** - From the connection list in the GUI you can add a

new WebDriver connection by setting the Connection Type to WebDriver. You must then select a Browser and can optionally specify Desired Capabilities.

- **WebConnect Command** - You can also establish a connection by using the WebConnect command. You can specify the Host, Port, Browser and URL to establish the connection.
 - **WebConnect** (Host: "New_Server", Port: 4444, Browser:"Chrome", URL:"https://www.google.com")
 - **WebServer() Function** - You can establish a connection by using the WebDriver() function to return a WebDriverConnection object
 - **set driver to** `WebDriver(host:"Win-10", browser:"chrome", url:"https://www.google.com")`
 - **WebDisconnect** (all)
-
- **Command Interaction** - You can now use the following Eggplant Functional commands with **WebElement Identifiers** in addition to image names and OCR text.
 - **Click** ([identifierType]: "[value]")
 - **Double-Click** ([identifierType]: "[value]")
 - **Right-Click** ([identifierType]: "[value]")
 - **MoveTo** ([identifierType]: "[value]")
 - **Drag** ([identifierType]: "[value]")
 - **Drop** ([identifierType]: "[value]")
 - **WaitFor** (duration,[id]: "[value]")
-
- **Selenium commands** – You can also use the following commands to interact with your target browser through WebDriver.
 - **SendKeys** "[text]" or **SendKeys** [webElement], "[text]"
 - **WebElementFound**([identifierType]: "[value]")
 - **OpenWebDrivers**()
 - **Submit**
 - **Refresh**
 - **GoBack**

- **GoForward**
- **WebScreenshot**

- **WebElement Identifiers** - The web elements that are used in Webdriver can be identified in the following ways:
 - **webID** — locate an element whose id attribute is equal to the indicated value
 - **webName** — locate an element whose name attribute is equal to the indicated value
 - **webClassName** — locate an element whose class attribute is equal to the indicated value
 - **webTagName** — locate an element whose HTML tag is equal to the indicated value
 - **webLinkText** — locate a link whose text is equal to the indicated value
 - **webPartialLinkText** — locate a link whose text contains the indicated value
 - **webCssSelector** — locate an element using the indicated css selector expression
 - **webXPath** — locate an element using the given xPath expression

- **WebDriverConnection and WebElement Objects** - In addition to the workflow commands listed above you can interact with the browser through a **WebDriverConnection** object. This includes invoking actions and getting properties of the browser page as well as identifying **WebElement** objects on the page to interact with. You can get the active WebDriver object by using the **WebDriver()** function with no parameters. You can find **WebElements** on the active WebDriver page using the **findElement()** and **findElements()** functions.
 - **WebDriverConnection Properties**

- **Name, Host, Port, Browser** -- the connection information
- **url** — read/write: the URL displayed in the browser
- **title** — the title of current page
- **pageSource** — the full HTML source of the current page
- **capabilities** — a property list indicating the capabilities of the webDriver on this browser connection
- **cookies** — a list of property lists for each of the cookies in the browser for the current site
- **activeElement** — returns a WebElement object for the current active element on the page
- **alertText** — the text of an alert, if one is shown

- **WebDriverConnection Functions**
 - **goBack()** — go back in the browser history (the ‘back’ button action of the browser)
 - **goForward()** — go forward in the browser history (the ‘forward’ button action of the browser)
 - **refresh()** — refresh the page (the ‘refresh’ or ‘reload’ button action of the browser)
 - **quit()** — explicitly quit the browser and terminate the WebDriver session (normally done automatically by SenseTalk when the WebDriver object is no longer in use)
 - **element()** — specify an element within the current page DOM, without immediately locating it
 - **findElement()** — locate a specific element within the current page DOM
 - **findElements()** — locate all matching elements within the current page DOM

- **WebElement Properties (Read-Only)**
 - **name** -- the name attribute of the element, generally only relevant to form elements
 - **id** -- the ID of the element, or "" if none is specified

- **class** -- the full CSS classes given to the element (or "")
- **tagName** — the HTML tag name of the element
- **text** — the displayed text of the element, if any
- **location** — the x,y coordinates of the element on the page
- **size** — the width and height of the element as displayed on the page
- **isSelected** — true or false, whether the element is currently selected or not
- **isEnabled** — true or false, whether the element is enabled or disabled
- **isDisplayed** — true or false, whether the element is currently showing on the page
- **WebDriver** -- the WebDriver object of the connection this element belongs to
- **WebElement Functions** - Note: Not all actions are applicable to all element types
 - **sendKeys(text)** — simulate typing the given text into the element (does not clear any previous contents)
 - **clear()** — clear the contents of the element
 - **submit()** — submit the element, or the form that contains it
 - **click()** — perform a click on the element
 - **doubleClick()** — perform a double-click on the element
 - **moveTo()** — move to the center of the element
 - **moveTo(offset)** — move to a specific x,y offset relative to the top left corner of the element
 - **attribute(attrName)** — returns the value of the attribute named attrName of the element
 - **element()** — specify an element within the element, without immediately locating it
 - **findElement()** — locate a specific element within the element
 - **findElements()** — locate all matching elements within the element

Scripting Improvements:

- Added the following global properties to dynamically control image-search heuristics that are used by image doctor:
 - ImageDoctorSearchTypes
 - AutoDoctorSearchTypes
 - OCRDoctorSearchTypes
 - OCRAutoDoctorSearchTypes

Results Improvements:

- Added a new type for *the ScriptLogging* global property: Actions. When the scriptLogging is set to Actions, global property changes do not appear in log files. This is helpful for users who want their results to reflect the steps they are performing against the SUT, without added information from Eggplant.
- Eggplant Functional now computes results statistics for Gherkin, API and Table-based tests.

License Improvements:

- Now, when a license is removed from the license server, Eggplant honors that removal by not automatically replacing the license.

Bug Fixes / Tweaks:

- Fixes an occasional problem capturing movies.
- Fixes a problem starting Eggplant Functional on Windows 10 Fall Creators Update
- Fixes a problem using RunWithNewResults on scripts with a dot in the name
- Fixes a problem where Results of scripts with dot in the name are grouped

together

- Fixes a problem calling the legacy Text Image Generator
- Fixes a problem accessing certain System environment variables
- Fixes a problem sorting results by the Run Date
- Fixed a problem overwriting -DefaultDocumentDirectory on Windows
- Fixed a problem where would clear Global Variables between drive calls when the GUI flag was enabled
- Fixed a problem where scripts could fail to end correctly
- Removed the gpbs and gdnc dependent processes on Windows
- Autocomplete will no longer stop on an underscore
- Helper suites will no longer appear on the Save Image Sheet
- Creating collections from the Update Image panel will now retain the original search settings
- Improved the highlighting of Feature steps
- Added all heuristics to the Run Preference Panel
- Fixes several performance and stability issues

Release 17.2.2

(7-November-2017)

Bug Fixes / Tweaks:

- Fixes a memory access issue when doing OCR operations on 64-bit Windows platforms.
- Fixes a problem setting a breakpoint while scripts are running
- Fixes a problem when attempting to use the auto update on a **CharacterCollection** search
- Fixes a problem that could occur when deleting multiple images at once
- Adds Connections to the values imported and exported from the preferences
- Fixes a problem returning the API:JSON response body
- Fixes several performance and stability issues

Release 17.2.1 (24-October-2017)

API Testing Improvements:

- API-Test Assistance– Adds the following types of auto-correction in API tests
 - Adds *http://* to the beginning of the URL
 - Removes duplicate *http* from the URL
 - Removes trailing *?* character
 - Validates JSON that is being submitted
 - Verifies that a body is present on *POST*, *PUT*, and *PATCH*
 - Validates XML body (Mac OS only)

Bug Fixes / Tweaks:

- Fixes a problem that could cause script execution to fail to end
 - Improves behavior when running Eggplant across multiple monitors
 - Fixes a problem that caused the API() function to persist beyond interrupted script executions
 - Fixes a problem that could cause API tests to fail to save
 - Added support for externally produced, grayscale images to be used in suites
 - For command-line execution, now supports using the *-param* argument in any position, not just as the final argument
 - Fixes a problem that could cause generated script commands to contain unused property placeholders
 - Updates Eggplant to support Zlib VNC encodings on Mac OS 10.13.
 - Fixes several performance and stability issues
-

Release 17.2.0

(26-September-2017)

Highlights:

- **API Testing** – There is a new script type, API Tests, used for testing REST and SOAP web APIs.
- **OCR Tuner** – The OCR tuner enables you to preview OCR behavior during the scripting process.
- **Character Collections**– Character collections are sets of character-images that you can search for as strings. Character collections are useful for consistent, but low-quality text that is not suitable for OCR use.

Scripting Improvements:

- **Suite Variables**– Variables that are set in Suite Settings and predefined to be usable by any Script in the suite, include API Scripts.
- You can now create ScreenShot connections from a **Connect** command by passing the full file path for the **name** and adding **type:"screenshot"**
- Added a **CSVFormat** function and a **CSVValue** function
- The **TypeEncodedText** command now logs that it was called.
- Adds the **TextStyle** Global Property– The new global property, **the textStyle**, is shorthand for a given style of the current text platform. For example, the following command:
 - `set the textStyle.caseSensitive to yes`
- **The TextStyle** can be used to refer to any of the styles defined in Preferences, as in:
 - `set the textStyle to "Android"`

User Interface Improvements:

- **OCR Tuner**– Can be turned on from the Remote Window Toolbar or from the Capture Area context menu. When shown the OCR Tuner will display what is read in the Capture area with the shown settings.
 - You can adjust the area and change settings until you determine the

best setting of properties for your SUT.

- You can then save those properties as a **Text Style**.
- **Diagnostic Preferences**– In Preferences > Run > Diagnostics, you can now choose the search heuristics you use for the Image Update feature, both for images and text, and both for manual and automatic updates.
- **Text Preferences**– Infrequently used settings have been moved into an Advanced section. Also, there is a new preference that enables logging of all text found with OCR.
- **Resource Files**– You can now drag new resource files into the Resource list in your suite, or add a new resource by right-clicking the Resource list and choosing Add Resource.
- **Character Collections** – Character Collections provide a way to identify hard-to-read text on the SUT; for example, low-resolution text. Character Collections work best with consistent, un-anti-aliased text.
 - Choose Control->Start Character Collection Capture Session.
 - In the Viewer window, click each character that you want to include in your collection. When finished select End Session.
 - In the Sessions panel name and open your session.
 - Select each character in the list to verify what it is, and adjust the capture area if necessary.
 - When you are through, click Generate Images.
 - Use Character Collection in a Script:

```
Log readCharacters(myRect,  
                  characterCollection:"LowResText")
```

Command Line Improvements:

- When starting from the command line you can now use **-password64** VALUE to pass in a Base64 encoded version of the password. Similarly you can use **-pass_code64** to pass in a Base64 encoded version of the passcode.
- You can now create a session log when running in Drive mode by passing **-DriveOutputFile PATH**

Bug Fixes / Tweaks:

- Hotspots will now be shown when Image Highlighting is enabled.
- RDP Connections will not attempt to reconnect when another user logs that session out.
- With the latest RLM License Server, licenses are always released after an hour delay (can be lowered to 10 minutes).
- Fixes a stability problem when eggPlant suites had unresolved helper suite references.
- Fixes a problem where auto-update failed to engage for some OCR searches.
- Fixes an [NSConcreteTask terminationStatus] exception when running a shell() command.
- Fixes a problem running the TestPlant RLM server on Linux.
- Fixes a rare problem closing suites with the (X) on Windows and Linux.
- Fixes stability problems in the Variable Watcher when set to Always Display.
- Fixes problems with invalid XML in RunHistory file.
- Fixes some stability issues parsing JSON.
- Improved memory performance when running a schedule.
- Fixes some stability issues using the suite-wide search field.
- Post-run scripts will no longer be logged (as-pre 17.1.5).
- Reverses behavior when script properties are accessed directly as an object.
- Restores behavior where a call to ConnectionInfo() will return information on a failed connection.
- Fixes an error when calling SuiteInfo().path in a post execution script.
- Fixes an exception when Adding a Representation on the Image Update Panel.
- Fixes several problems with find and replace looping through the entire suite.
- Fixes a problem with Helper Suites that had a dot in the filename.

- Fixes an NSSmallFloat exception when moving a suite window.
 - Adds support for PrintScreen key over RDP connections
 - Rapid Capture Sessions now minimizes the number of screenshots that are stored on disk.
 - Improves the look of the Status List
 - Improves some visual issues on the Save Image Panel.
 - Toggling **the captureMode** in a script works correctly now.
-

Release 17.1.5 (2-August-2017)

Highlights:

- You can now use Gherkin **Tags** to run a subset of your scenarios.
- You can now use Gherkin **Hooks** to call common behaviors before and after various Gherkin actions.
- You can now run Gherkin **Features** from a master script using the **RunFeature** command.

Scripting Improvements:

- Gherkin Scenarios can have multiple tags, separated by spaces, before the Scenario keyword. When running a **Feature**, the tags to run can be specified:
 - If tags are separated by commas, it means ANY of the tags (OR behaviors).
 - If tags are separated by SPACES, then it must match ALL of the tags (AND behaviors).
 - If a tag is preceded by a tilde (~), that means NOT.
 - "@a,@b ~@c" means run scenario if (A or B) AND not C.
 - Tags can be specified in the UI using the Tags field, and also from the command line using **-param "@tag"**.

- The following Gherkin Hooks can be provided in the corresponding script (or one included with #script):
 - `to handle hook_after_feature`
 - `to handle hook_before_feature`
 - `to handle hook_before_scenario`
 - `to handle hook_after_scenario`
 - `to handle hook_before_step`
 - `to handle hook_after_step`
- The new **RunFeature** command lets you specify a feature file to run in its own results context (similar to **RunWithNewResults**). You can optionally specify tags:
 - `RunFeature "MyFeature", "@MyTag"`
- You can now control whether a Gherkin feature skips the other steps of a scenario when an error is encountered:
 - defaults write Eggplant AbortScenarioOnError YES

Bug Fixes / Tweaks:

- Fixes a stability problem when working with large Microsoft Excel files.
- Fixes a problem where Excel files and databases were not closed when a script ended.
- Fixes a problem where edge detection engaged when users tried to click on a highlight rectangle from the Image Update Panel.
- You can now double-click on the Gherkin status line in the results to jump to the corresponding step declaration in the feature file.
- You can now double-click on the Gherkin status line in the results to jump to the corresponding step declaration in the feature file.
- Fixes a problem when working with suite assets found using the suite-wide search feature.
- Fixes problems where new collection folders weren't shown in the Save Image panel.

- Using Copy in capture mode stores coordinates for the current capture area and no longer takes you out of capture mode.
 - Fixes a problem where the Windows Uninstaller wouldn't terminate an active runscript. This could cause errors when doing a fresh install.
 - You can now disable bounding box edge detection:
 - defaults write Eggplant DisableDefaultBoundingBox YES
 - Fixes a crash on Mac that could occur when the Connection window was resized.
 - Fixes a problem processing certain XML files.
 - Fixes a problem with the **Mouse Location** item on the **Remote Window** toolbar.
 - Fixes a problem that could occur sending **InstallApp** to a mobile gateway.
 - Fixes a crash that could occur when allocating a new window on Windows.
-

Release 17.1.4 (30-June-2017)

Highlights:

- **LiveDrive** allows you to bring up eggPlant in Drive mode with the GUI available. This is particularly useful when debugging code sent from external agents. To enable this, use the argument **-runGui YES**
- You can now retrieve the **ResponseHeaders** of a URL:

```
put the responseHeaders of url "https://www.google.com"
```

Bug Fixes / Tweaks:

- Fixes a problem using **ODBC** on Linux platforms.
- Fixes a hang that could occur when a collection with many images was brought up in the **Image Update** panel.
- Fixes crashes that could occur in the **Image Update** panel.
- Adds the ability to reveal a Resource in the File System.
- Calls to retrieve **URL** values with a common **content-type** header will now

convert the returned data to **the defaultStringEncoding**.

- Prevents adding a full suite under the Image folder.
 - Fixes a problem importing images with capitalized extensions.
 - Fixes a problem where simultaneously starting multiple eggPlant instances on Windows could fail.
 - Fixes a problem with **shell()** commands on Windows that redirected output to a file
 - Fixes a problem using certain images for **Screenshot Connections**.
 - Fixes a problem where underscores '_' were not used with the **ValidCharacters:"*"** expression.
 - Fixes a problem sorting the the Connection List.
-

Release 17.1.3 (20-May-2017)

Bug Fixes / Tweaks:

- Fixed a crash that could occur after the Image Update Panel came up.
 - Fixed a problem where Execution licenses could prevent finding GUI licenses on 64-bit versions of eggPlant.
 - Added the ability to reveal a Resource in the File System
 - Fixed a compatibility problems with node expressions used as a container.
 - Fixed a problem where disconnecting from the SUT during a Turbo Capture Session could leave eggPlant in a confused state.
 - Fixed problems sending actions that used the Live Mode toggle key
 - Fixed problems setting Shift as the Live Mode toggle key
 - Fixed a problem where making changes on the Settings page of a suite could cause a hang.
-

Release 17.1.2 (22-April-2017)

Bug Fixes / Tweaks:

- Fixed a problem with the Cursor Location toolbar item
 - Removed some stray logging when **edge detection** is triggered
-

Release 17.1.1 **(20-April-2017)**

Highlights:

User Interface Improvements:

- **Edge Detection** is now the default behavior when capturing in the remote window. You can **SHIFT-click** (or drag) to move the existing capture area instead.
-

Connection Improvements:

- We have added support for TLS 1.2 in **RDP** connections.
-

Scripting Improvements:

- Extended Excel support up to 16,384 columns.
 - We have added the ability for Auto Update to trigger on "ImageFound()" calls.
 - **set the VerifyImageFound to "Always"**
-

Bug Fixes / Tweaks:

- Fixed a hang adding images when ADB was set to not launch on startup
 - Fixed a problem where Image Update Suggestions could appear blank on Windows
 - Added libcrypt library support for Fedora 21
 - Fixed an exception ending a script that started (but didn't stop) a movie
 - Fixed a problem with the first click in a Turbo Capture Session
 - Fixed an exception when using the color panel during OCR commands during a Turbo Session
 - Fixed a problem triggering OCR updates during Auto Update
 - Fixed a crash closing an image tab while search is still ongoing
 - Added extended tooltip information for Scripts in the side panel
 - Added support for **ApplicationKey** over RDP on 64-bit
 - Fixed a problem with Image Tolerance Sliders not updating the image as they moved
-

Release 17.1.0 **(30-March-2017)**

Highlights:

- **Excel Support** - eggPlant can now read (and write) values from XLSX files as part of your script. See the scripting section for more detail.
- **Image Update Remedies** - Quickly pick from eggPlant's suggested remedies with unique replacement images or attribute changes to make your scripts more robust.
- **Rapid Image Capture** - Using the Session feature you can quickly click on a number of interface elements you want to capture in rapid succession. The new edge detection mechanism will automatically hone in on most images.

- **Multi-Image Update Panel** - When searching using an image collection, all of the images are presented in the Image Update Panel.
-

User Interface Improvements:

- When in capture mode you can use **Edge Detection** to automatically size your capture region by **SHIFT-clicking** on the remote viewer screen.
- The **Image Update Panel** now shows recommended remedies. You can select one of the suggested remedies to capture a new image or alter the image properties.
- From the Control menu or Viewer Window Toolbar you can now start a **Rapid Image Session** - In this capture mode you can click on each interface element that you want to capture, when you are finished choose **End Session** to be taken to the Suite's Session manager. All of your locations will have suggested capture regions. You can adjust, name and then generate your new set of images.
- **Turbo Capture Sessions** have also been improved. Image borders will be detected using edge detection. When text is found at or near the image you can choose to insert an OCR command instead of capturing an image. You can also choose to use an existing matching image instead of capturing a new one.
- The **Image Update Panel** now will search all images in a collection or image search command. These images are listed at the top of the panel. You can select a particular image to diagnose or adjust properties.
- The **OCR Update Panel** now performs a standard search to catch timing errors. This panel will also shows recommended remedies that will match

text on the screen. You can use those settings temporarily or copy and apply them to your script.

Connection Improvements:

- We have now added support for **Tight** encoding in UltraVNC (VNC Preferences). This mode is a lossy compression that is helpful when using **Live mode** over a low latency connection. NOTE: It should NOT be used when capturing images or running tests.
 - We have added direct support for **SSH connections** on Windows. You will need to set your private key in your users SSH directory.
 - You can now use a screenshot as a **non-interactive SUT**. This is useful for capturing images after the fact or doing image comparisons.
 - We support **RDP** connections that have no password set.
-

Scripting Improvements:

- We have added two commands to support **encrypted** text in script windows. The **EncodeText()** function takes a string (and optional secret key) that will produce an encoded string that can be used with the **TypeEncodedText** command (which takes an encoded string and optional key) to type a string on the SUT. If you don't provide a key the Suite will generate (and store) one for you automatically. You can also generate an encoded string using the "Insert..." menu in the script editor.
- Access to **xlsx** (Excel) files can be done via function access or via database commands. The function access approach uses these 3 new

SenseTalk functions:

The **Workbook()** function is the starting point for access to an Excel spreadsheet. It takes one parameter which is the pathname to an Excel file. The **Workbook** function returns a workbook reference which in turn can be used with the **Worksheet()** or **Cell()** functions to access the file's contents, or obtain other information about the workbook.

```
set caseBook to Workbook("TestCases.xlsx")
```

The **Worksheet()** function gives access to a particular worksheet within a workbook. It can be called in two different ways, either passing two parameters (a workbook reference and a sheet identifier), or calling it as a function of the workbook with one additional parameter (the sheet identifier). The sheet identifier may be either the name of the sheet, or its number (1 for the first sheet in the workbook, 2 for the second, etc.).

```
set custSheet to Worksheet(caseBook, "Customers")
set custSheet to caseBook.Worksheet("Customers")
put caseBook's worksheet(2) into custSheet
```

The **Cell()** function is used to access an individual cell of a worksheet. It can be called by passing two parameters (a worksheet or workbook reference, and a cell identifier), or by calling it as a function of the worksheet or workbook with the cell identifier as its parameter. The cell identifier may be a traditional cell name, such as "C6", or a pair of numbers identifying the row and column, such as (6,3). The value returned by the **Cell()** function may be a string, number, boolean, or date, depending on what is stored in the spreadsheet cell.

```
put cell(custSheet, "D2") into balanceDue
put custSheet.cell(4,7) into workingBalance
```

The **Cell()** function can also be used to store a value into a cell of the worksheet, or if a reference to the Cell() return value is assigned to a variable, that variable acts like a cell reference, so if you store another value in that variable, that value will be written to the cell in the file.

put the date into cell(mySheet, 3,5) — stores the date into row 3, column 5
add 1 to custSheet's cell("E3") — this will update cell E3 of custSheet

The database approach uses SenseTalk's database capabilities. The database identifier property list must have "type" and "file" properties, like this:
(type:"excel", file:"myWorkbook.xlsx").

Then each **sheet** in the workbook can be treated as a table, accessed by sheet name. The first row of each **sheet** will be used as the column names and each row after that will be considered a record. For more information review our documentation on Tables: <http://docs.testplant.com/ePF/SenseTalk/stk-working-with-tables.htm>

Bug Fixes / Tweaks:

- The suite sections now have icons that correspond to their toggle buttons.
- You can now adjust the hot-spots when specifying images in a **Capture Session**.
- Searching a suite now also highlights matches in **Tables**
- Fixed a crash that could occur when editing scripts while a **Schedule** was running.
- Fixed an undo exception when using backspace
- **IgnoreSpaces** OCR diagnostic now properly ignores spaces in the string to find as well as the text read on screen.
- Added additional support libraries to support newer versions of **LINUX**.
- Fixed an exception that could happen when doing certain search terms.
- Scripts and breakpoints will now resolve resources when running a script using it's full path.
- Fixed a problem where you couldn't open a script from the left side if it was already selected.
- Fixed a problem where changes to the script name weren't shown.

- Prevent changing a script name to be blank.
 - Fixed a problem in pollDirectories: that would then prevent updates to suite resources.
 - Fixed a crash clicking **List Branches** in **SCM Checkout**.
 - Prevent a situation where the UI was unable to expand table columns.
 - You can now create a new **Image Folder** at the currently selected folder in the Image list.
 - Fixed a problem with **Windows Installer** that would start installing before the prior version finished uninstalling.
 - Fixed a hang when exporting **Preferences**.
 - Added the ability to see tree values in the **Variable Watcher**.
 - Clicking **Done** on the License Panel will now save the license key.
 - Fixed a problem setting file attributes on **Windows**.
-

Release 17.0.3 (11-January-2017)

Bug Fixes / Tweaks:

- Assets in the left side menu will now be shown properly for helpers suites that are referred to using a relative path.
 - Fixed a crashing problem that could happen with find and replace and bringing up assets from the left side menu.
 - Fixed a problem looping around during a find and replace.
 - Fixed a problem running scripts that were referenced using a user home path.
 - Fixed a problem renaming assets on Windows and Linux.
 - Fixed a problem where some new tabs would be brought up on the right side, beyond the viewable edge of the window.
-

Release 17.0.2 (18-December-2016)

Bug Fixes / Tweaks:

- Fixed a problem where running from the CLI on Windows would get progressively slower.
 - Fixed a problem where running from the CLI on Windows could use incorrect user preferences.
 - Fixed a problem with AltKey when connecting via RDP
 - Fixes to Next/Prev behavior from the suite wide search field
 - Restored the ability to edit Helper Suite paths. You can no longer attempt to delete "Helped By" Helpers.
 - Helper suites referenced with relative paths should now be visible in the left-side menus.
-

Release 17.0.1 (12-December-2016)***Bug Fixes / Tweaks:***

- Fixed a problem with WindowsKey, Keypad keys when connecting via RDP
 - Fixed a problem where image file extensions were added to scripts
 - Fixed a problem with Omega13 calling AttemptRecovery
 - Fixed problems establishing RDP connections using Linux
 - Fixed a problem on linux reporting NSMallocException
 - Fixes to Next/Prev behavior from the suite wide search field
 - Fixes a problem locating helper suites and recording results with some Windows paths
 - Fixed a problem running with old +Mobile keys.
 - Now the save image panel will reload when adding a helper.
 - Normalized clipboard behavior with RDP to exclude null terminating character.
 - Fixed a problem uninstalling 16.1.2 properly in some cases.
-

Release 17.0.0

(28-November-2016)

Highlights:

- **Movie Recording** is now available from the Control Menu or Remote Window Toolbar. You can also start and stop movie recording during your script with **StartMovie** and **StopMovie**.
- **Integrated Helper Resources:** You can view and use the scripts and images from your helper suites in any of the suites they are included in.
- **Suite Find and Replace:** You can find, and optionally replace, all instances of a string throughout your suite.
- **ShouldRepositionMouse:** The default for this setting has been changed from YES to NO. It is not necessary for most VNC servers because the cursor is sent separately from the screen image image. It **is** required when using an IP-KVM. It can be re-enabled in the **Preferences > Run** panel or from a script.
- The **Text** search type is now called **Smooth**: To differentiate the Text search type from OCR, we have renamed it "smooth". This search type is also useful for things besides Text. Existing images will continue to use the same search type, and the label "text" as a search type will still be honored if used in a script. ImageInfo(), however, will now report SearchType:"Smooth" instead of SearchType:"Text".
- **64-bit support:** eggPlant is now a 64-bit application, which improves performance and memory capacity.

Connection Improvements:

- **RDP enhancements:** RDP should work across a wider variety of security settings and with increased performance in most situations.
 - **PinchIn/PinchOut:** We have added a toolbar to the remote screen window that lets you manually perform a PinchIn or PinchOut.
 - **Settings for eggCloud connections:** We have added the ability to control default settings for connections made through eggCloud to the eggCloud preference panel. You can set the value for ColorDepth, ScaleRemoteScreen, and BlendScaledScreen.
-

User Interface Improvements:

- **Integrated Helper Resources:** You can view and use the scripts and images from your helper suites in any of the suites they are included in. This can be enabled using the context menu on the suite's Script and Image lists, respectively.
- **Suite Find and Replace:** You can find, and optionally replace, all instances of a string throughout your suite.
- **Scripts** can now be stored in subfolders inside a suite. Please be aware that the subfolder is part of the script's "name", and that subfolders are not automatically searched. So Script1 in Folder1 must be called using "run Folder1/Script1"
- **Result Images** that come from InitialSuites/FinalSuites and Helper Suites will be shown visually in the Results pane when reviewing logs.
- **New Helper Suite** menu item and button in the Settings tab allow you to create a new helper that is automatically included by the currently selected

suite.

- Reopening **Open Suite tabs** from a previous session can now be disabled on the General Preferences panel.
 - A new **-open** command-line argument. You can open a particular suites from the command line using **-open *filepath***. If other files were left open the last time eggPlant was closed then the suite (or script) specified on the command line is opened *instead* of whatever was previously opened.
-

Scripting Improvements:

- **Improved multi-line text finding:** A new optional attribute, `IgnoreNewlines`, when set to YES, matches a string even if it's broken over several lines.
 - **SetDeviceLocation** function added for Android SUTs. You can use the Android location simulation feature to tell the device it's at a particular location. You can either specify two number parameters (latitude and longitude) or you can provide a dictionary with any subset of these attributes: (latitude, longitude, altitude, bearing, speed, accuracy, time).
 - Allow **Which** and **Whose** in Each expressions. These simplify some Where clauses in Each expressions and make them read more nicely. **Which** is essentially "where each" and **Whose** is essentially "where each's".
-

Bug Fixes / Tweaks:

- Improved startup time on launching Windows version of eggPlant

- Improved Git compatibility to work with more Git tools (GitLab, etc.)
 - Added the ability to indicate a suite to open when starting from the command line.
 - Fixed a bug that could cause `remoteClipboard()` to fail when using RDP on a command-line script execution.
 - Fixed a bug that could cause the Windows key to erase selected text.
 - Fixed a bug that could cause eggPlant to crash when editing connection information during Connection Doctor use.
 - Fixed a problem where `CaptureScreen` was not honoring given rectangles.
 - Fixed a bug that could cause multiline text to be reported as being in the wrong location.
 - Increased the variety of white space characters ignored by the `ignoreSpaces` text (OCR) property.
 - Changed the SCM “Revert” button to “Discard changes” and fixed a bug that could cause the operation to fail on Windows operating systems.
 - Changed the Bonjour Discovery preference to be Off by default.
 - Fixed an uncommon crash associated with images that were not 32 bits per pixel.
 - Fixed a bug that caused RGB values provided for the `ContrastColor` text property to be ignored.
 - Fixed a bug where eggPlant would not look in the `/Library/Eggplant` directory for shared RLM license files.
-

Release v16.1.1

(22-July-2016)

Highlights:

- Improved script editing performance
- Choose a script to run before/after running a selection
- Improved launcher support on Windows
- **TeamLicenseServer service (Windows):** We have modified how the TeamLicenseServer is registered. If you previously set up the TeamLicenseServer as a service, please remove it and re-register it by using `InstallTeamLicenseService.bat`.

Scripting Improvements

- Improved multi-line text finding. A new optional attribute, **IgnoreNewlines**, when set to **YES**, matches a string even if it's broken over several lines (together).

User Interface Improvements:

- **Script to run before/after** a Run Selection is now available on the **Settings** tab of a suite. These options will be run when executing a selection of a SenseTalk script, Gherkin feature, or Table.
- Added source control management (SCM) information to the **Account Preferences** tab. These options let you enable/disable SCM, specify your user credentials, and specify which SCM program you would like to use.

Bug Fixes / Tweaks

- On Windows, the eggPlant.exe launcher has been removed. eggPlant Functional is now launched directly by calling the executable inside the Eggplant.app folder. This method avoids problems with antivirus software and makes it easier to pin eggPlant Functional to the Taskbar.
- Fixed a performance issue when naming images in a suite with many images.
- Upgrade to latest Android server, fixes a problem when switching the Android keyboard.
- Numerous fixes to support different **SCM** environments.
- Improved behavior when first connecting to a system under test (SUT) that is larger than your current screen.
- Fixed an exception when typing accented characters into the **Feature Editor** or **Script Editor**.
- Fixed a problem using **ValidWords** in conjunction with **IgnoreSpaces** during a text find.
- Fixed a problem with tab characters in the source string when **IgnoreSpaces** was enabled.
- Fixed occasional problems after capturing a new image.
- Fixed a problem where you could no longer abort a script.
- Fixed a problem where syntax errors in deeply nested property lists would cause very slow performance.
- The **Control > Launch App** menu will now be updated when new applications

are installed on a connected iOS device.

Release v16.10

(05-July-2016)

Highlights:

- Gherkin (.feature) Tests
 - Source Control Management (SCM) Integration
 - Connection Doctor
 - Enhanced Floating License Capabilities
-

Scripting Improvements

- **WaitFor** now returns the time spent waiting for an image to be found. You can access this value by immediately getting **the result**:

```
WaitFor 8.0, "myImage"  
put the result
```

- The new **appList()** function returns a list of apps on an iOS or Android SUT.
- **Command Line parameters to scripts:** Starting with eggPlant Functional version 16.00, you can pass individual parameters to individual scripts by using `-param myParameter`. The previous behavior, where all scripts received every argument after `-params`, is deprecated and its use discouraged.

```
runscript Script1.script -param "abc" Script2.script -  
param "xyz"
```

- **Mac OS X: Changed the default behavior for XPath expressions:** Prior to version 16.00, eggPlant Functional on Mac used nonstandard behavior when interpreting an XPath expression that did not begin with a slash ("/"). Starting with version 16.00, it now uses the correct, cross-platform behavior. You can re-enabled the old behavior on Mac with this script command:

```
set the StandardNodeExpressions to false
```

User Interface Improvements:

- **Gherkin (.feature) Tests** are a new section in the eggPlant Functional Suite window. Using standard Gherkin .feature files, you can create features, scenarios, tables, and step definitions that will be invoked in your SenseTalk scripts. Please see the [eggPlant Functional Documentation](#) for full details.
- **Source Control Management (SCM)** is now directly integrated with eggPlant Functional. Currently eggPlant Functional supports Apache Subversion (SVN) and Git (git). For suites that are under SCM, you will now get a status indicator and can Checkout, Update, Commit, Branch (Git), or Merge (Git) your files. For full details, please see the eggPlant Functional Documentation.
- Add **parameters** to the scripts called from a **schedule**.
- A new **Set Rectangle** control is available for the Viewer window toolbar. First, define the rectangle you want to identify using the capture area. Then, use this toolbar item to compute that rectangle using one (or two) images or a set of coordinates. You can use the new rectangle to limit **the search rectangle**, **readText**, or do a **ScreenCapture**.
- In addition to User **snippets**, custom **snippets** can now be stored in the suite.
- The **Show Resource** feature now works for handlers within a script.
- **Show Resource** can now also take you to quick help for the selected command or function.

Connection Improvements:

- New embedded **Android** server (Android Gateway 3.2) allows remoteClipboard access as well as a full Unicode keyboard.
 - **iOS** clipboard support, including full Unicode (requires iOS Gateway 3.2).
 - New GUI panel to install and uninstall mobile apps to your Android or iOS SUT.
 - Added common mobile buttons (**Back**, **Apps**, **Search**, **Power**, and **Volume**) to the Viewer window toolbar.
 - The new **Connection Doctor** menu helps to diagnose the most common connection problems.
-

Bug Fixes / Tweaks:

- Many memory and speed optimizations for all platforms but with particular benefit on Windows.
 - Improved the layout and adjustment of vertical toolbars in the Viewer window.
 - Cleanup to reduce inadvertently triggering virus warnings of the eggPlant application on Windows.
 - Fixed a crash that could occur when a SUT connection was lost.
 - Fixed a problem on Windows when editing text larger than the length of its display field.
 - Fixed a start-up problem with VNC detection in some Windows environments.
 - Fixed a problem with the **Shell** command not returning when called from **runscript**.
 - Fixed a problem using OCR functions when running eggPlant Functional from Amazon Web Services (AWS).
 - Fixed a problem with **Swipe** commands that specified a starting point.
 - Image names are no longer case-sensitive on Linux.
 - Removed many system log messages that were no longer useful.
 - Now properly honor the **Detailed Connection Logging** preference.
 - Record changes to the **Text Language** preference setting.
 - Fixed a problem that could lead to instability when turning suite sections off and on.
-

Release v16.01

(23-February-2016)

Highlights:

- Many stability and performance improvements.
 - **Updated** embedded Android server to support Android 6.0 devices.
-

User Interface Improvements:

- Added an option to **eggPlant menu->Viewer Preference** to control automatic checking of **Image Duplicates**. This can be disabled to improve performance on slower machines.
- Fixed some display problems with the **TypeText** panel, particularly on narrow devices.

- Corrected a display issue that could cause the edges of the Viewer window to be hidden. This did not impact script execution.
 - Added a maximum zoom size to the **Image Editor**.
 - Improved performance and stability when creating **Image Collections** with many images.
 - **OCR Update** panel now allows *Use Info* and *Copy* features when multiple matches are found.
 - Created an option to suppress achievement notifications on Mac.
 - Vertical Toolbar now supports the **Color** toolbar item.
 - Customizing when the vertical toolbar is up now brings up the horizontal toolbar temporarily.
 - In the script editor, clicking on an image token now shows the same preview pop-up that is available when using auto-complete.
-

Connection Improvements:

- Added menu support to create user-defined keyboards for RDP in **Control menu->RDP Keyboard Layout**.
 - Added support for **UK Keyboards** to RDP.
 - Fixed several RDP key events (French, English, & German).
 - Fixed a problem where eggCloud reservations would not extend if there was any interruption during the connection (even if the connection was re-established).
 - eggPlant no longer replaces newer versions of the Android VNC server when connecting.
-

Scripting Improvements:

- Fixed a compatibility issues introduced in v16 with the **ClipRectangle** property. ClipRectangle once again calculates the HotSpot before clipping is applied. The v16 **CropRectangle** still applies the HotSpot after clipping.
 - Fixed a problem with the **MonthNames** and **WeekdayNames** functions on Mac OS X.
-

Bug Fixes / Tweaks:

- Added logic to limit the amount of memory eggPlant uses to save images on disk in memory. The default is 200 MB.
 - eggPlant Functional no longer stores images in cache when **Detecting Duplicate Images**.
 - Fixed a problem with repeated calls to **Assert** in a called script.
 - Fixed a problem where the Image Doctor would fail when running against **RealVNC Mobile** devices.
 - Fixed a problem modifying the contents of the **Images** directory while eggPlant was **Detecting Duplicate Images**.
 - Added custom image editing cursors for Linux.
 - Fixed problems re-using the Tutorial SUT.
 - Added protection when user-defined keyboard mappings are modified while scripts are running.
 - eggPlant Functional now prevents images from having invalid clipRectangles/cropRectangles when an image is recaptured with a different size.
 - Fixed a problem that could occur when you externally modified suite **Resources** while the Suite was open.
 - Fixed a problem that could occur if you used searchRectangles with fractional values.
 - Added additional logging for diagnostic purposes.
 - Fixed a problem accessing the Tutorial suite on Linux.
 - Fixed a problem calling **RunWithNewResults** in a selection when the script had not previously been run.
-

Release v16.00

(28-January-2016)

Highlights:

- New **Image Editor** that allows you to mask colors, crop the image rectangle, and include/exclude pixels.
 - **OCR Update** panel that helps diagnose and suggests corrections for failed OCR text searches.
 - eggPlant now **Detects existing images** and recommends them when you capture an image that is already present in your suite.
 - **Achievement and Tutorial** system to help people come up to speed on all the latest features of eggPlant.
 - Added a test assets called **Resources** to suites. This provides a consistent place for data files and other test resources to be referenced from.
-

User Interface Improvements:

- In the **Suite window** you can now disable/enable the various side sections to hide areas you don't use.
- From the **Image** tab you can now access an edit mode:
 - Crop
 - Erase individual Pixels
 - Restore Erased Pixels
 - Remove Color Ranges
 - Exclude all except a Color Range
- During an **Image capture** if eggPlant detects the same image already in your suite you will have the option to generate your command using the existing image instead of capturing a new one.
- During an **Image capture** you can choose to save your image into an alternative suite (preferably a Helper suite).

- Added an **Image Update** diagnostic to try trimming your existing image.
 - Added menu-items to **Export** user preferences to transfer to another system and **Import** them.
 - **OCR Update** panel that helps diagnose and suggests corrections for failed OCR searches.
 - **Achievement and Tutorial** system to help people come up to speed on all the latest features of eggPlant.
-

Scripting Improvements:

- **For Mac we have changed the default image type to be PNG instead of TIFF.**
 - Existing TIFF images will continue to work.
 - PNG images are much smaller
 - PNG Images are more widely recognized
 - This change makes it easier to share scripts/suites across platforms
 - If you need to revert you can issue this user command
 - `defaults write Eggplant ImageFormat tiff`
 - In the **Suite window** you can now disable/enable the various side sections to hide areas you don't use.
 - **OpenTestCases** is now a proper function. It will return a copy of whatever test cases are open rather than a mutable object that you could change.
 - Added a function to get a **Resource's** full path by using **ResourcePath("resourceName")**
 - We will now honor any alpha values in Images using the **Precise** text search type.
-

Connection Improvements:

- eggPlant now has predefined keyboard definitions for RDP. In addition to our definitions users can create their own versions of these files for other languages. Select your preferred language from the **Control** menu
 - Fixed several RDP key events.
 - Fixed a problem where eggPlant would not connect to an RFB 5.x server.
 - Fixed a problem when entering incorrect RDP credentials.
-

Scripting Improvements:

- Improved Table Scripts to reload the data file when editing new lines.
 - Fixed some compatibility issues introduced in v15 with extras parentheses
 -
-

Bug Fixes / Tweaks:

- Fixed issues on Window with pasteboard access hanging or crashing eggPlant.
 - Fixed an warning on Mac OS X 10.11 when opening scripts.
 - Fixed a few problems when generating commands from the SenseTalk browser
 - Fixed some problems dragging tabs out of suites
 - Canceling an Add representation didn't always stop the new image from being added.
-

Release v15.22

(5-November-2015)

Bug Fixes/Tweaks:

- Added the ability to connect to eggCloud devices using just their name.
 - Updated Android VNC to 2.61.
 - Fixed a problem connecting to USB android devices from a script.
 - Fixed a problem switching connections to named eggCloud devices.
 - Fixed a problem where the pasteboard would lock-up (Windows).
-

Release v15.21

(17-September-2015)

User Interface Improvements:

- Improved the display to prevent flickering in the remote window.
 - Fixed a problem opening suites with many images and open Image Tabs.
 - Fixed some problems with Suite Searches.
 - Fixed some problems where restored breakpoints could appear on the wrong line (Mac Only).
 - Fixed a problem where Ignore Case was not being honored in the Find Panel (Windows).
-

Connection Improvements:

- Built-in Android Server 2.6 is included in **eggPlant Functional 15.21**. iOSGateway 2.6 is now available for download.
- Added options for Mobile VNC connections to scale the remote screen size down. Optionally allows scaled screens to be blended (for compatibility with older VNC servers).
 - Note: Requires iOSGateway 2.6 or Android Server 2.6.
- Added **Reboot** command for mobile connections.
 - Note: Requires iOSGateway 2.6 or Android Server 2.6.
- Added **KillApp** command to terminate mobile apps.
 - Note: Requires iOSGateway 2.6 or Android Server 2.6.
- Added **ExecuteRemoteCommand** command to .
 - Note: Requires iOSGateway 2.6 or Android Server 2.6.
- Added **SUTinfo()** function for Android devices. **SUTinfo** returns a property list of information about a connected Android device.

Properties include: type, device_os_version, and device.

- Note: Requires Android Server 2.6.

- Corrected some drawing issues when Mobile devices are rotated and the Remote Window was in Capture Mode.
 - Corrected a problem where connections to Android servers would fail after using up all available local ports.
 - Fixed a problem with the window size while rebooting a SUT connected to over RDP.
-

Scripting Improvements:

- Fixed some problems with **Answer From List** command
 - Fixed a problem where text copied out of scripts lost it's formatting.
 - Fixed a problem inserting extra parenthesis for some functions from the SenseTalk Browser.
 - Fixed some problems editing Paired Tokens.
 - Improved the selections when doing a Search in the SenseTalk Browser.
 - Fixed a problem searching the SenseTalk Browser on CentOS
 - The first step of a capture session will no longer default to "Skip Step"
 - Improved the custom Snippet field to expand for longer entries
 - Improved Table Scripts to reload the data file when editing new lines.
-

Release v15.20

(18-August-2015)

Highlights:

- New **SenseTalk Browser** that includes descriptions, templates, and documentation links for common scripting terms.
- **Auto-completion** of punctuation, control structures, and SenseTalk language elements in the script editor.
- Capability to create **custom code Snippets** as part of the SenseTalk Browser.
- **Variable Watcher** allows you to view and edit variable values during

script execution.

- Table tests now support **Data Driven** execution to iterate over values stored in a CSV table.
 - Ability to **refactor scripts** by extracting sections of code into a new script
-

Connection Improvements:

- Windows RDP now uses significantly less CPU on Windows.
 - Values in **Connect** commands and specified using command-line arguments now take precedence over existing values in the Connection List.
 - Built-in Android Server 2.6 is included in **eggPlant Functional 15.20**. iOSGateway 2.6 is forthcoming.
 - Added options for Mobile VNC connections to scale the remote screen size down. Optionally allows scaled screens to be blended (for compatibility with older VNC servers).
 - Note: Requires iOSGateway 2.6 or Android Server 2.6.
 - Added **Reboot** command for mobile connections.
 - Note: Requires iOSGateway 2.6 or Android Server 2.6.
 - Added **KillApp** command to terminate mobile apps.
 - Note: Requires iOSGateway 2.6 or Android Server 2.6.
 - Added **SUTinfo()** function for Android devices. **SUTinfo** returns a property list of information about a connected Android device. Properties include: type, device_os_version, and device.
 - Note: Requires Android Server 2.6.
-

User Interface Improvements:

- The Run window now has a variable watcher in which you can watch variable values while a script is running.

Options include the scope of variables shown and the frequency of variable updates (continuously or only when the script is paused).

- You can now click script and image names to edit them on Windows, rather than having to use the Rename menu item.
- The AHDB in the Run Window will now expand when multiple lines of text are entered into it.
- In the script list you can enable a color-indicator to show which scripts are enabled for Table Actions.

Scripting Improvements:

- Tables can now specify an optional CSV **Data File**. When specified the table actions will be repeated for each row in the file. The first row must contain column headers. Values for each column can be used in the **Action** or **Expected Value** field by using \$HEADER_NAME.
- Tables can now specify the return value from previous lines for use in **Action** or **Expected Value** fields. You can refer to the results of particular line with #LINE_NUMBER or #prev for the immediately preceding line.
- Auto-completion of punctuation, control structures, and SenseTalk language elements in the script editor. Options include parentheses, quotes, commas, curly braces, square brackets, if statements, repeat statements, handlers, script names, local variables, image names, and SenseTalk keywords
- The **answer** command has been extended with "from list", "multiple", and "default answer" options. Example

```
answer "What would you like to drink?" from list ("Coffee",  
"Tea", "Cocoa", "Water") allow multiple default "Coffee"
```

- Capability to refactor script by extracting sections of code into a new script. To extract code to a new script, select the code, right click, and choose "Extract to New Script" in the contextual menu
- Added an **assert** command. When an assert command is executed, eggPlant logs success, failure, warning, or an exception. To improve readability, you can insert the word "that" into an assert command. Example:

```
assert that x > 10
```

Several properties govern the behavior of the Assert command:

The `assertionBehavior` property currently defaults to **Exception**. It can also be set to **Error**, **Warning**, or **Pause** to establish the default behavior for any assert commands that don't specify a behavior. When the behavior is "Pause", a `logWarning` occurs and the script will pause in the debugger on the line following the assert command.

The **`assertionPassCount`**, the **`assertionFailCount`**, the **`assertionErrorCount`**, the **`assertionWarningCount`**, and the **`assertionExceptionCount`** are global counters that are incremented as appropriate each time an assert command is executed. Each failure will increment both the `assertionFailCount` and also one of the error, warning, or exception counters.

The `assertionsEnabled` property defaults to True. If set to False, assert commands are ignored and never executed.

- For Linux: Image Names are now treated as case-insensitive when used in standard eggPlant image commands. Local file operations will still defer to the case behavior of the local file system.

Bug Fixes / Tweaks:

Connections:

- Fixed some issues where a screen size change would not be properly detected at the script level.
- Fixed some problems where the connection list did not update with connections made from scripts.
- Fixed some problems getting a range exception when working in the connection list.
- Fixed a problem with the Bonjour discovery preference not being honored.
- Fixed a problem where RDP connections couldn't be re-established after the connection was closed externally.
- Fixed a problem making some RDP connections from the command line.
- Added support for `WindowsAltKey` to RDP connections.
- Fix mapping of F11 for RDP connections.

- Corrected an issue where local Android device names and serial numbers could fail to display correctly.
- Properly display names of eggCloud devices.
- Fixed a problem with reservations from eggCloud reporting the wrong time.
- Resolved an issue with very slow startups when the configured eggCloud server couldn't be reached.
- Fixed a problem where connecting to an eggCloud SUT using a rule would sometimes connect to the wrong SUT.

Scripting:

- Fixed a crash that could occur when closing a suite while a Table Test was running.
- Fixed several formatting issues in the Tables with values overflowing their cells.
- Fixed a problem where Tables could become disassociated with the scripts they were calling.
- Fixed a problem with **TypeText HomeButton**
- The RemoteWorkInterval will now be checked before Pinches take place.
- Attempting to copy files to/from a SUT will now throw an error if there is no connection.
- Dramatically improved memory usage with OCR commands.
- Fixed an occasional timing issue when specifying ValidWords for an OCR search.
- Better error messages when attempting to perform an invalid OCR operation.
- Fixed a problem where very long return values (e.g. from a URL read) would not be displayed properly on Windows

User Interface:

- Deleting all logs for a script will now remove the script in the Results list.
- Much better recording of Drag and Drop events in Turbo Capture mode on Windows and Linux.
- Fixed a problem opening results in other Suites from Schedules.
- Fixed a problem where aborted Schedules would show scripts with a success status.
- Fixed a problem where tolerant images would display as if they had

- the Pulsing search type selected.
- Fixed a problem where the HotSpot would be re-set to the middle when an image was recaptured using the Update Image Panel.
- Images fixed using the Discrepancy Search in the Update Image Panel will now use absolute pixel values rather than a percentage.
- Fixed a problem saving recaptured images to a helper suite from the Update Image Panel.
- Fixed a problem where the last line of a script was ignored by the Trace feature.
- Cleaned up many formatting issues in the Preference Panes.

Misc:

- Added a "Copy To Clipboard" button for the Host ID in the license panels.
- Fixed a problem on or some Linux distributions where external files and URL resources wouldn't open properly.
- Fixed an issue where special CLI licenses were sometimes prioritized ahead of GUI licenses.

Release v15.14

(09–June-2015)

Bug Fixes / Tweaks:

- Fixed a problem with reconnecting to remotely terminated RDP sessions.
- Fixed a problem with random numbers in scripts on Windows.
- Added support for ZLIB compression to the built-in Android VNC server.
- Added scaling support to the built-in Android VNC server.
 - Users can specify scaling in their Connect command by passing "--scale-screen RATIO" in the AdditionalArgs argument.
- Fixed a problem in the built-in Android VNC server that occurred when connecting and disconnecting repeatedly.

Release v15.13

(27–April-2015)

Bug Fixes / Tweaks:

- Added a GUI control in preferences to select the version of ADB.
- Improvements to automatically deployed VNC server for Android.
- Fixed a condition in the connection list that could result in connecting to the wrong SUT.
- Fixed a problem connecting to Android devices where the connection was prematurely closed by the host.
- Dramatically improved the Suite Search field on Windows when many images are present in the Suite.
- Fixed a problem with ad hoc eggCloud reservations not extending automatically.
- Changed Windows to not stop the ADB Server to facilitate testing multiple simultaneous devices. (Not recommended with ADB 1.0.31)

ADBStopServerOnQuit

Release v15.12

(6–April-2015)

Bug Fixes / Tweaks:

- Added logic to work with a wider selection of Android devices across multiple versions of ADB. By default ADB version 1.0.31 is used.
 - You can switch to the newer version of ADB (not compatible with some older devices) by issuing this command
defaults write Eggplant ADBVersion "1.0.32" (Mac/Linux)
defaults.bat write Eggplant ADBVersion "1.0.32" (Windows)
-

Release v15.11

(26-March-2015)

Highlights:

- eggPlant now Includes a built in VNC server for Android devices. It will be installed when connecting to devices if a VNC server is not already present.
- Improved integration with eggCloud servers.

User Interface Improvements:

- Improvements to the **Table** GUI and Placeholder tokens
 - Dramatically improved performance of **Find** and **Replace** on Windows
-

Connection Improvements:

- Added the ability to use domain\username with RDP.
 - Fixed a crash when RDP connections fail unexpectedly.
 - Corrected a problem launching eggOn app for Android via USB.
 - Addressed an issue connecting to multiple Android servers that were using the same port.
-

Scripting Improvements:

- Made ranges evaluate correctly when different units were used for the endpoints.
 - Fixed a problem where predefined variable would not act as empty when they were modified.
 - Normalized the line ending when reading text at a point and at a rectangle.
 - Fixed a problem archiving property keys with non-identifier characters.
-

Bug Fixes / Tweaks:

- Corrected an issue where **license** were not release when eggPlant was run from the command line or terminated abruptly.
- Fixed problems where the **RunHistory.xml** file could become corrupted.
- Fixed problems with eggPlant responding sluggishly/not responding.
- Fixed a problem reading non-ascii characters from databases.
- Provide feedback if **Bonjour/Zeroconf** is not enabled.
- Fixed a hang when a connection drops during a **CaptureScreen**.

- Cleared an exception that could occur with Show Image Usage.
 - Fixed a condition that could occur when using the **ValidWords** property on a Text Image search.
 - No longer update the **SuiteInfo** file if we haven't modified script metadata.
-

Release v15.10

(2-March-2015)

Highlights:

- New commands to **copy files** and **upload applications** to mobile servers.
- Support for **Pinch** gestures in iOS Gateway.
- New commands to set iOS Device **Orientation**.
- Added **VNC Server Discovery** for servers that advertise via Bonjour/zeroconf. (Previously Mac only)

User Interface Improvements:

- Added the ability to specify the resolution of an RDP session in the connect interface.
 - Added a checkbox to "Save Password" when making a connection.
 - Tables now allow you to define ACTIONS from scripts with spaces in their name.
 - Prevent saving scripts and images with invalid characters in the name (like Colon or Quote).
 - Added the ability to specify an eggCloud port number in the preferences.
 - Vertical and Horizontal Toolbar settings are now saved between sessions.
-

Scripting Improvements:

- Added the commands for **PinchOut/PinchIn** to invoke zooming on a multi-touch device. You can specify just a point, a point and a distance, or two points. Please see the documentation for a full description.

```
// Zoom in on an image with default duration (2 seconds)
PinchOut (At:"MyPhoto")

// Zoom out from a point for 300 pixels
PinchIn (At:(500, 500), Distance:300)

// Zoom in using two image points
PinchOut (At:"LEFT_SIDE", To:"RIGHT_SIDE")
```

- Added commands to **RotateLeft**, **RotateRight**, **SetDeviceOrientation** and **GetDeviceOrientation()**. This commands give you full control over mobile device orientation. Please see the documentation for a full description

```
SetDeviceOrientation "Landscape"
put GetDeviceOrientation()
```

- Added the ability to copy files to/from the **iOS Gateway**. Use the **copy file ... to ...** syntax with **sut:** before the filename to indicate a remote file.

```
copy file "/Users/jonathan/MyFile.txt" to "sut:/tmp/
MyFile.txt"
```

- Added the ability to install and sign applications to **iOS Gateway** controlled SUTs. Use the **InstallApp** command, which takes the **ApplicationPath** key as a reference for the file to install. It can begin with **sut:** to specify a file on the iOS Gateway machine. Optionally you can also specify a mobile provision file with **ProvisionPath** to use to resign the application. Again you can use **sut:** to specify a file on the iOS Gateway machine. Finally **CertificateName** must be a valid Certificate Name that has been stored in the keychain on the iOS Gateway machine.

```
installApp (applicationPath:"/Users/paul/Desktop/
YetAnotherSampleApp.ipa",
           provisionPath:"sut:/Users/paul/Library/
MobileDevice/Provisioning Profiles/0c7acd92-
b4fc-4295-9938-58b759287d3f.mobileprovision",
           certificateName:"iPhone Developer: Paul Landers
(7JZCTVR53Z)")
```


- Added new **archive()** and **unarchive()** functions which can be used to produce an archival text representation of any SenseTalk value and to efficiently restore the equivalent value from such an archive. The archive format spreads list values and property-list properties across multiple lines, making it more human-readable than the format produced by the standardFormat function (which is always on a single line), and the unarchive function is more efficient than the value function at reading it again (although value() can also be used to restore a value from the archive format).

```
put archive of complexData into file "/tmp/archive.txt"
put unarchive of file "/tmp/archive.txt" into restoredData
```

- Improved performance of the **put ... after** command when used to append text or data to a file.
- The **units** property of a value may now be set to a complex unit. Previously, only a simple unit was supported:

```
put 8 into bar
set bar's units to "ft" – this worked previously
set bar's units to "cubic ft / second" – this works now
```

- Added a new '**units(expr)**' syntax for units. This allows a variable or expression to supply the units for a value.


```
set measure to "ft"
set length to 5 units(measure) – sets length to 5 ft
```
- Fixed a problem with setting and retrieving built-in properties (such as format or units) of a property in a property list.


```
set invoice to (number:1, date:the date)
put the format of invoice.date – to show value, not format
set the format of invoice.date to "%Y %A %d"
```
- Improved the **formattedTime()** function to be able to interpret its second parameter correctly as a date/time if it is given in the format shown by the format string, even though it is not one of the usual formats included in the timeInputFormat global property.


```
set displayDate to formattedTime("%A %B %d", "Friday April 3")
```

Bug Fixes / Tweaks:

- Fixed several memory problems related to the Update Image Panel that could lead to crashes and unexpected behavior.
- Fixed crashes on Windows when using RDP from CLI without the user having a proper %HOMEPATH% (i.e. Jenkins),
- Fixed several inconsistencies with the **Team License Server** reporting of which licenses are in use.
- Fixed a problem with how renamed scripts were handled by the Tables and Results system of the Suite.
- Fixed a problem with Clipboard sharing in Live mode using eggMan.
- Fixed errors caused when passing whitespace or empty values as **ValidWords** of an OCR search.
- Fixed a problem where negative values for **the repositionPoint** were not properly honored.
- Updated ADB to use the newest version.
- Fixed a problem with incorrect content-length specifications from eggDrive with non-ASCII characters.
- Fixed a problem where you could start a recording while a script was running.
- Fixed a crash when inserting text before and after a string at the same time.
- Added some improvements for Text-Image-Generation to handle the the **Error: No Error** exception.
- Fixed some problems Removing All Breakpints.
- Fixed a problem with any predefined variable whose value is a list or a property list not being correctly treated as a list or as a property list in some contexts.
- Fixed a problem calling ScriptResults when the script was called from a Table.
- Fixed the standardFormat for data, color, and date/time values.
- Fixed the **join** command and **joined by** operator to correctly treat a range as a list and join its values.

Release v15.02

(19-January-2015)

User Interface Improvements:

- Added the ability to reorder actions in Keyword Driven Tables.
- Added the ability to drop Image names into in Keyword Driven Tables.

- You can type the name of a Script to select it within a suite.
-

Scripting Improvements:

- Fixed a problem where the last line of a script was not traced.
 - The “in” option of the **read** command can now also be specified as “timeout”, “timeout in”, or “time out after”.
 - Fixed the **sin()**, **cos()**, and **tan()** functions to properly handle parameter values in rotational units other than radians. Updated the **atan()** function to return a value with units of radians (which can easily be converted to degrees or other units of rotation).

```
put sin(90 degrees) – 1  
put atan(1) as deg – 45 degrees
```
 - Fixed the **diskSpace()** function, the **size** property of a file or folder, and the **memorySize** and **physicalMemory** properties returned by the **systemInfo()** function to return values with units of bytes (which can easily be converted to GB or other units of dataSize).
 - Fixed a bug with the **URLDecode()** function that would hang or throw an exception if called with a string containing a ‘%’ not followed by two hexadecimal digits.
 - Fixed a bug that would throw an exception for some forms of the **delete** command that resulted in an empty string.

```
put "a" into foo  
delete "a" in foo – this worked fine  
put "a" into foo  
delete the first "a" in foo – this would throw an  
exception
```
 - Added **boxListFormat**, **jsonListFormat** and **standardListFormat** predefined variables, which can be used to conveniently set the listFormat.
-

Bug Fixes / Tweaks:

- Fixed a problem with RedHat 6 package.

- Fixed a problem that would cause eggPlant to run out of memory with large Turbo Sessions.
 - Fixed a crashing issue that could occur when interacting with scripts or schedules in a suite.
 - Fixed a number of problems where UNDO could cause a crash.
 - Fixed a problem where quitting cleanly after Reading Text using OCR.
 - Fixed a persistent memory leak when eggPlant was running in Drive mode.
 - Fixed a problem calling a script as the first action in a Table from the CLI.
 - Fixed an occasional hang on Linux when running scripts.
 - Fixed an exception editing script info.
 - Cleaned up some fatal exceptions when the Global Results Directory did not exist.
 - The UI no longer prompts for a password when a script fails to connect.
 - Dragging a script into a Table will no longer open it in the GUI.
 - Fixed a hang in specific circumstances using URLDecode.
 - Fixed several cases of "NSCharacterConversion" exceptions.
 - Fixed forms of the delete command that threw exceptions.
 - Fix several issues with Chinese UI screens.
-

Release v15.01

(19-December-2014)

RDP Improvements:

- Added Type to **ConnectionInfo()** to be able to identify RDP connections in scripts.
 - Fixed hangs in some situations when making RDP connections on a Windows machine.
 - Fixed crashes when reconnecting to the same RDP connection repeatedly.
 - Improved Clipboard support for RDP sessions on Windows and Linux.
-

User Interface Improvements:

- Improved argument handling when hitting tab while working in Tables.
 - Improved the formatting of the Actual Value when receiving multiline values.
 - Fixed some focus issues with the Turbo Captures Image Rectangle
 - Fixed a problem picking Table Actions from Helper Suites on Windows.
 - Automatic Image Naming preferences is now also honored in Turbo Capture mode.
 - Fixed performance problem with Table Image selector for suites with thousands of images.
 - Improved behavior when updating the final image in a Turbo Capture Session.
 - Fixed some issues changing script and table names.
 - Added the ability to delete a table from the UI.
-

Scripting Improvements:

- Improved the OCR ValidPattern to automatically include other marks in the pattern (like percent symbol: %).
 - Dramatically Improved execution time on scripts that had thousands of posted results
 - Fixed a problem where minutes were treated as seconds.
 - Fixed a problem where username and password in a property list were not always honored for RDP connections.
-

Bug Fixes / Tweaks:

- Added Cancelling an OCR search on Windows / Linux.
- Fixed a problem where eggPlant could hang when terminating a connection.
- Added ChinesePRC and ChineseTaiwan to the Lanuage List.
- Fixed occasional hang running scripts on Linux with Run Window up on screen.
- Fixed a problem where **CaptureScreen** could return an image from a prior connection immediately after making a new connection.
- Fixed a problem where the Capture Screen UI would offer to save

- images as **.suite** files.
- Fixed a related problem where moving UP to the suite level in the Capture Screen UI would select the local suite folder and close the window.
-

Release v15.00

(2-December-2014)

Highlights:

- New **Tables** Feature for keyword-driven testing
- **Turbo Capture** mode for script recording
- Added support for **Microsoft RDP** connections
- **Upgraded OCR** Engine

eggPlant Functional v15 for Linux requires RedHat Version 6+

Keyword Driven Testing:

- Using the new **Tables** section of the Suite you can now create a straightforward workflow by selecting commands and functions from a list.
 - The most common eggPlant Functional primitives are already included
 - You can add your own scripts to the list by specifying the calling arguments and return value (for verify actions) in the **Script Info** section (Use the "i" button in the Script Editor).
 - In the **Table**, select an action and then use Tab to move between arguments and expected values.
 - For verify actions (functions), you can specify an expected value to be compared to the returned value.
 - The expected value can be any SenseTalk expression.
 - See the latest results for each line as you run the test; for past runs, you'll find results in the **Results** tab.
 - Read Full Documentation Online
-

Turbo Capture:

- You can now generate scripts using the new **Turbo Capture** feature,
 - interacting with the SUT in live mode and then naming your images afterwards.
 - Get started by hitting the **Start Session** button on your toolbar or by selecting **Start Turbo Capture Session** on the Control menu.
 - Interact with your SUT in live mode to perform a sequence of steps and then **End Session** when you are finished.
 - NOTE: If you want to record a mouse move (without a click) tap the Shift key.
 - Name your session, if you want, and then use the **Open Session** button to review your workflow and begin naming and adjusting your images.
 - You can stop at any time and return later, your progress is saved automatically.
 - When you are finished, hit the **Generate Script** button to record all the images and create your script.
 - Read Full Documentation Online
-

RDP Connection Support:

- Now you can make connections to computers using Microsoft's **Remote Desktop Protocol (RDP)**.
 - When adding a connection select "RDP" as your connection type, add your Windows username and password.
-

Upgraded OCR Engine:

- We've upgraded to the very latest **ABBYY version 11** on ALL platforms. This means that OCR will now behave the same between eggPlant platforms.
 - The new engine is much more accurate when reading text, but this may result in it reading things slightly differently than it did in version 14.
 - A new **ValidWords** property has been added, which can be set to either a list of words or a string containing words separated by whitespace. When **ValidWords** is used, the OCR engine will

- attempt to read everything as one of the specified words.
- During an OCR search setting the new **ValidWords** property with a value of * (asterisk) will **only** recognize words that are part of the target phrase.
 - We have added the ability to read rotated and upside-down text by specifying a **TextRotation** property with the value of **Clockwise, CounterClockwise, UpsideDown**.
 - You can now specify a regular expression to be used when doing a **ReadText** command by adding a **ValidPattern** property. The **ValidPattern** property should be set to a regular expression to be used for matching each word.
 - You can now customize what languages are shown in the OCR panels from the Text Preferences.
-

Scripting:

- When using a list or range as an iterator, in addition to using `NextValue` to retrieve the next sequential value, the new `CurrentValue` property gives access to the current value without advancing the `CurrentIndex`. The `CurrentValue` property is also a container, acting similar to a reference to that value, allowing you to replace or modify that value in the list.
- A `repeat with each` can now assign multiple loop variables on each iteration, from a source iterator that supplies a list of values.
 - `repeat with each (x,y,z) in ((12,22,14),(6,9,18),(15,3,44))`
- The same rules used for multiple assignment are applied, so variables can be omitted for values that you don't care about, and excess values are ignored unless the final variable name is followed by an ellipsis (...) to collect them as a list.
- A property list can now be used as an iterator. When iterating in a `repeat` loop, it supplies each key and its value as a list of two items (Note: To preserve compatibility with previous behavior, the `repeat` must specify a list of loop variables for this process to work). An object that defines its own iteration (i.e. an object with an `objectType` of "iterator") will use its own iteration as before; other objects will now iterate over the (key,value) pairs for their properties.
 - `set people to (Marty:12, June:16, Aaron:6, Grace:11)`
 - `repeat with each (name,age) in people`
 - `put name && "is" && age && "years old"`

- end repeat
- When iterating over the properties of a property list outside of a repeat loop, nextKey can be used to retrieve the next key, nextValue to retrieve the next value, or nextKeyValue to retrieve the next (key,value) pair. CurrentKey, currentValue, and currentKeyValue can be used to obtain the corresponding value for the current property without advancing the iteration.
- A start iteration command has been added to start or restart the iteration of any iterator. This is the only way to begin iteration again for a property list. It can also be used to start iteration for lists, ranges, and chunks, or you can achieve the same result for those types by setting their currentIndex to 0.
- Mixed fractions like "6 5/8" are now supported provided there is a single space character between the whole number and the fraction and no spaces around the '/' character. Simple fractions with units, of the form "3/4 ft", are now processed by the compiler as a single number (rather than 2 numbers and a division operator) provided the fraction is directly written with numbers (not variables or other expressions) and is followed by a unit name.
- Added support for special symbols \leq \geq $+$ $+$ \times $-$ \times \div \neq as operators in expressions. Also, the letter X can now be used as a multiplication operator.
- Added support for using ! before a quoted string as a shorthand for calling the merge function.
 - put !"The month is now [[the monthName]]" -- "The month is now September"
- Added the ability to include some additional characters in identifiers (variable names and handler names) in addition to the alphanumeric characters and underscores that were previously allowed.
- Added as words, as a word, as hex, as hexadecimal as equivalents for the corresponding format operators (format "words" or format "hex"). For **as words**, capitalization follows that of the word "words".
- Added **highestValue()**, **largestValue()**, and **lowestValue()**, **smallestValue()** as synonyms for the **maximum()** and **minimum()** functions, respectively, and added natural language syntax for them:
- Added but at least, but no less than, but at most, and but no more than operators to simplify the task of limiting a number to a minimum or maximum value (internally, these operators call the largestValue or smallestValue functions):
- Added **from now** as a synonym for the hence operator.
- Added a repeated operator to create a string of text repeated a given

number of times, up to a specified length, or to create a repeated list:

- put "\$" repeated 12 times -- "\$\$\$\$\$\$\$\$\$\$\$\$"
- put "Hello" repeated to length 12 -- "HelloHelloHe"
- For a list of repeated items, use as (a) list or (a) list of with repeated:
 - put "\$" repeated 6 times as a list -- ("\$", "\$", "\$", "\$", "\$", "\$")
- Added a number of operator for multiplication or repetition (Note: This syntax requires a literal number before the word of, not a variable). This is particularly useful for fractions of numbers and units. If the value is a number or a list, it is multiplied by the number. Otherwise, the number of operator will cause it to be repeated like the repeated operator.
- Changed the standardFormat function. It now records numbers without quotation marks, and uses scientific notation when appropriate to accurately represent the value, without using the numberFormat. Numbers with units will include the unit name. Strings are now quoted using the "@" quoting style, with embedded backslash escape sequences as needed.
- Added LightSpeed and googol as predefined variables.
- Added predefined variables heavyCheckMark, xMark, and heavyXMark in the Common symbols category.
- Allow before or after in addition to from in the seek command.
- An emdash character (—) can now be used to start a comment, since some text editors automatically replace two dashes (--) with an emdash.
- Added support for considering case / ignoring case in the intersection, union, excludeltems and uniqueltems functions. These functions now follow the current setting of the caseSensitive property by default (previously they were always case-sensitive).
- A range expression can now be written using "down to" in addition to "to" and "..".
- Strings containing numbers in scientific/exponential notation are now recognized as numbers.
- The word **by** can now be used before the step value in a repeat loop (in addition to **step** or **step by**).
- The maximum and minimum functions now return empty, not zero, when called with an empty parameter (or with any list containing no numbers).
- Fixed a tiny bug that would allow two @ signs before a string literal. Only one is allowed.
- Fixed an error that would allow "database" to mean "new database" -- the word "new" is now required.

- Fixed a bug with « » quotation marks, which were omitting the first character inside the quotes.
 - Fixed addition/subtraction of a number and a single-item list containing a number to always result in a simple number, not a list.
 - Fixed multiplication of a number and a single-number list to always result in a single-number list (scalar multiplication of a number and a list).
 - Fixed an obscure bug with setting a property of a list item of a non-existent property of an empty variable.
 - Fixed an issue with predefined variables not being recognized as having a value in certain contexts.
 - Improved error message for some cases where a command is called with a range.
-

Scripting Units Support:

- Added units support. Numeric values may now include a unit name to identify the type of value. SenseTalk is aware of the relationships between different units and automatically converts to compatible units as needed during calculations or when requested.
- Many unit types are recognized, and more will be added in the future. The **unitTypes()** function gives a list of all of the different categories, including synonyms in some cases.
 - put the unitTypes
- Many unit names are recognized for each type, including singular, plural, and abbreviated names for many units. The complete list of unit names is given by the **unitNames()** function (there are thousands of unit names).
- To use units, simply include a unit name after a number or variable name within an expression:
 - put 3 feet into counterHeight
 - add z grams to totalWeight
- Derived units like area and volume can be expressed in several ways:
 - set surface to 9 cm² -- 9 square centimeters

- set area to 2 yds*ft -- 6 square feet
 - set base to 5 sq in -- 5 square inches
 - set vol to 3 cu yds -- 3 cubic yards
 - A value can be converted from one unit to another (compatible) unit by specifying as followed by the other unit name:
 - put 1 gallon as liters -- 3.785412 liters
 - The mathematical operators and functions all maintain and/or convert units as appropriate, to the extent possible. For example, the following addition first converts 8 inches into centimeters before adding:
 - put 8 inches + 3 cm — 23.32 centimeters
 - A value may be tested to see if it is of a particular unit type using the is a operator:
 - put 4 cm is a distance — True
 - put 9 ounces is a duration — False
 - Two global properties control the behavior of the units system. The unitsEnabled is now set to True by default to enable the use and display of units. Setting it to false will cause unit names to be ignored and treat all numeric values as plain numbers.
 - The strictUnits property is also set to True by default. This setting preserves the integrity of calculations, throwing an exception for any calculation involving incompatible units, including addition, subtraction or comparisons between plain numbers and those with unit values. Set it to false to provide more lenient behavior, which will not throw exceptions but may cause values to lose their unit association in some situations.
-

Bug Fixes / Tweaks:

- We now back record helper suites so you can see which other suites are "helping"
- Opening logs that are in a helper suite will no longer raise an exception.
- Nonexistent suites will no longer show up in the Open Recents menu.

- Fixed an occasional hang when running a script from Command Line.
- Fixed a number of license messages that were unclear when no GUI license is found.
- Added a **LogWarning** when an Ask/Answer panel is called while running from in command line mode.
- Added the ability to duplicate a Schedule.
- When pasting "fancy" quotes from external sources into a script we will now convert them to standard ASCII quotes to prevent parsing and compiling problems.
- Fixed an occasional crash when connecting to a SUT.
- Added a feature to temporarily freeze the remote screen to simplify capturing images of transient UI elements.
- Fixed an occasional hang when running a script from Command Line.
- Added keyboard shortcuts to switch between open tabs in a suite. Command+Shift+[and Command-Shift+] on Mac Control+Shift+[and Control+Shift+] on Windows/Linux
- Fixed a problem where Function key shortcuts didn't show in the menus.
- Now includes **CaptureScreen** among the list of "interesting" log lines when filtering.
- On Mac 10.9 and 10.10 we have modified eggPlant Functional to not **AppNap**. This power saving feature caused disruptions to scripts running in the background.
- Fixed a rare crash when launching a VNC server on Android over USB.
- Applying an image "Fix" from the results will reflect that change on any open copies of the image.
- When selecting an image from the screen during Update Image the replacement image will no longer grow by 2 pixels in both dimensions.
- Fixed a problem with mousing over strings that corresponded to Windows named ports (like PRN or COM1).
- Fixed some problems with Copy/Paste throwing NSRange exceptions.
- Fixed some exceptions that could happen when customizing the Remote Window Toolbar.

Viewer Window:

- **Zoom To Actual Size menu item.** Added a command to return the Viewer window to it's standard size.
-

Bug Fixes / Tweaks:

- Fixed a problem with detecting Android USB devices on Windows.
 - Fixed a problem where capturing to a new image collection that preventing saving an image.
 - Made the window zoom-in towards the capture area instead of the corner.
 - Improved UI of eggMan welcome screen for clarity.
 - Fix a problem with toggling back to Live mode when the Capture mode Guide was present, also added the ability to close the Capture Guide.
 - No longer present an Update Available panel when running eggPlant with the -RunOnLaunch option to start a script immediately.
 - More consistent display of the product name when running eggMan vs eggPlant Functional.
 - Fixed a problem hitting an Internal error when starting an image search while the Image Doctor was already searching.
 - Fixed a problem trying to call Omega13 before a suite had been set, particularly when using eggDrive.
-

Release v14.20

(4-August-2014)

Highlights:

- Zoom In/Out in the Viewer window
 - New script commands/functions for use with mobile devices
 - Introductory Help for new users
 - Numerous bug fixes
-

Viewer Window:

- **Zoom In/ Zoom Out menu items.** Added these zoom commands to the Control menu, enabling you to zoom in and out in the Viewer window.
-

Enhanced Mobile-Device Control

- ***PressHomeButton* and *PressBackButton* commands**, which press the respective button on mobile devices. (Note: In older versions of Android, the *PressBackButton* command presses the menu key.)
 - ***ADBCall* function for Android.** This function enables you to make Android Debugger (ADB) calls from eggPlant Functional, and returns the output of those calls.
 - ***LaunchApp* command for iOS Gateway.** This command launches a given app on a device connected through iOS Gateway.
-

Scripting Enhancements

- Mixed fractions like "6 5/8" are now supported provided there is a single space character between the whole number and the fraction and no spaces around the '/' character.

```
put 1 3/4 is 1.75 -- true
```

- Added the ability to include underscores ('_') in numeric literals to separate groups of digits for greater readability.

```
put 1_000_000 is a million -- true
```

- Numbers can now be expressed in scientific notation without a '+' or '-' before the exponent. If no sign is given, a positive exponent is assumed (this also applies to the *JSONValue()* function for evaluating JSON expressions):

```
put 1e6 is a million -- true
put 1.23e6 equals 1.23e+6 -- true
```

Bug Fixes / Tweaks:

- Fixed a problem where eggPlant Functional could hang on the Update Image panel.
 - Fixed a SenseTalk crash related to internal caching of named objects (most commonly script files on disk).
 - Fixed a problem that could cause eggPlant Functional to crash when running a script with a large block quotation.
 - Fixed a problem that could cause the OCR to fail when reading text at a point.
 - Fixed a problem that caused aborted scripts to be recorded as failures when the Update Image panel was triggered.
 - Fixed a problem that caused eggPlant Functional to incorrectly report that an image contained no searchable pixels.
 - Fixed a problem that caused aborted scripts to be recorded as failures when the Update Image panel was triggered.
 - Fixed a problem that caused breakpoints in the Run window to be ignored.
 - Fixed a situation in which users could not use their maximum number of eggPlant Functional licenses.
 - Enabled the JSONValue function to handle numbers in scientific notation without a sign on the exponent.
 - Enabled mobile commands such as Tap to trigger Omega13 script recovery.
 - Made several improvements to the way eggCloud SUTs are shown in the Connection List.
-

Release v14.12

(12-May-2014)

Scripting:

- Setting the **ScriptLogging** to **Off** now continues to show TestCase logs, it can be disabled by setting the logging to **Silent** or **Minimal**.
- Added support for using the Alt-GR key (Right Alt Key) for auxiliary keyboard functions. If you prefer having a right hand Alt modifier key please set the following preference

```
defaults write Eggplant USE_ALTGR_FIX 0
```

Bug Fixes / Tweaks:

- Fixed a problem where License Expiration warnings could not be dismissed.
 - Fix a situation where eggPlant could hang when capturing an image.
 - Fix a problem where Chinese licenses would not properly load the Chinese UI.
 - Fixed a problem calling runscript on Linux where quoted values might not be properly passed resulting in argument confusion.
 - Improved eggCloud automatic reservations to reserve for an extra block if the current block was just about to expire (less than 1 minute).
 - Fixed a problem where very deeply nested scripts would display incorrectly in the Schedules Panel.
 - Several fixes to icons and colors on Linux systems.
-

Release v14.11

(27-April-2014)

Highlights:

- Toolbar buttons for the HomeButton and RotateMobile commands
 - Additional image preview in script results
 - Several bug fixes
-

Viewer Window:

- **RotateMobile and HomeButton Toolbar Items.** Added toolbar buttons that insert *TypeText RotateMobile* and *TypeText HomeButton* commands into your script.
-

Script Results:

- **Image Previews for Images that Were Not Found.** It has been the case that when you click on a line that references a found image in your script results, a preview of that image is shown. Now that same behavior also applies to lines that reference an image that was not found.

Bug Fixes / Tweaks:

- Fixed an issue that could cause eggPlant on Mac to not properly report a Tolerant image search type.
- Fixed an issue where an expiring license warning could show up BEHIND the splash panel, which made it appear that the launch had hung.
- Fixed an issue on Windows and Linux where a Suite would store its default Results directory as a Results OVERRIDE (which would then persist if the suite was moved or copied).
- Fixed an issue that could cause eggPlant to crash when you were saving a new image.
- Fixed an issue that could cause eggPlant to hang when the Update Image panel opened.
- Fixed an issue that would cause eggPlant to falsely report that it had previously crashed.
- Changed the behavior of full-screen VNC refreshes. Previously, eggPlant could close and re-open your VNC connection on a full-screen refresh; this is no longer the case.
- Fixed an issue that was causing the Connection List filter to not work on Windows and Linux systems.

Release v14.10

(10-March-2014)

Highlights:

- Update Image Tools for Retraining Scripts.
- Automatic Detection of WaitFor times.
- Deeper integration with eggCloud (v1.1.5).
- Enhanced Remote Screen Window for Mobile Devices.

Interface:

- **Update Image Menu.** Added several new features to make it easier to update the images in your scripts. Use the Run > Update Image menu to control Image Doctor settings (now expanded) and new options for updating images:
 - What to do when multiple occurrences of an image are found
 - What to do when an imageFound function returns false

- Whether or not to validate every image search, successful or not
 - **Update Image Panel.** Expanded the former Image Doctor panel with the following features:
 - An OCR search heuristic, which reads text in your image and attempts to find instances of that text on the screen
 - A Prior Location heuristic, which shows where your image was last found in a completed test.
 - The New Representation field, in which you can set a suffix with which to auto-name new images added to collections
 - The Properties feature, which shows the properties that are being used for your current image search, including the default properties of your image, and properties that are overridden at the script level.
 - **Diagnostic Menu.** This new menu allows you to choose the image-searching heuristics that are used in the Update Image Panel.
 - **Mark Prior Result.** From within a Scripts Results panel, you can mark which run you want to use as your “Prior Result” for use by the Prior Location Heuristic on the Update Image Panel.
 - **Vertical Remote Toolbar.** We have replace the obsolete “Mobile Drawer” with a vertically formatted toolbar that allows much better access on portrait-layout Mobile devices (iPhone and Android). Additionally we
 - **Insert Repeat Loops.** From the Script Insert tool you can highlight a block of your test script and create a repeat loop around the section.
 - **insert test cases.** Also from the Script Insert tool you can highlight a block of your test script and turn it into a TestCase.
 - **Image Backgrounds.** Added background colors for image previews in the suite editor and Save panel, click to cycle through backgrounds.
-

eggCloud Integration:

eggPlant 14.10 REQUIRES eggCloud 1.1.5 or higher for proper integration

- **Ad-Hoc Reservations.** You can now double-click on any available (not reserved) SUT to create an ad-hoc reservation.
 - **Automatically Extend Reservations.** You will now automatically extend an connected SUT reservation if the device is still available when your reservation ends.
 - **Scripted Connections.** Connect to a SUT by defining a set of properties. If a device is available (or you have a matching reservation) that device will be used. EG:
 - **Connect (**
-

Scripting:

- **WaitFor property.** WaitFor can now be a property of any image-searching command. Like the WaitFor command, the WaitFor property is the maximum time eggPlant will wait for the given image to appear on the SUT. Example:
 - Click (image: "Done_button", waitFor: 2 minutes).
- **Automatic WaitFor time calculation.** When you capture an image and generate a script command, eggPlant now automatically inserts a WaitFor property in the command if there was a significant delay detected before the image appeared on the SUT.
- **Support for Additional Image Formats.** We now support the following image formats on all platforms: .tiff, .tif, .png, .icns, .ico, .gif, .bmp, .jpg, .jpeg.
- **Default Data Format.** Added the `DefaultDataFormat` global property, which determines the format of BLOB data in scripts. Valid settings are auto, text, hex, base64, and size. The auto setting uses hex for shorter values and size for longer values. The text setting maintains backward compatibility.

- The predefined variables **yes**, **no**, **on**, and **off** are now treated as native boolean values. This allows the equal operator to compare them as equal to **true** or **false** (when Normal boolean comparison is in effect, see below).
- Added **the booleanComparison** global property. When set to Normal (the default), this now allows the values Yes and On to compare as equal to True (or each other) and Off and No to compare as equal to False. Although Empty is treated as a valid boolean value (equivalent to False) it will not compare as equal to False. If **the booleanComparison** is set to "Strict", the values Yes, No, On, and Off will compare as strings, and thus will not be equal to True or False. If **the booleanComparison** is set to "Lenient", any value that evaluates to the strings "Yes", "No", "On", or "Off" will be treated as booleans for purposes of equality testing with other booleans.

```
put Yes is true -- true
set the booleanComparison to Lenient
put "off" = false -- true
```

- Updated the **StandardFormat** function to properly support the new boolean values.
 - Enhanced the **JSONFormat** and **XMLRPCFormat** functions to work with the new boolean values, and also to correctly handle <missing value> (null). In XML-RPC missing values are rendered as <nil /> which is not part of the XML-RPC spec but is a semi-common extension used by Python and some other systems. SenseTalk now recognizes <nil /> and <ex:nil /> when reading XML-RPC values and will convert them to <missing value>.
 - Changed the **add properties** command and **adding properties** operator to treat a property of the destination property list whose value is <missing value> the same as though the property were not there, and replace it with the corresponding source property value.
 - Enhanced the **charToNum** and **numToChar** functions to be able to work with Unicode values greater than 0xFFFF.
 - Fixed the **move file** and **copy file** commands to work correctly when the source file is specified with a path beginning with "~".
 - Fixed a problem when a recordPrototype property for a database table included properties corresponding to columns in the table, those values would be automatically propagated to the database in some cases.
-

Bug Fixes / Tweaks:

- Added FAILURE tag support to JUNIT logging for better compatibility with JIRA and Jenkins.
- Failure to license is no longer considered a crash.
- Improved the ability to submit support requests through proxy servers.
- Fixed a problem that caused an ADB delay when not connecting to an Android server.
- Resolved a problem where eggPlant would not terminate when an Android connection was left open to an Android SUT.
- Fixed a problem that could lead to a “Tried to add nil to array” when an Android device was plugged in via USB.
- Cleaned up some inconsistencies with the preview text when using the Script Insert tool.
- Fixed a variety of problems with running multiple simultaneous Searches using the Image Doctor.
- Fixed a problem where eggPlant would pause when mousing over a network resource declaration beginning with a double-slash (//).
- Cleaned up several cosmetic artifacts in the preference panes on Windows.
- Solved a lock-up problem when closing the suite while generating a command using an existing image.
- Cleaned up some display artifacts with the auto-completion tool on Windows.
- Resolved a situation when schedule runs completed.
- Improved handling of the **ClipRectangle** property for images.
- Resolved a hang when using the Find feature in the Run Window.
- Fixed a crash on Mac when trying to SSH to an unavailable port.
- Improved behavior of Context Menu in the Connection List.

Release v14.01

(13-November-2013)

Highlights:

- Support for improved eggOn for Android.
- Ability to enter a license from the command-line.
- Numerous Bug Fixes.

Licensing:

- eggPlant now supports the ability to enter a license key from the command line by passing the `-LicenseKey` and `-LicenseUser`. After applying the license eggPlant will immediately terminate with success condition (0) if the license was accepted.

Mac/Linux:

```
Eggplant.app/runscript -LicenseKey "xxx-xxxx-xxx" -  
LicenseUser "zzzzz"
```

Windows:

```
eggPlant/runscript.bat -LicenseKey "xxx-xxxx-xxx" -  
LicenseUser "zzzzz"
```

Connection List:

- The Connection List now groups eggCloud and Standard connections separately.

Bug Fixes / Tweaks:

- Fixed a problem on Windows where the `URLTimeout` property was never honored. If URLs are timing out now, you might need to increase this value from its default of 30 seconds.
- Fixed a bug where negative image hot spots were off by a pixel.
- Fixed a bug where **Precise** image search type was not honored when an image was reloaded.
- Fixed a problem on Mac OS X 10.9 where Gatekeeper would not recognize eggPlant's signature.
- Fixed a problem restarting Android VNC servers connected through USB.
- Fixed a bug with autocomplete would sometimes insert a stray `<CR>` character into the script.
- Corrected an `NSInternalInconsistencyException` on Windows during

- socket communications (including Drive mode).
 - Fixed a bug where the screen capture buffer doesn't refresh properly after an exception is thrown.
-

Release v14.00

(24-October-2013)

Highlights:

- Added Database Integration via ODBC
 - Automatic Image-Naming for images with Text.
 - New Rotated Image Searching
 - Introduced new Mobile Commands
 - Connect List Filtering and Other Improvements
-

Compatibility

- **Mac OS X 10.7, 10.8, 10.9**
This release is compatible with Mac OS X 10.7 and higher, Mac OS X 10.6 and below are no longer supported.
 - **Windows XP, 7, 8**
This release is compatible with Windows XP, 7 and 8. Please be aware that releases AFTER April 8th, 2014 will drop Windows XP compatibility.
 - **Linux Debian, RedHat**
This release is compatible with 32-bit versions of Debian and RedHat based on glibc 2.5 and higher. 64-bit distributions must have supported versions of all dependent 32-bit libraries.
-

ODBC Database Integration

- Added the capability to read to and write from available ODBC databases. See the eggPlant documentation for complete usage and instructions.

Interface

- **Auto Image-Naming.** Now, when you capture an image, text that is detected in the image becomes the default image name. You can disable this behavior in the the eggPlant->Preferences->Viewer.
- **Image Capture.** Has now been improved with capture handles, dynamic cursors to indicate resizing and the following modifier keys can be use:
 - Arrow Keys Move The Capture Area
 - +Ctrl/Command Move The Hot Spot
 - +Alt/Option Enlarge/Shrink the Capture Area
 - +Shift Move In Increments of 10
- **Image Scaling and Rotation.** The Image interface now allows you to specify the default scale and rotation values to use when an image is searched for. This value can be overridden in a script.
- **Auto completion** in scripts has been added for Windows and Linux versions of eggPlant. You can trigger completion by hitting the Escape key or selecting "Edit->Complete" from the Menu. The editor will attempt to complete your statement with a valid eggPlant Command, constant image name, script name, or variable from your current script.
- **Image-Popover preferences.** Added preferences to all users to turn enable and disable image popovers, both in the image list and in scripts.
- **eggCloud preferences.** Added a preference panel to connect to an eggCloud installation.

Connection List:

- The Connection List now has a filter field to narrow down the connections shown by display name. If you are using the eggCloud

device-reservation system, you can also filter the Connection List by SUT operating system, version, and meta tags.

- The Connection List now will now show devices available from an authenticated **eggCloud** system. You are able to create reservations and connect to reserved devices directly from the connection list.
- The Add/Edit Connection panel now has a **Connect** button. When you click the **Connect** button, a connection is immediately established.
- The Add/Edit Connection panel now has a ***Remember This Connection*** checkbox, which is selected by default. When this checkbox is selected, your current connection is permanently saved in the Connection List; otherwise, the connection is removed at the end of your eggPlant session. Temporary connections are shown in italicized text in the connection list.
- Connections made from a script and reverse connections are no longer automatically saved in your Connection List.
- You can now choose the columns that you want to view and hide in the Connection List. From the Connection menu, use the Customize Columns submenu to select the items you want to see.

Scripting:

- The current **RecordMovie** command on Mac OS X is now **deprecated**. The command can still be called but it will not be supported on future versions of Mac OS X. A replacement movie technology will be available in a future release of eggPlant.
- For **scaled image searching** we now provide TWO modes of scaling relative to the current SUT size. The mode previously named scale: "Auto" is now called scale: "Proportional" to clarify that it will scale proportional to the smaller change in SUT height or width. To scale an image to the SUT height and width independently use scale: "Stretch".

- Added swipe commands (***SwipeUp***, ***SwipeDown***, ***SwipeLeft***, and ***SwipeRight***) that mimic touch-screen swipe actions better than DragAndDrop. The global property ***the SwipeSpeed*** corresponds to the conventional *MouseDownSpeed*.
- Added other mobile commands that behave the same way as their conventional counterparts:
 - Tap = Click
 - DoubleTap = DoubleClick
 - Press = MouseDown
 - Release = MouseUp
- You can now add a *Rotate* property to an image search, to indicate a degree of rotation (or list or range of degrees) if you expect an image to appear at a different angle than was originally captured. Rotation is clockwise.

```
Click (name: "appIcon", rotate: 180) // Searches
for appIcon at 180 degrees from its original angle
Click (name: "appIcon", scale: 45 to 90 by 5) //
Searches for appIcon tilted 45 to 90 degrees, in 5
degree increments
```

- We now have a function **SuiteInfo()** that returns the following formation about a Suite:
 - **Name**: the name of the suite
 - **Path**: the full path of the suite
 - **ScriptsFolder**: the path of the Scripts folder
 - **ImagesFolder**: the path of the Images folder
 - **ResultsFolder**: the path of the Results folder
 - **Version**: the eggPlant version number of the suite (usually the version of eggPlant that created the suite)
 - **Description**: the description of the suite as entered on the Settings tab of the suite
- Added a **format**, **with format**, or **using format** operator to allow specifying a particular format for an individual numeric, date, or data value.


```
put (13 / 2) with format "00.000" -- displays "06.500"
```
- Added the ability to access the **format** property of a variable containing a number or a data value (previously only date/time values could have a format).

```
put 13 into foo
set the format of foo to "hex"
put foo -- 0xD
set foo's format to "words"
put foo -- thirteen
put "000.000" into foo.format
put foo -- 013.000
put foo's format -- 000.000
```

- **The numberFormat** or the **defaultNumberFormat** property (or the format property of a numeric variable) can now be set to "Hexadecimal" or "Hex" to display numbers in hexadecimal format:
set the numberFormat to "hex"
- Added the **defaultDataFormat** global property to control the way binary data values are displayed. The defaultDataFormat can be set to any of "auto", "text", "hex", "base64", or "size". In addition, the auto and text formats can include a parameter: "auto: *maxHexSize*", "text: *stringEncoding*".
- Added **trimAll()** function (similar to trim, trimStart, and trimEnd) to remove all whitespace (or other characters) from a text value. This function takes one or two parameters. The first parameter is the source text value to be trimmed. The second (optional) parameter is a text string defining the set of characters that will be removed from the source text. If the second parameter is not given, whitespace and newline characters will be trimmed by default.
- Added **keepCharacters()** function which removes all characters from text except for alphanumeric characters (letters and digits) or another set of characters that you specify. This function takes one or two parameters. The first parameter is the source text value to be filtered. The second (optional) parameter is a text string defining the set of characters that will be kept in the source text. If the second parameter is not given, letters and digits will be kept by default.
- Fixed expressions of the form **a.func(b)** or **a's func(b)** to work when a is a variable. Previously, expressions of this form worked when a was a number or a literal, but not a variable. Either syntax is equivalent to calling **func(a,b)**.
- Added a new syntax for comparing two values for equality: **val1 and val2 are equal**.
put 5 * 3 and 15 are equal
put "CAT" and "cat" aren't equal considering case

- Added **the same as** as a synonym for **equal** or **equal to**. The following all report 'true':

```
put 5 * 3 is the same as 15
put 4 + 6 isn't the same as 2 * 12
put "cat" and "dog" are not the same
```
- Changed the **standardFormat()** function to use square brackets rather than curly braces for lists.
- **HotSpotScaling**. Added the HotSpotScaling global property, which determines whether image hot spots are scaled when you search for a scaled image. The values are as follows:
 - **Inner**. Inner works the way hot spot scaling has worked in the past: if a hot spot is within the searchable pixels of an image, it is scaled with the image. If the hot spot is offset from the image, it is not scaled. Inner is the default value of the HotSpotScaling.
 - **All**. Hot spots are always scaled proportionally when you search for a scaled image.
 - **None**. Hot spots are never scaled when you search for a scaled image.
- **Ask/Answer behavior**. When Ask or Answer commands are called in a script run from the command line, they will now act as though they had been called with a timeout parameter that timed out immediately. When Ask File or Answer File commands are called, they will act as though their Cancel button had been clicked. The previous implementation was undefined (it would behave differently in different environments), but it can be restored by using:


```
defaults write Eggplant
OldCommandLineAskAnswerBehavior YES
```
- **BooleanComparison**. Added the booleanComparison global property, which offers choices of how boolean values are compared:
 - **Normal**. When set to Normal (the default), this now allows the values Yes and On to compare as equal to True (or each other) and Off and No to compare as equal to False. Although Empty is treated as a valid boolean value (equivalent to False) it will not compare as equal to False.
 - **Strict**. If the booleanComparison is set to Strict, the values Yes, No, On, and Off will compare as strings, and thus will not be equal to True or False.

- **Lenient.** If the booleanComparison is set to Lenient, any value that evaluates to the strings "Yes", "No", "On", or "Off" will be treated as booleans for purposes of equality testing with other booleans.
-

Bug Fixes / Tweaks:

- Fixed an editor bug that could raise a glyphRangeForCharacterRange: exception.
- Fixed a bug where sometimes the Run Selection did not properly correspond to the actual selection in the Script Editor.
- Fixed a long-standing issue on Windows that limited how many times you could make a new connection. You should now be able to make thousands of calls to the **Connect** command.
- Fixed several crashing bugs that could occur when the image was near the edge of the screen.
- Fixed a problem where the **Run Date** column would display bad characters when Korean was the active language.
- Fixed a bug that caused bug submission to fail sometimes.
- Fixed a bug that sometimes caused times to be offset several hours in script logs.
- Fixed an exception that could occur when viewing logs on Linux.
- Fixed a bug that caused the Viewer window capture buttons to be disabled in capture mode.
- Fixed a bug that caused the Run Window buttons to be disabled.
- Fixed a bug that caused in-script text platform changes to be overlooked.
- Fixed a problem where Text Commands created using the Generation Tools did not properly reference the Text Style from the Text Platform.
- Fixed a problem where **Scaling "Fixes"** applied by the **Image Doctor** would not persist. Using the **Image Doctor** now sets the Image's default scale as shown in the **Image Editor**.
- Fixed a problem where Adding a Representation using the Original Location would throw an exception.
- Fixed a problem where Adding a Representation would not properly switch to the new folder if "Remember Save Image Folder" was disabled.
- Fixed a problem with some toolbar icons on Retina display Mac OS X machines.

- Fixed a problem displaying the "Add..." button on the Connection List on CentOS.
 - Fixed a problem when trying to connect to a RealVNC device from the command-line. This could happen before the RealVNC initialization had been completed.
-

Release v12.30

(15-July-2013)

Highlights:

- New in-line menus for Linux.
 - Several improvements for International support
 - Fixed several stability and usability issues.
-

Interface:

- Added the ability to sort Scripts Results by each of the columns.
- Added a Preference to disable having the Capture Image panel remember your last location.
- The menus on Linux are now displayed within the application windows by default. You can revert to the old behavior by issuing this command:

```
defaults write Eggplant NSMenuItemStyle  
NSNextStepInterfaceStyle
```

Scripting:

- **ReadText()** will now throw an error when no text is found at an image location or at a specified coordinate. This matches the behavior when a rectangle is specified.
- **ReadText()** will now log it's results for an image location or at a specified coordinate. This matches the behavior when a rectangle is specified.
- All script and image references should now properly normalize their Unicode values. This should allow for scripts and images to be found regardless of how the title is entered.
- Improved the behavior of the **Write** command to be more intuitive

- after an earlier **Read** command.
 - Fixed a problem where the Scale attribute was always returned as 1.0 for **FoundImageInfo()**.
 - **ScrollWheelDown()** with no parameter will now scroll down by one (instead of up by one)
 - **ScrollWheelDown()** and **ScrollWheelUp()** will now only allow scrolling by limited values to prevent locking up eggPlant and the VNC server.
-

Mobile:

- Added the following attributes for the **ConnectionInfo()** for USB connected Android devices:
 - **AndroidDevice** will be TRUE
 - **DeviceSerialNumber** will be set to the Serial Number of the device.
 - Fixed a problem with License Errors when connecting to RealVNC Mobile servers.
 - Improved the Viewer window to not shrink when mobile devices are rotated.
 - Fixed a problem where eggPlant would terminate the VMLite server on Android devices.
 - Fixed a problem launching on Mac with RealVNC Mobile keys.
-

Bug Fixes / Tweaks:

- Fixed a crashing problem when using Image Doctor/Image Search on some images that became entirely transparent.
- Fixed a problem with eggPlant hanging on Ubuntu 12 during startup.
- Fixed an occasional problem launching multiple simultaneous copies of eggPlant on Windows.
- Fixed a problem launching on Windows with some international date settings.
- Fixed a bug when creating a new suite with the same name as an old suite.
- Fixed a problem with AutoDoctor not engaging for Execution/Command-Line Only licenses
- Fixed a problem where opening an Image would re-open previously closed tabs in a Suite.

- Fixed a problem in **SendMail** when "Content-Type" was set.
 - Fixed a problem where the Image Doctor would scale images to an unreasonably small size resulting in false image matches.
 - Fixed a problem with some images not scaling properly.
 - Fixed an occasional crashing problem when closing script tabs.
 - Fixed a problem when executing selections of code that hadn't been reformatted. This could lead to various Range exceptions or crashes.
 - Improved memory usage to prevent numerous out of memory conditions.
 - Fixed a problem with the Auto-Doctor hanging during certain scaling searches.
 - Fixed a problem displaying context menus after certain screen resizes.
-

Release v12.21

(19-May-2013)

Highlights:

- Added the ability to connect directly to Android devices over USB.
 - Improved the stability of VNC connections to VMLite Servers on Android.
 - Fixed several bugs.
-

Android Support:

- Android devices with a USB connection to the eggPlant machine will now automatically be available in the drop down when adding a VNC connection.
 - When a USB connected VMLite Server stops unexpectedly on Android it will automatically be restarted by eggPlant.
 - Android devices can be specified in a **Connect** command by specifying their Device ID.
 - Android devices can be specified from the Command-Line by specifying their Device ID for the **-host** argument.
-

Scripting:

- Added the **LicenseInfo()** function, which returns information about the license your eggPlant was started with.
 - Added the **-MaxWaitForLicense** argument. When you run eggPlant from command line, this argument specifies the number of seconds eggPlant should wait for a license to become available, if necessary.
-

Bug Fixes / Tweaks:

- Fixed numerous of out of resource problems on Windows that could manifest during long runs or when many windows were open.
 - Fine-tuned the behavior of image "popovers" to be less intrusive.
 - Fixed a bug that could cause scripts to scroll upon formatting changes
 - Fixed a bug that caused image-property overrides to be disregarded in scripts
 - Fixed a bug that could cause the eggPlant splash screen to obscure the license panel on Windows
 - Fixed inordinate memory growth when resizing the capture area on Windows
 - Fixed a schedule bug whereby running a selection of a script and then adding that script to a schedule would cause the selection (not the whole script) to run as part of the schedule.
 - Fixed a bug that could cause windows to open in unexpected positions upon launch.
 - Fixed a bug that prevented Asian image names from being displayed properly in the image list.
 - Fixed an unexpected change in the SenseTalk date format. (Dates that appeared in *month/day/year* format were being returned as *year-month-day time*. The former format has been restored.)
 - Improved the appearance of tabs showing long resource names on Windows.
-

Release v12.20

(25-April-2013)

Highlights:

- Added a new **Scaling Image Search** to search for images at different

sizes.

- Scaled images can be automatically found based on changes in SUT screen size.
- The new **Image Browser** allows viewing different images within the image tab.
- Added **Image Popovers**, which show images as you hover over image names in your suite and scripts.
- Added the ability to show which images are/are not used in your suite via the **Image Usage**
- Suites now provide a **Resource Preview**, so you can briefly view resources without opening additional tabs.

Image Scaling:

- You can now add a *Scale* property to an image search, to indicate a scale multiplier (or list or range of scale multipliers) if you expect an image to appear at a different size than was originally captured.

```
Click (name: "appIcon", scale: 2.0) // Searches for  
appIcon at 2 times its original size  
Click (name: "appIcon", scale: .5 to 1 by .1) //  
Searches for appIcon at half size up to full size, in .  
1 increments
```

- You can use the *DefaultScale* property in a script, to set the scale value for all of your images. (Setting the scale value on an individual image overrides the *DefaultScale*.)

```
Set the DefaultScale to (.5, .75, .1) // Every image  
will be searched for in those three sizes
```

- You can use the *Scale: Auto* property in a script, to set the scale value based on the size of the current SUT relative to the size of the original SUT.

```
Click (name: "appIcon", scale: "Auto") // Searches for  
appIcon based on changes in SUT resolution
```

User Interface:

Image Usage

- In the contextual menu in the Image list in your suite, you can choose *Show Image Usage* to find out which images are used in scripts. Images that are used in scripts are shown in green. Unused images are shown in red.
- **Images considered "used"**: An image is considered used if its name appears as an uncommented string in a saved script in the same suite.
- **Images *not* considered "used"**: An image is *not* recognized as used if it is only used by scripts in other suites, it only appears in comments, it is only mentioned as a variable or constructed string, or the mention of it is not saved.
- **Refreshing Used Images.** Used images are not updated dynamically. As your image usage changes, choose *Show Image Usage* again to update the image list.

Image Browser

- In the new image browser at the bottom of the image tab, you can now see the siblings of your selected image. If you select one of these images, it is shown on the same image tab. If you select another image in the image list, a new image tab opens.
- **Image folders/collections:** You can navigate in and out of folders in the image browser by double-clicking folders to enter, and clicking the "Up" button to exit the current folder.

Run Window

Scripting:

- **JSONformat():** This function takes a SenseTalk property list and returns it in proper JSON format.


```
put contentValues's JSONFormat into outfile
```
- **JSONvalue():** This function takes a JSON string and returns a SenseTalk property list.


```
put JSONValue(file "/tmp/myJSON.txt") into
fileContents
```
- Added a **XMLRPCValue()** function to parse text in XMLRPC format (as generated by the XMLRPCFormat function) and return an equivalent SenseTalk value.

- Added a **monthName()** function to give the name of the month of any date, or of the current date if no parameter is given. An adjective may be used to request long, abbreviated, and short forms of the name.

```
put the monthName
put the abbreviated monthName of "7/4/1776"
```

- Added a **weekDayName()** function to give the name of the month of any date, or of the current date if no parameter is given. An adjective may be used to request long, abbreviated, and short forms of the name.

```
put weekDayName("7/4/1776")
put the short weekDayName
```

- Added a **keyForValue()** function to return the key corresponding to a particular value in a property list, or empty if the property list does not contain that value. If more than one property is equal to the specified value, the key to one of these values is returned

```
set myPets to (cat:"Yellow", dog:"Black")
put keyForValue(myPets, "Black") -- dog
put myPets.keyForValue("Yellow") -- cat
```

- Added an **allKeysForValue()** function to return a list of all keys in a propertyList that have a particular value, in no particular order. If the property list does not contain that value an empty list is returned.
- Fixed a small parsing issue that prevented calling a function on a number by using the '.' syntax. This can now be done, provided the function name comes directly after the "":

```
put 65.numToChar -- A
put -42.numberWords -- negative forty-two
```

- String literals prefixed by the '@' sign may now include backslash escape sequences.
- Added a **cube root** function.
- Added **is contained by** as a synonym for the **is in** operator.
- Added **multiplied by** as a synonym for the **times** operator.
- Added **counter** as a synonym for the **repeatIndex** function.
- Added **is past** as a synonym for the **greater than** operator.
- Allow the word **given** to be used between a handler name and its formal parameters.

to calculateArea given length, width

- Allow the word **position** or **location** to be used after **before** or **after** in a **range** or **offset** expression.
the range of "money" in accounts before position
lastFoundPos
 - Fixed a bug with **the duplicatePropertyKeyMode** -- when set to "list" mode it generated incorrect values whenever a key appeared more than two times in a property list expression. It now correctly generates a list of all of the supplied values.
 - Updated the STTreeNode XModule to version 3.1. Tree nodes can now be deleted using a node expression (previously only item expressions worked with the delete command for deleting tree nodes):
delete node "product/description" of order
-

Bug Fixes / Tweaks:

- Improved start-up performance on Windows operating systems.
- Improved performance of reading large data files during script execution.
- Improved Japanese text input on Windows operating systems.
- Fixed a problem using the "File->Open Script" that could lead to a crash on Windows.
- Fixed a significant memory leak caused by executing scripts that were not open in the interface. This leak would eventually lead to a crash.
- Fixed a problem opening suites that had unwritable scripts or folders.
- Initial suite display performance improved to create a smoother loading experience.
- Fixed a problem where using "Save All" could cause other scripts to appear empty.
- Fixed a problem where the panel alerting of external changes to a script would come up in an awkward screen location. These alerts now show up within the script editor.
- Fixed an error with "Duplicate TestCase" when running selections or running commands from EP:Drive.
- Fixed a problem where the Save Image Sheet wouldn't update to reflect that a new script or suite was selected.
- Fixed problems bringing up the proper line after an error when

- running a selection.
 - Fixed a number of problems setting or removing breakpoints that were in handlers.
 - Fixed problems with breakpoint display with different font settings.
 - Fixed a problem with right clicking on breakpoints on Mac.
 - Fixed a problem with the script insertion point not always moving when generating commands.
 - Bold formatting is now properly applied when scripts first opened.
 - The run window will now remember it's placement between launches.
 - Added a warning when setting **the finalSuites** or **the InitialSuites** to an invalid location.
 - Added a warning instead of an error when trying to use an invalid language for an OCR operation.
 - The **default** color will now work with Android VNC servers.
 - Fixed a problem where calling a non-existent script from the command line would generate bogus results files.
 - Fixed a problem where using "Reveal In Explorer" on a script would take you to the default suite directory.
 - Fixed an exception when pressing the Windows/Logo Key on the TypeText panel.
 - Fixed an occasional problem where new scripts didn't allow renaming.
 - OCR text once again highlights in green when it is found during a script run.
 - Removed second Quit menu item on Windows and Linux.
-

Release v12.11

(19-February-2013)

Highlights:

- Made significant memory and performance improvements on Windows and Linux.
 - Added device-button support for VMLite server on Android.
 - Resolved a number of issues concerning breakpoints.
 - Improved support for entering Japanese into scripts on Windows.
-

Scripting:

- Fixed a crash that could occur with **pass message and continue** and also fixed it to properly set the result when a value is returned by the passed message.
 - List expressions can now be written using square brackets [] instead of parentheses () or curly braces { }. This allows most JSON or JavaScript literal expressions to be evaluated by the value() function, or to be copied and pasted directly into a SenseTalk script and used as a valid expression. Note that there are still some incompatibilities, however, most notably the use of \ escape codes in JavaScript strings which are treated literally by SenseTalk.
 - **The numberFormat** property can now be set to "Words" to display numbers formatted as words. To include digits after the decimal point, "words" can be followed by a decimal point and a combination of "0"s and "#s as in other numberFormats:


```

set the numberFormat to "WORDS"
set the numberFormat to "Words.00###"

```
 - Fixed an inconsistency with numeric comparison using the **greater than** or **less than** operators that would sometimes report one number as being greater than or less than another very close value even though the **equals** operator would report them as being the same (due to the difference between them being less than 0.00000000001). Numbers that are so close in value that they compare as equal will no longer also compare as different.
 - Setting **the defaultStringEncoding** to an invalid value will now throw an appropriate exception.
 - The word **each** is now optional in some **repeat** statements:


```

repeat for x in 3..12 -- same as repeat for each x
in 3..12
repeat with term in words 3 to 12 of phrase

```
 - A very abbreviated **repeat** statement that omits the word **in** or **of** (and even the word **with**) is now available when working with an iterator (such as a list) using the default **it** as the repeat variable:


```

repeat with 3..12 -- same as repeat with each in
3..12
repeat with (dog, cat, hamster) -- repeat with
each item in
repeat (red, green, blue) --

```
 - Added a variation of the **repeat until** and **repeat while** statements that will execute the statements within the repeat loop at least once *before* checking the condition:


```

repeat at least once and continue until
<condition>
repeat one time and until <condition>
repeat one time and continue while <condition>
repeat at least once while <condition>

```
-

Bug Fixes / Tweaks:

- Fixed a number of crashes related to memory problems on Windows and Linux.
- Added *PowerButton*, *VolumeUpButton*, and *VolumeDownButton* support when connected to a VMLite server.
- Fixed a problem that could occur when you ran a schedule that had been closed and reopened.
- Fixed an exception that could occur when you used the manual Image Doctor
- Fixed a bug that caused breakpoints to fail in a recently re-named script or handler.
- Tweaked the image list to show newly-created images while a search filter is applied.
- Fixed a bug that caused the Set/Remove Breakpoint menu item to fail.
- Fixed a bug that prevented image collections from being named after their original image by default.
- Improved the functionality of the color picker on Windows.
- Tweaked the search filter to include script results.
- Improved the look and feel of dragging resources into a script on Windows and Linux.
- Visual feedback while renaming Scripts, Image, or Schedules on Windows has been made more clear.
- Fixed a problem where eggPlant would not fully uninstall when installing a new version.

Release v12.1

(18-January-2013)

Highlights:

- Redesigned and improved Team License Server application (formerly called Floating License Server).
 - Korean text is now fully supported, via the TypeHangul command, in scripts, and in resource names.
 - Licenses are now platform agnostic, not tied to a specific operating system (Mac, Windows, or Linux).
-

Licensing Improvements:

- The former FloatingLicenseServer application has been redesigned with additional functionality, and renamed as the **TeamLicenseServer**.

The TeamLicenseServer interface is a single window with two areas. The top part of the window provides information about the Team License Server itself including the name and number of its own license, the host id of the machine it is running on, and the server status.

The lower part of the window shows a continuously-updated list of all of the eggPlant instances that are currently running using team licenses that are linked to this server. Information given about each eggPlant instance includes the license number, the maximum number of users for that license, the host machine and user account where eggPlant is running, and the length of time (in hours and minutes) that eggPlant has been running on that machine.

A Terminate button can be used to remotely abort a selected running instance of eggPlant. This will cause the running eggPlant program to terminate immediately without warning. Only eggPlant version 12.1 and later will respond to the terminate message. Earlier versions of eggPlant can not be terminated remotely.

The TeamLicenseServer daemon process can now be set to run as a Windows service using the **InstallTeamLicenseService** batch file in the eggPlant install directory.

- Licenses are no longer issued for a specific operating system platform (Mac, Windows, or Linux). This simplifies licensing. Node-locked licenses are still valid for a specific machine only but the platform doesn't matter when generating a key. Team (floating) licenses can be used on any machine (regardless of platform) within reach of the Team License Server.
- Node-locked licenses may now optionally include support for running eggPlant in Drive mode.

Korean Text Support:

- Support for using Korean text has been added or enhanced in the following areas:

The script editor now implements entering Korean text through the IME on

Windows. This support extends to other text entry fields as well, including the names of scripts, images, and SUTs.

The TypeText command can type all asian characters on Android SUT devices running the VMLite VNC server.

The new TypeHangul command breaks Korean characters down into jamos and types them on the SUT.

For TypeHangul to work, a Korean IME must be active on the SUT. If the Windows Hangul/English IME toggle is visible on the SUT, TypeHangul will toggle between Hangul and English as necessary; otherwise, your scripts must keep track of the current state of the IME and perform the actions to toggle as needed.

Korean text is now properly displayed in the top half of the Run window, and in the Name column of the Connection List panel.

User Interface:

- A Script Info button (blue circle with the letter "i") has been added to the toolbar in a Script tab. This button will show or hide a number of items of information above the script, including: Length (this will update as the script is edited to show the length of the script); Modified (the date and time the script was last saved); Created (date and time the script was created); and Description (an editable field for entering any descriptive information about the script, which will be saved with the suite).
- The Color Picker tool (Magnifying Glass) on the Color Panel now works within eggPlant on all platforms.

Scripting:

- Added **push**, **pop**, and **pull** commands to facilitate use of lists as stacks and queues. The **push** command adds items to a list, and the **pop** and **pull** retrieve values from a list while also removing those items from the list.

The **push** command is simply a synonym for the **insert** command. In its most basic form it simply adds an item to the end of a list:

```
push 7 onto myStack
```

The **pop** or **pull** commands can then be used to retrieve values from the end or beginning of a list, respectively. To use a list as a LIFO (last in, first out) stack, simply **push** values onto the stack and then **pop** them off again in the reverse order:

```
push "A" onto myStack
push "B" onto myStack
pop myStack into x -- x is now "B" and "B" is
deleted from myStack
pop myStack into y -- y is now "A" and "A" is
deleted from myStack
```

To use a list as a FIFO (first in, first out) queue, **push** values onto end of the queue and then **pull** them from the front of the queue in the same order that they were added:

```
set waitingList to an empty list
push "A" onto waitingList
push "B" onto waitingList
pull from waitingList into x -- x is now "A"
pull from waitingList into y -- y is now "B"
```

The **pop** and **pull** commands can also be used to obtain values from the middle of a list, from a nested list, or to obtain multiple values at once. If a destination container is not specified, values removed from the list are put into **it**:

```
set source to "a".. "z"
pull item 14 of source into x -- x is now "n"
push (A,B,C) into item 13 of source -- item 13 is
now (m,A,B,C)
pop from nested item 13 of source -- sets IT to C,
item 13 to (m,A,B)
pop the last three items of source into tail --
tail is now (x,y,z)
pop items (2,5,1,4) of source into z -- z is now
(b,e,a,d)
```

- Updated the STTreeNode XModule to version 3.0. This release improves the functionality of **node** (XPath) expressions, fixing a problem on recent versions of Mac OS X and (optionally) correcting node expressions to work more consistently with the XPath standard, and consistently across platforms. Specific changes include:

Added the **StandardNodeExpressions** global property, which can be set to true to get the new (standard) behavior on all platforms (this will primarily only affect the behavior on Mac, as the Windows and Linux implementations were

already working in the standard way). Like all SenseTalk global properties this can be set in the defaults by setting the `STStandardNodeExpressions` default to true. By default this property is false, to maintain backward compatibility for older scripts that may rely on the previous behavior. It is recommended to set it to true for all new scripts that use node expressions.

Added **as document tree** as a more convenient way to call the `DocumentTreeFromXML()` function. If the `standardNodeExpressions` is left turned off, using "as document tree" instead of "as tree" for parsing XML into a tree will ensure standard behavior of node expressions for that tree.

Fixed a bug on Mac that would throw an XQuery Error 6 exception for any node expression beginning with "/" or "/" unless the tree was created with the `DocumentTreeFromXML()` function. These node expressions will now behave the same on all systems regardless of the `standardNodeExpressions` setting.

Bug Fixes / Tweaks:

- Improved aborting of a script to abort immediately during an image search rather than waiting until the search completes.
- Fixed a problem with copying and pasting text between eggPlant and other applications on Windows systems.
- Fixed a problem with evaluation of **node** (Xpath) expressions in XML trees being different on Mac and Windows. To avoid breaking scripts this is left unchanged by default, but compatibility can be achieved by setting **the StandardNodeExpressions** to true.
- Fixed a problem that would throw an XQueryError exception on Mac for **node** (Xpath) expressions beginning with "/" or "/"
- Fixed a problem with pasting into the License Installer panel on Windows.
- Fixed an issue that prevented selecting multiple scripts' results at once in the Results section on the left side of a suite window, or selecting multiple executions in the Results tab.
- Fixed an issue with the "Manual Doctor" menu item being blank on Windows.
- Fixed a problem that would fail to warn the user when saving a script in a write-protected location. A warning is now provided, and if the script was being closed it remains open so work is not lost.
- For a suite located in a read-only directory, the following menu items are now disabled: Schedule-->New; Images-->New Folder; and Images-->Add Image.
- More informative error messages are given when an incorrect or invalid license key is pasted in the key field. Also, extra space before or after the key is ignored, making entering a license key more friendly.
- Fixed a problem that would throw an exception after dragging a script or image name into a script on Windows.

- Added a preference option, `EditCaptureAreaInBackground`. When enabled, this option will allow the capture area in a Remote Screen window to be moved or resized even when that window is out of focus (not the main window).
- Fixed a problem that would create an unnecessary set of nested directories in a "Users" directory within the Default Suite Directory on Windows.
- Provide an appropriate error message when trying to add a script to a suite without the needed permissions (previously this would throw a meaningless exception about a nil Font).
- Creating a new script now properly displays the suite window if it had been minimized on Mac.
- Improved recognition of duplicate licenses being run on the same machine on Windows.
- Fixed a problem that would drastically slow down script execution in a linked group of suites in which each suite included all of the others as Helpers. The slowdown was noticeable with 5 or 6 interlinked suites, problematic with 7, and unusable with 8 or more.
- Fixed a problem that would progressively obscure the top of a script with unsaved changes when repeatedly clicking the Cancel button after trying to close the script.
- Switched the icon used for the Comment button in the script editor when no text is selected to make its function clearer.
- Fixed a problem with being unable to reopen a schedule that had been closed, on Mac OS X 10.7 systems.
- Fixed a bug that made the current SUT connection initially unavailable in a post-run script.
- Fixed a bug that would crash the Team License Server when right-clicking to get a context menu in the License Installer panel (when first licensing the app) on Windows.
- Fixed a problem that could show the wrong value in the User name field when adding or editing a connection.
- Fixed some problems with closing a schedule tab while the schedule is running. This would stop the schedule from running, and reopening the tab for that schedule would not show the information for the previous run.
- Fixed a problem that would display "ERROR: newFrame not cleared in RunController" when running scripts from the command line in a suite with a pre-run script enabled.
- Fixed a problem that could leave eggPlant Drive on Windows in a confused state when a client application was killed.
- Installs on Windows will now honor the `\S` flag to perform a silent install without any alerts being displayed.
- Improved cleanup of ABBYY keys in the registry when eggPlant is uninstalled on Windows.
- The eggPlant startup file can now be pinned to the task bar in Windows 7.

- Crashes that occur on Windows will now capture backtrace information that will greatly improve our ability to fix such problems in the future.
-

Release v12.02

(12-December-2012)

Highlights:

- When a tab is selected in a suite window, the corresponding item is now highlighted in the list on the left.
 - Fixed a number of bugs.
-

Bug Fixes / Tweaks:

- Removed the "Quit After Next Script Run" option when closing the last window on Windows, which would cause eggPlant to hang.
- Fixed a problem that could prevent another suite from being opened while a script is running, paused, or previously run in some situations.
- Eliminated some error messages while installing on Ubuntu 11.04+.
- When submitting a bug, question, or feature request, the pre-supplied description text now matches the type of request.
- Fields in the bug report / question / feature request window are now cleared only when the panel is closed or submitted.
- After creating and naming a new script, the focus is now set to the script editor, ready for user input.
- Fixed a problem in which a modified script that had been previously changed and closed, could be closed without the user being prompted to save their changes.
- Fixed a problem with the state and arrangement of windows not always being remembered and restored properly.
- Fixed a problem with running a script from the command line that would require at least a partial path including the script's suite.
- Fixed a problem that would sometimes report an exception when quitting with unsaved changes and choosing to save all scripts.
- Changed the title of the "Settings" button to "Settings..." to make it clear that it is a button that opens a tab.
- Fixed a crash when an image tab was closed while performing an Image Doctor search.
- The Run Selection button is now disabled when a script is running.

- Fixed a problem with the F8 - F12 items on the TypeText pulldown menu generating digits rather than the correct code.
 - Improved the behavior when attempting to use the CaptureScreen or TraceScreen command to save an image to an invalid location (with insufficient privileges or unreachable path). An appropriate message will now be reported.
 - Fixed a problem with some exception logs that would display incorrect information in the results tab.
 - Fixed an issue that would sometimes fail to select the name of a newly-created script for ease of naming it.
 - Fixed a crash that could occur when running a script from another using RunWithNewResults.
 - Fixed a crash that could occur on Windows or Linux when searching in a suite containing many images.
 - Fixed a crash that could occur on Mac OS 10.6 when closing a suite window.
 - Fixed a problem that could throw an exception when clicking the Image Doctor button.
 - Fixed a crash that could occur when changing platform types or styles on the Text Preferences pane.
 - Fixed an issue with the suite window interface occasionally appearing to "freeze" on Windows.
 - Fixed a problem in which the detailed result log displayed on a result tab could be for a different (previously selected) run date than the run date currently selected in the upper list in some situations.
 - Fixed a crash that could occur sometimes on Windows and Linux when typing in a panel after closing a suite.
 - Fixed an exception that could occur when dragging an image into a script in a different suite.
 - Fixed an issue in which the same script could be open in multiple tabs in the same window, sometimes leading to a crash or hang.
-

Release v12.01

(11-November-2012)

Highlights:

- The Copy command can now be used in a Remote Screen window in Capture mode to copy the rectangle coordinates to the clipboard.
- The Open Recent menu now works on Windows and Linux.
- All open windows are now listed on the Window menu.
- Added a Save As... item on the File menu to create a copy of the current

script within the suite.

Bug Fixes / Tweaks:

- Fixed a problem accessing the GUI for the Floating License Servers.
- Fixed a problem that would erroneously display a panel warning of unsaved changes when closing a suite window (Windows).
- Fixed a problem that would throw an exception when accessing the Run Options preferences (Windows) if you had installed over a previous version of eggPlant.
- Fixed a problem with suite windows which were open on a second monitor being moved to the main screen when eggPlant is relaunched.
- Fixed a problem in which using Command-w (or Control-w) to close a script tab would not warn of unsaved changes.
- Fixed a problem with screen colors being displayed incorrectly (blue) when connecting to certain RealVNC mobile servers (Mac).
- Added the Documentation item to the Help menu (Mac).
- Fixed a problem that would give the message "Command string selected a bearer that could not be loaded" when connecting to some mobile devices (Windows).
- Fixed the Cancel button when searching with the Image Doctor.
- Fixed an NSInvalidArgumentException that could occur when using the Image Doctor.
- Fixed the Show Resource item on the Edit menu.
- Fixed the Comment/Uncomment item on the Edit menu.
- Fixed the Current Script item on the Window menu.
- Fixed the Open Recent submenu on the File menu to show a list of recently opened suites (Windows and Linux).
- Fixed a problem with the Show Script button on the alert panel for a script failure, that would open the documentation in a browser.
- Fixed an issue that a script name did not stay selected after renaming it.
- Fixed an issue with a collapsed left-side navigation area popping open again when the window is resized.
- Fixed an issue with long script names being truncated in the script list (Windows and Linux).
- Fixed a display issue with editing of long script names in the script list (Windows and Linux).
- Fixed a problem accessing the Context Menu in the Navigation Area (Mac 10.6).

Release v12.00

(30-October-2012)

Highlights:

- Redesigned Suite Window with all Scripts, Images, Results, and Schedules shown in tabs within the window.
- Multiple Schedules can now be created in each Suite.
- New LogSuccess and BeginTestCase / EndTestCase commands for enhanced tracking of validation points.
- Script results and statistics are now available in XML/JUnit format.
- Added Pre- and Post-run scripts for each suite, that are run before and after every script run that would generate results.
- Added a Template script for each suite, that is copied for newly created scripts in that suite.
- Documentation is now in HTML format online, for rapid updates and ongoing enhancement (<http://docs.testplant.com>).

Compatibility

- **TypeText with backslash codes. (e.g.: \n, \t, \\)**
The ability to use backslash codes in a TypeText command is now disabled by default. If you were using the codes or the double-backslash then we recommend modifying your scripts to use the predefined constants, Example:
`TypeText "This is a backslash:" & TAB & BACKSLASH & RETURN`
produces
***This is a backslash: ***

If needed, the prior behavior can be enabled through any of the following mechanisms:

Within the preferences: "Enable Backslash Escape Codes" checkbox in the Preferences->Run Option->Keyboard.

Within a script: using "set the enableBackslashEscapeCodes to yes"

From the command line: using "--enableBackslashEscapeCodes YES"

- **Command Line Return Value**

The default behavior is now for eggPlant to report the number of failures, so 0 indicates all scripts succeeded. Attempting to run a non-existent script is now counted as a failure. If eggPlant is unable to run (such as for no license) a value of 127 is returned. To request the prior default behavior (reporting the number of successful script runs) pass - ReportFailures NO on the command line.

Consolidated Suite Interface:

Each Suite now has just one window which displays any scripts being edited in that suite as well as the images, results, schedules and other information for that suite in a tabbed interface.

- **Left-Side Lists:**

On the left side, each Suite window displays lists of all of the scripts, images, results, and schedules in the suite. Double-clicking any item in these lists will open a tab displaying that script, image, result history, or schedule. In addition, a Settings button will open a tab displaying settings and helpers for the suite.

The relative sizes of the lists on the left side of the suite window can be changed by dragging the section titles up or down.

Each section has an action context menu for selecting appropriate actions in that section. On Mac, the action menu for a section is also available as a pull-down menu (click the gear icon in the section title).

Single-clicking an already-selected item in the Scripts, Images, or Schedules lists will edit the name of that item (Mac only -- on other platforms, use the Rename item from the action menu).

- **Unified Search:**

The search field above the lists filters all of the lists by name and/or (where appropriate) by contents. For a script, the content is the script itself. For an image the content is its description, and for a schedule the

content is the names of the scripts in the schedule. The magnifying glass icon presents a menu to select the type of search to use, as well as a history of recent search terms (press return after entering a search term to add it to the history list).

- **Tabbed Interface:**

Tabs can be opened by double-clicking an item in one of the lists, by selecting the "Open" item from one of the action menus, or by various other shortcuts or actions that will open a tab automatically.

Tabs can be dragged to change their order and may be closed by clicking the "x" icon in each tab.

Drag any tab out of a suite to create a new window, or drag a tab into the tab bar of another window for the same suite (you can't drag a tab to a different suite).

All open tabs in the suite window are remembered when the suite is closed and will be restored when the suite is next opened.

- **Scripts:**

Each script tab allows editing of a script and also (optionally) of a description for the script. To access the description field, drag the top edge of the script edit area down to reveal the description field. Attempting to close a script tab for a script with unsaved changes will display a warning message above the script (unless the script is also open in another window). The warning message provides options to cancel the message and continue editing, or to save or discard the changes and close the tab.

- **Images:**

The action menu for Images includes a Create Collection option which will immediately create a new image collection containing all of the selected images.

- **Results:**

The Results list shows the names of all scripts for which results are

present in the suite. Opening a tab for a given script's results shows a list of every individual execution of that script, identified by the run date and time.

- Schedules:

A suite may now include multiple schedules (lists of scripts that can be run as a batch). Each schedule is given a name when it is created. If the **runscript** command is used to run a suite from the command line, it runs the first schedule for that suite. Or a specific schedule can be run by including a slash and the schedule name at the end of the suite path. For example, this command would run the first schedule in MySuite.suite:

```
/Applications/Eggplant.app/runscript /path/to/  
MySuite.suite
```

This command would run the schedule named "SmokeTest" in the same suite:

```
/Applications/Eggplant.app/runscript /path/to/  
MySuite.suite/SmokeTest
```

- Settings:

The Settings tab incorporates the contents of both the previous Helpers tab and Settings tab, plus some new settings. Helper suites for the script can now be disabled or enabled by clicking a checkbox for each suite.

- Also:

A status field at the bottom of the suite window displays the name of the currently running script.

The Info drawer has been eliminated, and the associated information incorporated into the appropriate tabs.

Pre-run and Post-run Scripts:

- A new option allows specifying a pre-run script that will be executed before each script run, and a post-run script that will be executed after

each script run. Select which scripts to use on the Settings tab for a suite, which also has checkboxes for turning the execution of these scripts on and off.

The pre-run script will be called with three parameters. The first is the full path of the script that is about to be run. The second is a list of all of the parameters (if any) that will be passed to the script. The third is the full path to the log file that will be created during the script run.

The pre-run script has the ability to change the parameters that will be passed to the script. To do this, the pre-run script should return a property list with a "parameters" property whose value is a list of the parameters.

The post-run script is called after the script run has finished. It is called with two parameters. The first is the full path of the script that was run. The second is the scriptResult property list for the run, similar to the result returned by the RunWithNewResults command.

In addition to the parameters and return values described above, the pre- and post- scripts can interact with the script being run through a shared environment, including:

- connections -- open a connection in the pre- script and it will be the active connection when the script being run starts
- global and universal variables are common across all three scripts
- global properties are common across all three scripts, including the InitialSuites and the FinalSuites

New Script Template:

- A template script for new scripts may also be specified on the Settings tab for a suite. If a template script is chosen, each new script that is created in that suite will begin as a copy of that script.

Result File Format Changes:

- An XML file (LogFile.xml) containing summary information gathered from the TestCases in a script run is produced alongside the full text log (LogFile.txt) in the result folder for each script run. The LogFile.xml file follows the reporting format used by JUnit and similar unit testing tools,

to make it easy to integrate with other tools that can accept this format. A checkbox option is available on the General Preferences pane to disable production of this file.

- The information in the SuiteStatistics.csv file is now mirrored in XML format in a SuiteName_Statistics.xml file. This XML file is now the primary format used by eggPlant, with the CSV file still supplied for compatibility. Both are located in the Results directory.
- The information in the RunHistory.csv file is now mirrored in XML format in a RunHistory.xml file. The XML file is now the primary format used by eggPlant, with the CSV file still supplied for compatibility. Both files are found in the folders created for each script within the Results directory.
- The RunHistory.csv file and corresponding .success or .failure file in each run directory now have an additional column at the end containing the success count (number of LogSuccess commands executed) for each run.

Scripting:

- Added a **LogSuccess** command to create log entries for successes in the script. Successes are shown in green in the detailed log view in a Results tab, and a new column shows the Success count for each script run. In addition, the Successes count is also included in the various ScriptResults property lists, and as one of the statistics gathered by the BeginTestCase/EndTestCase commands. A script run that completes successfully but that doesn't explicitly call LogSuccess will be treated as having a success count of 1.
- Added **BeginTestCase** and **EndTestCase** commands to mark the beginning and end of any section of code within a script (or across scripts during a run) as a Test Case for purposes of gathering information. Each test case is given a name. A script run may include multiple test cases which may be sequential portions of a script or may overlap with one another in any way you choose. The information gathered for each test case includes the duration of that test case (in seconds) and the number of Errors, Successes, Warnings, and Exceptions that occur or are logged while that test case is open. Information about open test cases can be obtained at any time through the **OpenTestCases()** function. One test case is created automatically at the beginning of each run, with the name of the script that is being run. At the end of a run, any test cases that are still open are ended and reported automatically.

Example code:

```
    BeginTestCase "A" -- this marks the start of the
reporting period for case A
    -- other commands here
    EndTestCase "A" -- this ends the reporting period
for case A and logs the information that was gathered
    BeginTestCase "B" -- start recording statistics
for case B
    -- other commands here
    BeginTestCase "C" -- start recording statistics
for case C
    -- other commands here
    EndTestCase "B" -- close case B and report
statistics gathered since case B was opened
    -- other commands here
    -- if the run ends with TestCase C still open, it
will be ended and reported automatically
```

- Added support for buttons used by VMLite VNC server on Android:

- Home button: HomeKey
- Menu button: F2
- Back button: Escape
- Search button: F5
- Call/dial: F3
- Lock phone: F4
- Rotate Viewer Window: F6
- Navigate between apps Up/down/left/right: UpArrow, DownArrow, LeftArrow, RightArrow

Other Changes:

- The eggPlant documentation is now in HTML format, stored and updated regularly on the TestPlant web site at: <http://docs.testplant.com>.

- The eggPlant OCR functionality now includes support for Asian languages Chinese, Korean, and Japanese.
 - When a script run fails, the alert panel that is displayed now offers a choice of whether to display the script (previously, the script was always displayed).
 - Added a mechanism to detect when eggPlant fails to launch. On the next launch attempt it will now offer to move aside the user preferences, which may eliminate the problem that is preventing it from launching successfully.
 - Connection information can now be copied from the Connection List and pasted as text in a script or other text document.
-

Bug Fixes / Tweaks:

- When a **WaitForAll** command fails, it now reports exactly which image was not found (for both the ImageDoctor and in the log). Previously it always reported the first image as the one that was not found.
- Fixed a problem where **ColorAtPoint()** did not allow getting the colors at the far extremes of the screen. Note: The actual screen goes from (0,0) to (Server Width -1, Server Height - 1) but for simplicity/compatibility we allow (Server Width, Server Height) to mean the right edge and bottom edge of the screen respectively.
- Connections can now be dragged from the Connection List with the Alt key held down (on Windows) and dropped into a script to generate a Connect command with the full connection property list (previously this capability was available on Mac and Linux, but not on Windows).
- Fixed an occasional crash following a RunWithNewResults.
- Fixed an occasional random crash when double-clicking on Windows systems.
- Fixed an issue that menus would sometimes not appear in a window when it was first opened on Windows systems.
- Fixed a bug that would fail to transmit large quantities of data (such as a 50K file) over a socket connection on Windows.
- Improved behavior when attempting connection to a SUT fails and times out.
- Fixed an issue in which the Email and Subject fields on the bug report / support panel would be marked in red even after the user entered information in those fields.
- Fields in the bug report / support panel are now cleared when the panel

is brought up again.

- The bug report / support panel now allows attaching of crash logs on Windows systems.
 - Eliminated some spurious GLayoutManager warnings in the console log (Linux and Windows).
 - Fixed some issues with windows occasionally being "lost" (such as windows which were minimized on quitting being lost upon relaunching eggPlant) (Windows and Linux).
-

Release v11.33

(31-July-2012)

Bug Fixes / Tweaks:

- Fix a buffer overrun that could corrupt memory when loading a script.
 - Fix a problem with eggPlant trying to create a file/folder when it lacked permissions.
 - Memory Logging is no longer enabled by default
-

Release v11.32

(26-July-2012)

Highlights:

- Added Code Signing support for Mac OS X 10.8.
 - Significant memory and performance improvements.
-

Scripting:

- Improved performance of the Log area in the Run Window when the maximum buffer length is reached.
- Added friendlier syntax for accessing some of the folders identified by the specialFolderPath function:
 - put the home folder
 - put the temporary folder
 - put the documents folder
 - put the desktop folder
- A fancy (curly) apostrophe may now be used in place of a regular (straight) apostrophe if desired:

add one to sam's age -- regular apostrophe
add one to sam's age -- fancy apostrophe (whoopie!)

- Improved error reporting for syntax errors caused by unrecognized Unicode characters.
 - Added the ability on Mac to prevent the machine from going to sleep while a script is executing.
You can control this behavior with: **setOption "PreventSleep", NO**
-

Bug Fixes / Tweaks:

- Memory optimizations for scripts that create objects.
 - Major memory optimizations when running many suites which all list each other as Helpers.
 - Fixed a bug where Test Images would appear faded in the Image Viewer.
-

Release v11.31

(28-June-2012)

eggPlant Drive/Command Line:

- Added support for resolving symlinks when calling scripts from CLI.
 - Fixed a problem where an Alert Panel could halt execution if a script or suite can't be located.
 - Fixed a problem where the content-length didn't account for multi-byte characters in the response.
 - Fixed a problem where drive execute commands that did NOT generate a value for **the result** could still transmit data in **the result** property, this slowed down communication and created confusion.
-

Mobile:

- Fixed a problem where eggPlant could hang if it unexpectedly lost connection to an Android device.
 - Fixed a problem where the size reported by RemoteScreenSize() was not adjusted after a rotating a mobile device.
-

Scripting:

- Added the ability to control the default mobile encoding with `RealVNCDDefaultEncoding`.
You can control this with: **set the `RealVNCDDefaultEncoding` to ["JPEG", "JRLE", "ZRLE", "TRLE", "SLRLE" or "Raw"]**
 - The Log area in the Run Window can now optionally store its output into the Result folder.
You can control this with: **set the `SaveRunLogOutput` to [yes/no]**.
 - The Log area in the Run Window now has a maximum buffer length that it will display to limit memory growth.
 - Fixed a bug that would continue to report a **result** or **return** value in the AHDB for subsequent commands.
 - Fixed a bug that would cause a tree passed as a parameter to be incorrectly treated as text when comparing with another tree. This would result in incorrect comparison results in some cases.
 - Fixed an exception when trying to access a range of bytes beyond the length of a value.
 - Fixed a bug with **iso8601** date parsing. That code has been tightened up to:
 - only allow whitespace after a date/time value (previously it allowed any characters following the recognizable date/time)
 - require consistent delimiters within the date and within the time (only '2012-06-15' not '2012-06 15' or '2012-0615')
 - only allow '.' as the date delimiter when a full Y.M.D date is given (with two dots: '2012.06.15' not '2012.06' or '2012.')
 - Changed undocumented messages sent for commands like "copy file" from "copyfile" to "st_copy_file" to make them less likely to interfere with users' custom scripts:
 - `st_copy_file`, `st_move_file`, `st_accept_connections`,
`st_add_properties`,
`st_remove_properties`, `st_retain_properties`, and
`st_replace_properties`.
 - Added **missing value** as a constant, for future use.
-

Bug Fixes / Tweaks:

- Major memory optimizations on logging and output commands.
- Fixed a problem running eggPlant on a fresh install of Ubuntu 12.04.
- Fixed an exception when trying to name an Image without specifying a Suite to save it into.
- Removed unnecessary warnings about Unable to Load Suite for a custom object.

- Fixed a bug on Windows when deleting scripts.
 - Fixed a bug on Windows where eggPlant would create spurious Result folder hierarchies.
 - Fixed a bug on Windows with the Failed Connection Alert Panel not dismissing properly.
 - Fixed an exception when double clicking on an empty row in a Suite's script list.
 - Fixed a bug (introduced in 11.0) where the RunHistory.csv would have (null) entries for scripts that had never run.
 - eggPlant: Windows will now check for Licenses explicitly in C:\Library\TestPlant in addition to the current drive.
-

Release v11.3

(03-May-2012)

Highlights:

- New Mobile VNC Servers with broader mobile platform support.
- Added support for Mobile connections to eggPlant: Mac.
- Added support for Integration with HP's Quality Center.
- Enhanced **SendMail**, including encryption support for Gmail and Exchange servers.
- Upgraded OCR on LINUX for improved performance and stability.

Mobile:

- Support for Android devices with an installed Remote Control Service(RCS) and VNC Server.
Devices with a vendor signed RCS do not require root permission.
 - Support for USB connections on Windows and Linux.
 - Improved Performance and Stability for Mobile connections.
 - Added mobile device information available through the **ConnectionInfo()** function.
-

Command Line/Drive:

- Added **-type** as a command line argument to specify the connection type to use with **-host**. Valid arguments are "**mobile**", "**standard**", "**legacy**", or

"automatic" (default).

- When specifying **-ReportFailures** eggPlant will now exit with error code 1 if it encounters a problem before it begins to run a script, e.g.: Unable to license, unable to resolve script name. Previously it would report 0 since no individual scripts had actually failed.
-

User Interface:

- Added the ability to sort by the **Last Run** column in the Suite's Script list.
 - Preferences -> Mail, now allows you to specify the default **smtp_transport** and **port** that the **sendMail** command will use.
 - Windows now uses standard keyboard shortcuts for things like selection, move from word to word, move to end of line, etc.
-

Scripting:

- The **readTable** function will now take a 2 coordinates for a point or 4 coordinates for a rectangle directly (without needing to put them in a list first).
- The **SendMail** command now supports an **htmlBody** property. This property can be set in addition to the **body** property to provide HTML copy in addition to or instead of the plain text.
- The **SendMail** command now supports an **smtp_transport** property. This property can be set to SSL or TLS to send email using those encryption methods. The port will default to 465 or 587 in these cases. This allows sending to almost all SMTP servers including GMail and Exchange (with SMTP enabled).
- Added a new **post multipart** command to provide the ability to post multipart data to a server:

```
post multipart contentToSend to URL serverURL
```

In the example above, contentToSend can be a single text value or list of values to send including the contents of one or more files, or may be a property list or list of property lists to provide more control over the format in which the data is transmitted or supply additional information about the contents. The following example will upload a file to the server:

```
post multipart (filename:"/tmp/xyz", body:file "/tmp/xyz" as data) to URL serverURL
```

Specifying "as data" causes the file contents to be encoded in base64.

Including the filename is optional, but recommended when sending a file.

- Added a **multipartMIME()** function to produce a property list containing Body and Header information for the given content, as it would be sent by the **post multipart** command. This can be useful for troubleshooting a problem or for constructing an HTTP message as text. This function can be called with a single value, a list of values, or a property list or list of property lists. See the description of the post multipart command for full details.

The value returned by the multipartMIME function is a property list with a Body property whose value is a fully-formatted string in multipart-MIME format. The Header property is a property list containing appropriate headers to accompany the Body text.

```
put (multipartMIME("Secret Message" as data)
adding (request:"/")) as http
```

Bug Fixes / Tweaks:

- Fixed a crash with submitting a support request on Windows.
 - Fixed a bug on Windows deleting a newly created script.
 - LINUX version is no longer restricted to running on a single CPU.
 - Fixed a problem on Mac where the Remote Screen Window would stop redrawing.
 - Fixed a problem where the Remote Screen was sent to the back with the **show RemoteWindow** command.
 - Fixed a problem entering licenses into the "No License" panel of the Floating License Server.
 - Allow specifying folders relative to a user directory (~) and allow creating multiple levels of directories when setting the **GlobalResultFolder** or an individual suite's **Result Folder**.
-

Release v11.22

(12-March-2012)

Bug Fixes / Tweaks:

- Fixed a display problem on Mac OS X where the viewer window would cease to draw updates. (Scripts would continue to run but the local view was frozen).
- Fixed a problem where searching for a text string with spaces would only

work if the IgnoreSpaces flag was **ON**.

Release v11.21

(9-March-2012)

Scripting:

Because of the importance of maintaining script compatibility, we are setting the new OCR features of 11.2 to be opt-in choices. Scripts written in earlier versions of eggPlant will function as before with no modification.

- Added optional IgnoreSpaces property when doing a Find Text operation. OFF by default, but when enabled spacing will be ignored for text matching. Useful when the OCR sees interprets extra spaces or omits spacing in some cases.
 - When doing an image search you can optionally specify a single asterisk ("*") for the ValidCharacters property to have OCR only identify the characters in your search string.
-

Bug Fixes / Tweaks:

- Fixed a display bug in the Viewer Window when Ubuntu visual effects were turned on.
 - Fixed a problem with eggPlant recognizing that new updates are available in some environments.
-

Release v11.2

(2-March-2012)

Highlights:

- New **eggPlant Drive** mode allows eggPlant to be run in a service mode. (See **eggPlant Drive** Documentation for complete information)
- XML reading/writing support for Windows and Linux.
- Addition OCR Support for almost 200 languages.

User Interface:

- Added a setting to change (or disable) the Countdown Delay when the Manual Image Doctor engages.
- Improved the Image Doctor so that it will always run for script commands regardless of pending DO button actions (from the Ad-Hoc Do Box). Previously using the Do box would temporarily disable the Image Doctor.
- Allow setting a license key directly from the "No License Found" panel.

Scripting:

- Added a pause for the KeyDownDelay when auto-sending a shift key for a capital letter.
- Added support for the following mobile buttons on devices that support them:
PowerButton, DismissButton, BackButton, UnlockButton, EditButton
- When using OCR for an EveryImageLocation() search we now only report a given location once even if it's found with multiple matching modes.
- Added support to call ReadText() with two numbers (for a point) or four numbers (for a rectangle).
- Added IgnoreUnderscores property when doing a Find Text operation, ON by default because underscores are often overlooked by OCR.
- Added IgnoreSpaces property when doing a Find Text operation, ON by default because spaces are often mis-counted by the OCR engine.
- The XML and Tree functionality provided by the STTreeNode XModule is now available on Windows and Linux as well as on Mac.
- Added the ability to send and receive HTTP messages.
 - When a socket is used for communication using the HTTP protocol, SenseTalk provides support for directly reading and writing HTTP messages. To read a message, use one of these commands (or a variation):

`read an HTTP message from socket clientSock`

```
read from socket myHTTP for an HTTP request
read an HTTP response into theReply from
socket remoteService
```

The value read will be a property list with all of the information contained in the HTTP message that was read, including Method, Path, and Header properties, similar to this:

```
(Header:(Accept:"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "Accept-Encoding":"gzip, deflate", "Accept-Language":"en-us", Connection:"keep-alive", Host:"localhost:5991", "User-Agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/534.51.22 (KHTML, like Gecko) Version/5.1.1 Safari/534.51.22"), HTTPMessageType:"Request", Method:"GET", Path:"/", Version:"HTTP/1.1")
```

- Similarly, an HTTP message can be sent across a socket using the write command. To write an HTTP message from a property list, use the new **HTTPMessage()** function or **as http** operator to convert the property list to an HTTP message and use the write command to write that to the socket:

```
write HTTPMessage(status:200, body:htmlContents) to
socket webClient

write replyMsg as http to socket customerSocket
```

The HTTP message to be sent should be in the form of a property list that may include these properties:

HTTP Request Message:

Method: GET or POST (if omitted, GET will be assumed if no Body is supplied, and POST if a Body is supplied).

Request: path to the resource being requested on the server (required).

Version: if not supplied, HTTP/1.1 will be assumed.

Header: if given, must be a property list of header keys and values.

Body: the body of the message. The body may be text, or it may be a property list, in which case it will automatically be encoded in key/value pairs.

HTTP Response Message:

Version: if not supplied, HTTP/1.1 will be assumed.

Status: a numeric status value to send. If not specified, defaults to 200 (meaning OK).

Reason: text of the reason for the returned status; defaults to OK.

Body: the content being sent. For a web server, this would typically be the HTML code of a page. The body may be a property list, in which case it will automatically be encoded in key/value pairs.

- Added the ability to send and receive XML-RPC messages.
 - To send an XML-RPC request, the **post xmlrpc** command is used:

```
set myRequest to
(methodName:"system.listMethods")
post xmlrpc myRequest to url serverURL
```

The message being posted must be a property list with a `MethodName` property and optionally `Params` and `Header` properties:

MethodName: the method name being called (REQUIRED)

Params: an array of parameter values. Specify "as boolean", "as text", "as date", etc. when needed to ensure individual values are encoded as the correct type for the method being called.

Header: a property list of other header elements (optional)

User-Agent: if not specified in the `Header` property, the SenseTalk long version will be supplied: SenseTalk Release 1.53

The URL given in the `post` command may include a path to identify a particular service on the server. The response from the server will be stored in the variable `it`.

- To send an XML-RPC response, use one of the functions **XMLRPCResponse()** or **XMLRPCFault()** along with the **write** command:

```
write XMLRPCResponse(responseData) to socket
rpcClient
write XMLRPCFault(33, "bad mojo") to socket
```

rpcClient

- The **XMLRPCResponse()** function takes one or two parameters. The first parameter is the value being returned in the response and is required. It may be any type of value, including a list or property list. The second parameter is optional. If supplied it must be a property list containing any additional HTTP headers to be included in the message.
- The **XMLRPCFault()** function requires at least two parameters: an integer fault code, and a fault string. A third optional parameter, if it is included, must be a property list containing any additional HTTP headers to be included in the message.
- To receive an XML-RPC message on a socket, use the **read** command and specify **xmlrpc** or **xmlrpc message** (to read any type of XML-RPC message), **xmlrpc request** (will throw an exception if the message received is not a request), or **xmlrpc response** (will throw an exception if the message received is not a response or fault) as the type of value to read:

```
        read an xmlrpc message from socket
clientSocket into msgRecvd
        read an xmlrpc request from socket
clientSocket
```

The following keys are recognized: (XMLRPCRequest)

Request: The URI to post to

Host: The host posted to

URL: The URL posted to, including host and request URI

Either the Host or URL is REQUIRED. If Host is given with no Request, the request is set to "/".

User-Agent: if not specified, the SenseTalk long version will be supplied:
SenseTalk Release 1.53

Header: other header elements (optional)

Method: the method name being called (REQUIRED)

Params: an array of parameter values. Specify "as boolean", "as text", etc. when needed.

- Added **xmlrpcFormat()** function which returns a string representing any value in the format required for use within the body of an XML-RPC message.
- Added **XMLContentEncode()** function to convert text into a safe format for including in the content of an XML document by replacing any occurrences of the characters "<" , ">", "&", and single and double quotes with their respective XML entity values.
- Added **XMLContentDecode()** function to decode a string containing certain XML entity values, such as those generated by the XMLContentEncode() and return the original text.
- **The timeInputFormat** may now include a special "**iso8601**" option, which is included in the initial setting. When "iso8601" is included as one of the entries in the timeInputFormat list, SenseTalk can recognize dates and times in many formats that conform to the ISO 8601 international standard.

ISO 8601 date formats that are recognized begin with 4 digits for the year followed by 2 digits for the month and 2 digits for the day. The year, month, and day may be separated by a space, dash, or period. Time formats that are recognized begin with a 2-digit hour followed by 2 digits for the minute followed by 2 digits for the second. The second may be followed by a decimal point (or comma) and a fraction of a second. The hour, minute, and second may be separated by a colon or space. If both date and time formats are included, the date format must come first, and may be separated from the time format by a space or a capital letter 'T'.

Dates must begin with the year (which must always be 4 digits long, using leading zeros for years before 1000). The month and day are optional, but including the day without the month is not permitted. If the day is omitted, the first day of the month is assumed. If both the month and day are omitted, January 1 is assumed. If a date is given without a time, SenseTalk assigns a time of noon on the given date.

If a time is given it must begin with the hour. The minute and second are optional, but the second cannot be included if no minute is given. If omitted, the minute and second are assumed to be zero.

A date/time value may be followed by the capital letter 'Z' to designate a

zero offset from UTC (Coordinated Universal Time) or a '+' or '-' followed by either 2 or 4 digits to indicate the number of hours, or hours and minutes offset from UTC.

Currently, SenseTalk will not treat an all-numeric value (all digits and possibly a decimal point) as a date/time, so for any value to be recognized as an ISO 8601 date/time it must include at least one separator between elements, or the letter 'T' after the date or before the time (or between the date and time if both are given), or the letter 'Z' or a time zone offset at the end.

- Added ability to specify 'as boolean' (or call **asBoolean()** function). Also 'as integer', 'as base64' (or call **asBase64()** function). These are primarily useful to control how values included as parameters in an XML-RPC message (or when calling the XMLRPCFormat() function) will be treated.
- Added **base64Encode()** and **base64Decode()** functions to encode any data into a base-64 text representation and to decode a base-64 text string into a data value.
- Allow 'as a string' as synonym for 'as text'.

Bug Fixes / Tweaks:

- Fixed a problem with an error on shutdown that prevented the process from terminating cleanly (although the script run had finished).
- Color Picker values now match what is inserted into the script. (Ignoring screen calibration). This includes OCR Tolerance, TIG Foreground and TIG Background colors.
- Fixed a problem where OCR ContrastTolerance range didn't include values at the very end of the range, eg color: (0,0,0) and tolerance of 30 wouldn't match (30,30,30) only up to (29,29,29).
- When Option-Clicking on an empty line eggPlant no longer tries to match an image and load the image interface.
- Added additional bullet-proofing so OCR calls will raise an error rather than terminating when a resource isn't available.
- Fixed a problem with the DO button becoming disabled when terminating a script while there was a pending DO action.

- Fixed a memory leak when standardizing paths. Also one with incoming socket connections resulting from the "accept connections" command.
- Fixed a problem in the **makeURL()** function when encoding list values within a query such that the **extractURL()** function could not recover them. Each value is encoded separately now. Before:

```
put makeURL(query:(a:1, b:(10,22,33)))
--> http://?a=1&b=%2810%2C22%2C33%29
put makeURL(query:(a:1, b:
(10,22,33))).extractURL
--> (query:(a:"1", b:"(10,22,33)"),
scheme:"http")
```

Now:

```
put makeURL(query:(a:1, b:(10,22,33)))
--> http://?a=1&b=10&b=22&b=33
put makeURL(query:(a:1, b:
(10,22,33))).extractURL
--> (query:(a:"1", b:(10,22,33)),
scheme:"http")
```

Release v11.11

(28-November-2011)

Bug Fixes / Tweaks:

- Prevent license expiration notices from interrupting CLI scripts.
- Provide proper notice when email lookup fails on license panel.
- Allow the Licenser Host field to take a host name as well as an IP address.
- Fixed a problem with leaving a blank value in the Licenser Host field.
- Removed the obsolete Fixed Port checkbox from Floating Licensing Options.
- Fixed a problem getting a "Too Many Users" error on Windows.

Release v11.10

(16-November-2011)

Highlights:

- Added an interface for managing licenses directly within the application. (Internet Connection Required)
 - Added an option for eggPlant to check for updates. (Internet Connection Required)
 - Added support for RealVNC Mobile connections on Windows (Android, Blackberry, Symbian S60 and Windows Mobile)
 - **This will be the final release with support for Ubuntu 8 and earlier.**
 - **This will be the final release with support for Fedora 12 and earlier.**
-

User Interface:

- Added a setting to change (or disable) the Countdown Delay when the Manual Image Doctor engages.
 - Script names in the Results pane of the Suite window are now shown in green or red to reflect whether the most recent run of that script resulted in success or failure.
-

Scripting:

- Improved ReadText() and ReadTable() to return multi-line strings separated by **return**.
 - Added the ability to find OCR text that includes **return** in the string.
 - Added CaptureDate and CaptureHost to the information recorded when an image is captured, and available through the ImageInfo() function.
 - Enhanced the ability to identify strings when finding OCR text.
 - You can now specify a HotSpot override when using OCR text finding commands.
 - Added the ability to specify a -username when making a connection via the CLI.
-

Bug Fixes / Tweaks:

- Fixed problem with launching Eggplant for Linux on multiprocessor machines.
 - Added the ability to drag and drop text within a Script Editor window (when this feature is enabled in the Preferences).
 - Fixed the shell() command on Windows to handle path's with spaces in the path name.
 - Restored crash log submission on Mac OS X 10.7 (Lion).
 - Dramatically reduced memory use when doing long waitFor's or imageFounds within a script.
 - Fixed a number of memory leaks doing intensive repeat loops.
 - Fixed an occasional hang on Mac when a connected rendezvous server goes away.
 - Fixed problem that OCR property settings passed as an options parameter to ReadText or ReadTable would persist for future calls (stored in the readTextSettings global property).
 - Fixed a problem on Mac OS X 10.7 (Lion) where you couldn't paste into the find/replace panel.
 - Fixed an exception when changing the **predefinedVariables** on Mac OS X 10.7 (Lion).
 - Prevent creating a new Suite in place of an existing suite.
 - Prevent **Save As...** for Suites which would copy all data.
 - Improved the behavior in the Save Image panel when attempting to create a new folder with the same name as an existing folder.
 - Fixed some problems in the Save Image panel when a subfolder has the same name as one of its parent or ancestor folders.
 - Fixed problem with the creation date of the Eggplant.app itself.
-

Release v11.03

(09-October-2011)

Highlights

- Added embedded required libraries for OCR engine for compatibility on systems where libraries had been updated.

Bug Fixes / Tweaks:

- Properly restore connection size after a connection is closed abnormally.
- Fixed a problem where Eggplant would allow scripts to fail when a connection was reconnecting.

- Fixed a problem where Eggplant would report no active connection when a connection was reconnecting.
 - Fixed a lockup problem when Eggplant is reconnecting while screen is being searched.
 - Fixed the inability to using arrow keys in certain text fields within the Suite Window.
 - Fixed a crashing problem when an image in a sub-folder was selected when another image updated.
 - Fixed a rare problem where images would not show up in the Images Tab after being added.
 - Removed temporary TEXT properties, with a *, when using FoundImageInfo().
 - Fixed a crash following an exit repeat during iteration of a list, if the list is then iterated fully.
 - Removed a spurious warning about the "Spell Checker" when running an ask or answer command.
 - Support Submit panel no longer submits when return is pressed.
 - Fixed a problem when launching multiple instances of Eggplant simultaneously, reporting GSAApplicationLock already locked.
-

Release v11.02

(12-September-2011)

Bug Fixes / Tweaks:

- [Added support for RedHat Enterprise 5 Distributions of Linux.](#)
 - Pastes from the main menu will now remove formatting characters and colors.
 - Made the ReadText and ReadTable commands conform to the RemoteWorkInterval delay before reading values from the screen.
 - Fixed a problem with certain newline characters showing up in the logs.
-

Release v11.01

(25-August-2011)

Bug Fixes / Tweaks:

- [This release is NOT compatible with Red Hat Enterprise 5.x. A subsequent releases will restore RHE 5.x support.](#)
- Added the ability to specify image names for the SearchRectangle of an OCR mouse command.

- Fixed a problem with ReadText() when the searchRectangle was set.
 - Fixed a problem with the Script Editor context (eg: right-click) menu.
 - Fixed a potential hang when connections needed to be re-established.
 - Added greater details when unable to verify a valid Floating License.
 - Fixed a problem where the release notes could reopen every time the application launched.
 - Added support to run on SELinux platforms without needing "Permissive" set.
 - Added support for UltraVNC 1.09 connections.
 - Fixed a crash after setting the searchRectangle.
-

Release v11.0

(15-August-2011)

Highlights

- Added a multi-purpose **ReadText()** function to read text from any screen using OCR.
 - Added the ability to search all scripts in a suite.
 - Added find and replace capabilities to script editor.
 - Added the ability to easily repeat part of a script for a given length of time, or until a specified time.
 - Added support for **RealVNC Mobile 1.5** Connections.
 - There is now a generic Text Search engine for finding text strings on the screen.
If you have NOT modified your text preferences this is now the default Text Platform.
 - Fixed a number of bugs and compatibility issues – (See Below).
 - **This release is currently NOT compatible with Red Hat Enterprise 5.x. A subsequent releases will restore RHE 5.x support.**
-

User Interface:

- Added a Search Field for Scripts, allowing you to search script names or the full content of every script in the suite.
- Added a "Last Run" column on the Scripts pane of the Suite window, that shows the time when each script was last run. The time for a successful run will be shown in green; for a failed run in red. Clicking the last run time will

show the results for that script run in the Results pane.

- Added a connection type pop-up to the VNC Connection panel that allows specifying explicitly which protocol to use for a VNC Connection. For users with mobile enabled this includes the RealVNC Mobile Connection type.
- Enabled editing of script names directly in the Scripts tab. Simply select a script then click its name a second time to begin editing. Double-clicking quickly will still open the script.
- Added support for find and replace in the find panel when editing scripts.
- Eggplant will no longer allow you to create two images by the same name in the same folder, regardless of what file type extensions they use.

Scripting:

- Changed the initial setting of **the defaultLineDelimiter** to be the set of all common line endings (CRLF, LF, CR, LineSeparator, ParagraphSeparator) rather than just the Return (LF) character.
- Added **ReadText()** function which takes one or two parameters. The first (required) parameter is either a rectangle specifying the area of the remote screen from which text is to be read, or a point or image indicating the location of some text to read. The function returns whatever text was recognized from that part of the screen. In the case of a rectangle, it will read all text contained within that rectangle. In the case of a point, it will attempt to locate text near that point, and will read and return as much text as it can find adjacent to that text on the same line.

The second (optional) parameter is a property list of TextReading options.

Example:

```
    put readText((40,0,400,30))
--> Finder  File  Edit  View  Go

    put readText((200,10))
--> Edit
```

- Added a **HighlightRectangle** command which draws a rectangle highlight in

the Remote Screen window. It requires a first parameter which is the rectangle to highlight. An optional second parameter is a property list of options, which may include color and duration properties. The **color** option specifies the color to use for drawing the rectangle, which defaults to red. The **duration** property specifies the length of time the highlight rectangle will be shown. If not specified it will be shown for the default ImageHighlightDuration time. Specifying zero for the duration will cause the highlight to remain until the next time a highlight rectangle is shown, either by another call to this command, or by any screen search command that draws highlights.

- A new Text Finding Engine (**OCR Search**) now exists in the Eggplant Preferences -> Text section.

If you have NOT modified your text preferences this is NOW the default Text Finding Engine.

If you have modified your text preferences you can create a new platform that uses OCR. You can always change your Default Text Platform using the Text Preferences. When using the **OCR Search** text engine you can specify text in any image finding command. You don't need to specify font characteristics but you can select language options for various styles.

Example:

```
click(Text: "File") // If OCR is your default platform
moveTo(Text: "1:34 AM", TextPlatform:"OCR") // If OCR is not the
default platform
```

- The **RunWithNewResults** command will no longer change the status of any connections. Connections that are open when the command is issued will stay open when the called script begins, and connections that are open when the called script finishes will stay open in the calling script.
- Added an **every** option to the **offset** and **range** functions, to obtain the offset location of every occurrence of one text string within another or the full range (starting and ending positions) of every instance of some text within a larger text string. In addition to the natural language syntax, these can also be called using new **everyOffset()** and **everyRange()** functions.

```
put every range of "at" in "concatenation" into
atRanges
put atRanges -- (5 to 6,9 to 10)
```

- Added initial support for **units** associated with numeric values. Any variable holding a numeric value can have a unit type associated with it. Two types of units are officially supported at present: a time interval (or duration) type with a unit of **seconds** and a byte size (or data size) type with a unit of **bytes**. Values created with the special syntaxes for these types will automatically be assigned with those unit types:

```
    put 2 hours into waitTime -- a duration with units
of 'seconds'
```

```
    put 16 megabytes into capacity -- a byteSize with
units of 'bytes'
```

- Added the **unitsEnabled** global property to specify whether units should be shown when displaying a value (the initial setting is false):

```
    put 1 minute 30 seconds into delay
    put delay -- 90
    set the unitsEnabled to true
    put delay -- 90 seconds
```

- Added the ability to **repeat for duration** to create a repeat loop that terminates after a specified length of time has passed. This type of repeat loop will execute the body of the repeat block, then check whether the specified length of time has elapsed since the repeat began. If so, the loop is terminated, otherwise the body of the repeat loop is executed repeatedly with the time being checked at the top of the loop each time until the specified time duration has passed.

```
    repeat for 3 seconds
        add 1 to counter
    end repeat
    put "Counted to " & counter & " in 3 seconds"
```

- Extended the **is a** operator to allow testing whether a value is a particular type of unit. For example you can test whether a value is a duration using the "is a duration" (or "is a timeInterval") operator, or whether it is a byte size using the "is a byteSize" (or "is a dataSize") operator:

```
    put 3 minutes into x
    put x is a duration -- true
    put 5 megabytes is a dataSize -- true
```

- Added the **units** property of a variable to access or change the units associated with that variable. For timeInterval values, the units will be "seconds"; for a byte size value it will be "bytes".

```
    put 1 minute 30 seconds into delay
    put delay's units -- seconds
    set delay's units to "bytes"
```

Other units may be used but are not specifically supported, so should be used experimentally with the understanding that the behavior may change as new unit types are added in the future:

```
set the unitsEnabled to true
put 12 into snack -- (must be a number before
assigning units)
set snack's units to "peanuts"
put snack -- 12 peanuts
subtract 5 from snack -- (eat a few)
put snack -- 7 peanuts
```

- Added ability to **repeat until** *time* to allow a script to repeat a series of steps until a particular date/time has arrived.

```
repeat until "5:00 PM"
    doSomeWork
    wait a minute
end repeat
```

- Added ability to **wait until** *time* to allow a script to pause execution until a particular date/time arrives. If the time specified is already in the past, execution proceeds immediately.

```
wait until "5:00 PM"
wait until now + 1 second -- evaluated repeatedly;
will wait forever
wait 1 second -- do this instead
```

- Added a **sorted** operator that provides all the functionality of the **sort** command. As an operator, it produces a sorted value without changing the source value. The sorted operator takes the form "*value sorted options*" where options include the same options available for the **sort** command, plus "**as chunkTypes**" to sort the value as something other than a list of items.

```
put (99,3,2,5,32,36,45737,26) sorted --
(2,3,5,26,32,36,99,45737)
put randomList sorted in ascending order into
sortedList
put "Eggplant is great" sorted as words --
```

```
"Eggplant great is"
    put "Eggplant is great" sorted as words by the
length of each
    -- "is great Eggplant"
```

- Added the **asTextLimit** global property to set a limit on the size of internally-generated text representations of values. If a text representation is requested that exceeds this limit an exception is now thrown. This protects the system from possible crashes caused by extremely large values. The default limit is ten million characters.
- Added a **globallyUniqueString()** function to return a unique string each time it is called. The value is unique across systems, application instances, and individual calls to the function.
- Added some additional pieces of information returned by the **SystemInfo()** function, and by the **OSInfo()** function on Linux and Windows systems.
- Added the **predefinedVariables** global property. This is a property list containing the definitions of all of the predefined variables. Any changes made to this property will affect any subsequent use of an unassigned variable whose name is one of its keys. By modifying this property, predefined variables can be created or removed, or their value can be changed.

```
put st -- "st"
set the predefinedVariables.st to "SenseTalk"
put st -- "SenseTalk"
```

- Added some additional predefined variables, in the Common Symbols, Numeric Symbols, and Miscellaneous Symbols categories. To see all of the predefined variables and their values, simply execute this command:

```
put the predefinedVariables
```

- Added **increasing** and **decreasing** as synonyms for **ascending** and **descending** for specifying the order of a sort operation.

Bug Fixes / Tweaks:

- Fixed a discrepancy between big and little endian mobile devices which resulted in a pixel shift issue.

- Fixed a problem setting the maximum window size to not account for the toolbar size.
- Added encryption support for RealVNC Mobile servers.
- The **imageFound()** function will now throw an exception if called with only one parameter that is a number. If the first parameter is a number it is treated as the time to search, so at least one image must also be supplied. To search for an image whose name is a number without also giving a time parameter, include the image's file type extension
- Fixed an occasional problem cutting text from a script.
- Fixed a problem where changes made to the Text properties were not made available to the AHDB windows. Now changes made in preferences will cause the AHDB context to reload the default values.
- Fixed a problem that could occasionally lead the **remoteClipboard()** function called with a time parameter to throw an exception incorrectly indicating that no new value was available.
- Fixed a crashing/memory problem when setting the **SearchRectangle** repeatedly in a script.
- Fixed a bug in the cursor location toolbar which would report an incorrect value when the screen was scrolled.
- Fixed a rare hang when running a script from the command line.
- Fixed an issue setting the fonts used for scripts and logs.
- Fixed a problem with the "**New Script...**" menu item being disabled when a window other than a script or suite window was selected.
- Improved the behavior when typing the name of an image to save that is the same as (or begins with) the name of an image folder or collection.
- Fixed a memory leak of a socket, which could lead to a crash when checking connection availability.
- Fixed a problem with display of values in the Duration column on the Licenses panel.
- Fixed a bug with the Remove button on the Licenses panel deleting the last license rather than the selected license.
- Improvements to the UI behavior of all search fields.
- Fixed a bug that would throw an exception if **Add Image** was used to add an image to a commented line in a script.
- Fixed time values specified without a particular date to be treated as a time on the current day (previously a bug was treating these as being on the first day of the current year).
- Fixed a bug with deleting the first item of a list while iterating over that list (could cause the second item of the list to be skipped).
- Fixed a bug (introduced in 10.22) with displaying lists or property lists containing trees (including results of "all nodes of" expressions). The trees would appear as an empty property list.
- Fixed some problems with the **standardFormat()** function, which in some

cases (such as values containing ">" followed by a return) would produce an incorrect representation that could not be turned back into the original value using the value() function.

- Fixed a problem with checking whether a range **is within** another range.
- Fixed a bug that would display ranges within lists or property lists as lists, rather than as ranges.
- Fixed a bug that would throw an exception if a **next repeat** statement was executed more than 10 times within a try block inside a repeat loop, with the message "SenseTalk Runtime Exception - TRY statement exceeds maximum nesting limit per handler (10)".
- Fixed a bug (introduced in 10.22) with displaying lists or property lists containing references to objects.
- Fixed a script formatting bug (introduced in 10.2) that would cause problems when pasting any odd-length text string at a location exactly 99 characters before the end of a script that is at least 200 characters long.
- Fixed a bug with standardFormat() that would throw an exception in certain cases (such as when a property list contained a standard Cocoa object).
- Improved the error message when the call depth limit is exceeded to provide a helpful hint. This error was also **renamed from "SRUN_RecursionLimitExceeded" to "STMaxCallDepthExceeded"**.
- Fixed a number of script memory leaks, including some when accessing chunks of a value.
- Fixed a crash when accessing lines of a file in certain situations.
- Fixed a crash when setting a variable to one of its items that is an integer.
- Fixed a bug that could lead to a crash if an exception occurred within a sort condition inside a try block.
- Fixed a problem with formatting of lists and property lists that would fail to change the quote format back after rendering a list within a property list or vice versa.
- Added additional details to warning when unable to locate suite for a messenger.

Release v10.23 (10-Dec-2010)

Highlights

- Bug fixes, stability and performance improvements.
-

User Interface

- Fixed a crashing problem when deleting or replacing an image and then accessing images through the Images Tab.
 - Fixed a problem when deleting logs from the Script Results pane.
 - Fixed a problem using the Insert.. pop-up button to insert a RightClick command.
 - Cleanup of About Panel GUI.
-

VNC

- Fixed a crashing problem when using the Zlib based encodings. ZlibHextile, ZlibRaw, and ZRLE should all be more stable.
 - Reduced Remote Screen "Shimmering" when in Full Size mode -- this shimmer still occurs in scaled mode.
-

Scripting

- Added a **sorted** operator for beta testing. This takes the form "*value sorted options*" where options include the same options available for the **sort** command, plus "**as chunkTypes**" if you want to sort the value as something other than a list of items. This feature is considered beta, as it needs more testing and may undergo some finishing tweaks before the next release.

```
put (99,3,2,5,32,36,45737,26) sorted --
(2,3,5,26,32,36,99,45737)
put randomList sorted in ascending order into
sortedList
put "Eggplant is great" sorted as words --
"Eggplant great is"
```

- Added **millionth** as a valid ordinal number.
 - Added **millisec**, **millisecs**, **microsec**, and **microsecs** as abbreviations for millisecond, milliseconds, microsecond, and microseconds.
-

Bug Fixes / Tweaks

- Fixed a problem creating TextPlatforms or TextStyles that only differed by case.
- Cleanup of preference panes GUIs.
- Fixed some issues with storing data into byte chunks that would previously convert values to and from text internally.
- Fixed the standardFormat function to format lists with a space after the opening brace so that nested lists can be evaluated correctly by the value function without being interpreted as block quotes.
- Fixed the "number of lines of" function and "each line of" expressions to allow a list for "delimited by" and use the lineDelimiter if delimiter is omitted or empty.
- Fixed a problem with error reporting when an error occurred in evaluating the setup conditions of a repeat loop, that would report as being an error in the preceding line of the script.
- Fixed a bug with accessing the properties of an object that was passed into a handler as a single-item list containing the object.
- Fixed a bug with accessing the currentIndex property or iterating using nextValue or nextAssignedValue of a list when that list contains a single value which is an object.
- Fixed a bug with assigning a list of chunks to a value that would treat them as references ("put items (3,4) of x into y").
- Fixed a bug when performing vector arithmetic using a list containing references to other values, that would alter the referenced values.
- Fixed a problem where Remote Windows sometimes tracked the mouse when not the active window.
- Fixed a problem with GetOptions reporting about FullScreenFailsafe.
- Fixed a bug that prevented iterating over the same range or list multiple times in a nested manner.
- Fixed a bug when saving a script to a different Suite.
- Fixed a bug where non-active Remote Windows would report that they were running the script.
- Fixed a bug that would prevent similar objects (that were equal in value, but distinct objects) from both being included in the helpers or early helpers of another object.
- Fixed a crash when the SUT screen resolution changed.
- Fixed a problem where enabling/disabling some encodings wouldn't take effect.
- Fixed a bug using the TypeText pull-down menu.

Release v10.22 (9-Sep-2010)

Highlights

- Bug fixes, stability and performance improvements.
-

User Interface

- Fixed a problem entering a VNC password on RedHat based distributions.
-

VNC

- Added control to optionally exclude VNC screen synchronization using the defaults write Eggplant RequireSync NO
-

Scripting

- Improved performance when displaying large lists or property lists or converting them to text.
- Restored the default behavior of **AsTextProperty** to **enabled**.
- Added optional "indent" property to the listFormat and the propertyListFormat. Setting this property to a suitable string such as " " will cause individual list items or property list entries to be displayed on separate lines with values indented by multiples of the indent value according to the nesting of the structure.

```
set list to (apple, banana, cherry)
put list -- (apple,banana,cherry)
set the listFormat's indent to " " -- indent by 2
spaces
put list --
(
  apple,
  banana,
  cherry
)
```

Bug Fixes / Tweaks

- Fixed a hanging problem when viewing **ScriptResults()**.
 - Fixed a bug that prevented iterating over the same range or list multiple times in a nested manner.
 - Fixed a bug that would prevent similar objects (that were equal in value, but distinct objects) from both being included in the helpers or early helpers of another object.
 - Executions of **runscript** will once again block and report output until the scripts have finished executing.
-

Release v10.21 (12-Aug-2010)

Highlights

- Improved VNC performance and compatibility including KVM enhancements.
 - Major stability improvements.
-

User Interface

- You can set a Default search type for new images in the Viewer Preferences.
 - Results for scripts called with RunWithNewResults can now be easily jumped to by using the Linker Arrow in the Result Viewer.
 - The Format menu has been removed, as creating scripts in RTF format is no longer supported. Similarly, the Default Script File Format has been removed from the Script preferences pane.
 - The Restore Defaults action on the Text preferences panel now displays a dialog warning that it will remove any custom text platform information.
 - Added the ability to pick a SUT from the Connection List on the Schedule Tab.
 - A new command on the Help menu provides a link to the new online tutorials.
-

VNC

- Fixed a crashing problem when using the Zlib based encodings. ZlibHextile, ZlibRaw, and ZRLE should all be more stable.
- Added support for maintaining a connection when changing SUT screen resolutions.

Scripting

- The current setting of **the ScriptLogging** property is no longer reset by the **RunWithNewResults** command. The ScriptLogging property is now shared in common between the master script and any scripts called by the RunWithNewResults command, so any change to the property in one script will carry over to the others.
- Improved performance of **scriptResults()** function.
- Enhanced performance of the **format()** function. This function also now supports additional format codes %a, %A, and %@ (%a and %A are not available on Windows).

Bug Fixes / Tweaks

- Fixed sporadic crashing problems associated with entering protected code.
- Fixed a crash when the SUT screen resolution changes.
- Fixed an occasional crash when being prompted for a VNC password.
- Fixed a problem requesting a trial license before any license has been entered.
- Fixed a problem accessing Suites from a scripts which had not previously been opened.
- Fixed a bug with a scripted TIG not working correctly when the TIG script is not located in the current suite.
- Fixed a problem undoing a script change.
- Fixed a problem where enabling/disabling some encodings wouldn't take effect.
- Fixed a bug testing the existence of an object property when the property name is in a variable.
- Fixed a problem with some error messages (such as when attempting to access a URL) not reporting properly.
- Improved some error messages.
- Fixed a bug which would cause a crash when performing various operations on a list or property list that contained a circular reference to itself.
- Fixed a bug with deleting an item from a range.
- Fixed a bug with copying list items between lists or to a different place within a list, that would incorrectly treat the copy like a reference to the original item.

- Fixed a bug with passing a list item as a parameter to another handler, that would in effect pass it by reference.
- Fixed a bug with sorting items in a list nested within another list.
- Fixed a bug that would throw an exception when trying to iterate "by reference" over the items of a list (using "repeat with each item of theList by reference") when the list was received as a parameter or is itself a reference.
- Fixed a bug with **the exception** not being set to empty on a single-line **try** statement that doesn't throw an exception.
- Fixed a problem introduced in the compiler in release 1.36 (Eggplant 10.03) that would treat the plain words 'date' and 'time' as function calls rather than variables.
- Fixed a problem with forcing dates to compare as text in some situations ("the time as text is someTime" would compare as times rather than text).
- Always use normal SenseTalk formatting when storing structured values into a text chunk or a file.
- Fixed a problem with storing into files in some situations. This now correctly stores the respective values into the two files (previously it did nothing):

```
put ("abc","def") into (file "/tmp/a", file "/tmp/
b")
```

- Fixed a bug that would return empty when using a range to access items in a list, such as:

```
put items 5..10 of 20..1
```

Release v10.2 (10-May-2010)

Highlights

- New Text Recording feature for Remote Window allows recording most key commands.
- Dramatically improved script formatting and editing performance.
- Major stability and User Interface improvements.

User Interface

- The New Text Recording feature allows you to just start typing when in Capture Mode in the Remote Screen Window. A panel showing the TypeText command to be inserted will allow you to preview your command before

sending it to the SUT. This feature replaces the insert TypeText action.

- Menu actions now use the traditional Linux Ctrl-Key instead of the Meta-Key.
 - You can set a Default search type for new images in the Viewer Preferences.
 - You can set a Default Image Format for recorded images. Issue this command in a terminal using a valid TYPE: ("png", "tif" or "tiff"):
default write Eggplant ImageFormat TYPE
 - You can now double-click in lists to open Connections, edit Scripts and more.
 - Major performance improvements in script editing.
 - Bug Submission Panel now provides feedback on if the submission was accepted.
 - Improved Control over helper suites, allows setting Home and Default Suite directory relative paths by using tilde(~) and dot(.) respectively. Select a helper and then double-click the path to edit.
-

VNC

- Dramatically reduced the number of screen updates Eggplant requests during typical exchange with the VNC server.
 - Added the ability to forgo full-screen updates before image searches fail. These updates are difficult for slower VNC servers and certain KVM devices to handle. To disable full-screen updates issue this command in a terminal:
default write Eggplant FullScreenFailsafe NO
 - Added a **mouseRepositionSpeed** to allow setting the reposition to happen slowly, rather than jumping. Generally this is not recommended as sliding the mouse can cause side effects on the screen, but for KVM and other devices it is sometimes necessary.
-

Scripting

- Quoted lowercase strings used in the context of the auto-released **ShiftKey** will now be interpreted as upper-case letters by all VNC servers. Example: TypeText shiftKey, "n" will print a capital "N" previously it was dependent on the target VNC Server.
- When running in FullScreen mode **Ask** or **Answer** panels will now be displayed on top of the remote screen.
- **Line** chunks are now defined as chunks of text separated by any of a list of delimiter strings (as specified by the lineDelimiter property), rather than always

being separated by return characters.

- Added **the lineDelimiter** local property and **the defaultLineDelimiter** global property to permit changing the set of strings used as delimiters between "line" chunks. Either property can be set to a list of delimiter strings. The initial setting of the defaultLineDelimiter is simply the return constant (also known as linefeed or LF) for compatibility with earlier versions. Setting the defaultLineDelimiter to empty is a shortcut to set it to the list (CRLF, LF, CR) to match any of the common line endings. The lineDelimiter property is local to each handler execution context (or frame) and is set to the value of the defaultLineDelimiter at the start of each handler.

The order of the delimiters is significant, with preference being given to delimiters earlier in the list. So setting the lineDelimiter to (CRLF, LF, CR) will match CRLF (a carriage return character followed by a linefeed) as a single delimiter rather than as two separate line delimiters.

- Added **the characterFiller**, **the wordFiller**, and **the lineFiller** global properties to give control over how strings are extended when putting values into non-existent (out of range) character, word, or line chunks.

The characterFiller is initially set to "." (a period), so putting something into a character position beyond the current value in a container will fill in the intervening positions with dots (NOTE: this is a change from earlier behavior):

```
put "A" into notes
put "G" into character 7 of notes
put notes -- "A.....G"
```

The characterFiller isn't restricted to a single character. A longer string will be repeated as needed to fill in the space:

```
set the characterFiller to "Ha"
set laugh to empty
put "!" into character 7 of laugh
put laugh -- "HaHaHa!"
```

The wordFiller is initially set to "?" to represent any unspecified words (NOTE: this is a change from earlier behavior):

```
set phrase to "start here"
put "the end" into word 7 of phrase
put phrase -- "start here ? ? ? the end"
```

Set the wordFiller to empty for the same behavior as earlier releases (simply fill in with a single space), or to a different value:

```
set the wordFiller to "umm..."
put "Hello and" into greeting
set word 5 of greeting to "welcome!"
put greeting -- "Hello and umm... umm... welcome!"
```

The lineFiller is initially set to **return** for the same behavior as earlier versions,

but may be set to a different line ending (such as CR or CRLF) or some other text if desired.

- Added **top()**, **bottom()**, **left()**, and **right()** functions to get the location of the indicated edge of a rectangle (a single value).
- Added **center()**, **topCenter()**, **bottomCenter()**, **leftCenter()** and **rightCenter()** functions to get the point at the center of a rectangle, or at the center of the top, bottom, left or right edge, respectively, of a rectangle.
- Added the ability to test whether any value is an iterator using the "is an iterator" operator (lists, ranges, and objects with an objectType of "iterator" are all iterators).

```
put (1,3,5,9) is an iterator -- true
```

- Fixed a problem with multiple each expressions within an expression that could cause a hang. It is now possible to use more complex expressions that combine each expressions, like these:

```
put each item of "A".."C" & each item of 1..4
-- ((A1,A2,A3,A4),(B1,B2,B3,B4),
(C1,C2,C3,C4))
put each item of 1..10 times each item of 1..10
into timesTable
```

- Parentheses can now be used around "for each" expressions and "where" clauses to enhance readability and limit the scope of the expression:

```
put each.first && each.last (for each item in
employeeList)
put each line of sales (where item 3 of each >
5000)
```

- Added **is at least** and **is at most** as synonyms for the **greater than or equal to** and **less than or equal to** operators. Also added **is no less than** and **is no more than** variants for these operators:

```
set x to 5
put x is at least 5 -- true
put x is no more than 7 -- true
put x is at most 5 -- true
put x is no less than 5 -- true
```

- Assigning values to chunks with out-of-range negative indexes now properly inserts chunks at the beginning of the container as necessary:

```
set list to (1,2,3)
put "X" into item -6 of list
put list -- (X,,1,2,3)

put "A" into goingDown
put "G" into character -7 of goingDown
put goingDown -- "G.....A"
```

- Changed name of the initialHandler script in a folder object from "<initialHandler>" to "_initialHandler_" (for greater compatibility across different file systems).

Bug Fixes / Tweaks

- Fixed problems with script execution hanging when running Command Line (runscript).
- Selecting text within a script now works when dragging beyond the window region.
- You can now use the Open Script dialogue to open scripts inside any Suite.
- You can now select any text in the Run Window (Script or Log).
- Fixed a problem calling ConnectionInfo() on a server that didn't exist - this now returns an empty property list.
- Fixed a problem using the Connect() commands which could result in an exception.
- Fixed a problem where the Connect() command was case-sensitive on properties that were passed in.

Release v10.12 (09-November-2009)

Linux

Highlights

- New Text Recording feature for Remote Window allows recording most key commands.
- Images are now stored in PNG format for compatibility to all versions of Mac OS X.
- **Text Platforms are now stored in the file `~/Library/Eggplant/TextPlatforms.plist`. See below for more detail.**

User Interface

- The New Text Recording feature allows you to just start typing when in Capture Mode in the Remote Screen Window. A panel showing the TypeText command to be inserted will allow you to preview your command before sending it to the SUT. This feature replaces the insert TypeText action.
- Added the ability to enable/disable caching at the TextPlatform level.
- Added the ability to reset TextPlatforms to their default values.

- Clearer enabling/disabling of controls when doing Image Doctor searches.
 - Improved Suite Info Panel
-

Scripting

- When running in FullScreen mode Ask or Answer panels will now be displayed on top of the remote screen.
- **Line** chunks are now defined as chunks of text separated by any of a list of delimiter strings (as specified by the `lineDelimiter` property), rather than always being separated by return characters.
- Added **the `lineDelimiter`** local property and **the `defaultLineDelimiter`** global property to permit changing the set of strings used as delimiters between "line" chunks. Either property can be set to a list of delimiter strings. The initial setting of the `defaultLineDelimiter` is simply the return constant (also known as linefeed or LF) for compatibility with earlier versions. Setting the `defaultLineDelimiter` to empty is a shortcut to set it to the list (CRLF, LF, CR) to match any of the common line endings. The `lineDelimiter` property is local to each handler execution context (or frame) and is set to the value of the `defaultLineDelimiter` at the start of each handler.

The order of the delimiters is significant, with preference being given to delimiters earlier in the list. So setting the `lineDelimiter` to (CRLF, LF, CR) will match CRLF (a carriage return character followed by a linefeed) as a single delimiter rather than as two separate line delimiters.

- Added **the `characterFiller`**, **the `wordFiller`**, and **the `lineFiller`** global properties to give control over how strings are extended when putting values into non-existent (out of range) character, word, or line chunks.

The `characterFiller` is initially set to "." (a period), so putting something into a character position beyond the current value in a container will fill in the intervening positions with dots (NOTE: this is a change from earlier behavior):

```
put "A" into notes
put "G" into character 7 of notes
put notes -- "A.....G"
```

The `characterFiller` isn't restricted to a single character. A longer string will be repeated as needed to fill in the space:

```
set the characterFiller to "Ha"
set laugh to empty
put "!" into character 7 of laugh
put laugh -- "HaHaHa!"
```

The `wordFiller` is initially set to "?" to represent any unspecified words (NOTE:

this is a change from earlier behavior):

```
set phrase to "start here"  
put "the end" into word 7 of phrase  
put phrase -- "start here ? ? ? ? the end"
```

Set the wordFiller to empty for the same behavior as earlier releases (simply fill in with a single space), or to a different value:

```
set the wordFiller to "umm..."  
put "Hello and" into greeting  
set word 5 of greeting to "welcome!"  
put greeting -- "Hello and umm... umm... welcome!"
```

The lineFiller is initially set to **return** for the same behavior as earlier versions, but may be set to a different line ending (such as CR or CRLF) or some other text if desired.

- Added **top()**, **bottom()**, **left()**, and **right()** functions to get the location of the indicated edge of a rectangle (a single value).
- Added **center()**, **topCenter()**, **bottomCenter()**, **leftCenter()** and **rightCenter()** functions to get the point at the center of a rectangle, or at the center of the top, bottom, left or right edge, respectively, of a rectangle.
- Added the ability to test whether any value is an iterator using the "is an iterator" operator (lists, ranges, and objects with an objectType of "iterator" are all iterators).

```
put (1,3,5,9) is an iterator -- true
```

- Fixed a problem with multiple each expressions within an expression that could cause a hang. It is now possible to use more complex expressions that combine each expressions, like these:

```
put each item of "A".. "C" & each item of 1..4  
-- ((A1,A2,A3,A4),(B1,B2,B3,B4),  
(C1,C2,C3,C4))  
put each item of 1..10 times each item of 1..10  
into timesTable
```

- Parentheses can now be used around "for each" expressions and "where" clauses to enhance readability and limit the scope of the expression:

```
put each.first && each.last (for each item in  
employeeList)  
put each line of sales (where item 3 of each >  
5000)
```

- Added **is at least** and **is at most** as synonyms for the **greater than or equal to** and **less than or equal to** operators. Also added **is no less than** and **is no more than** variants for these operators:

```
set x to 5  
put x is at least 5 -- true  
put x is no more than 7 -- true  
put x is at most 5 -- true
```

```
put x is no less than 5 -- true
```

- Assigning values to chunks with out-of-range negative indexes now properly inserts chunks at the beginning of the container as necessary:

```
set list to (1,2,3)
put "X" into item -6 of list
put list -- (X,,,1,2,3)
```

```
put "A" into goingDown
put "G" into character -7 of goingDown
put goingDown -- "G.....A"
```

- Changed name of the initialHandler script in a folder object from "<initialHandler>" to "_initialHandler_" (for greater compatibility across different file systems).

Bug Fixes / Tweaks

- Text Platforms are no longer part of the preferences (aka defaults) system and are now stored in the file ~/Library/Eggplant/TextPlatforms.plist. When 10.1 first launches it will relocate your Text platform preferences to there.
- Fixed a bug where the first call of an image in a scripted TIG as a Text Image Generator was ignored.
- Fixed a bug where image search type was reset when doing Image Doctor: Replace Image.
- Fixed a bug where Eggplant could hang from Command Line when Auto Doctor was enabled.
- Fixed a bug when making a collection of an image with a DOT in it's name.
- Fixed a problem passing file names into the PLAY command.
- Fixed a host of problems entering and storing information in the Text Preferences.

Release v10.05 (04-September-2009)

LINUX

Highlights

- Easier startup commands on LINUX
- LINUX now ships with the Eggplant Floating License Server.
- Fixed many a number of bugs and other stability issues.

User Interface

- The Eggplant Floating Licenser application is now included for users with floating licenses.
 - The Eggplant Package Now Installs the following Shell Commands:
 - eggplant** - to launch eggplant and log the output.
 - efls** - to invoke the eggplant floating license server.
 - defaults** - to alter user preferences when required.
 - eggplant-debug** - to launch eggplant using GDB when requested by TestPlant support.
-

Bug Fixes / Tweaks

- Setup Eggplant installer to run on SELinux enabled systems.
 - Fixed warnings raised by the installer about setting a partial context.
 - Fixed a bug setting breakpoints and typing text on the first line of a script.
 - Eggplant output on LINUX is now automatically stored in ~/Library/Logs.
 - Fixed a bug changing the default suite directory on LINUX.
 - Fixed a bug on LINUX setting text generators in the Text Prefs pane.
 - Fixed a hanging bug on LINUX when processing UserDefaults -- commonly hit when reconnecting SUTS on launch.
 - Fixed a hanging bug on LINUX as a result of xwindows exceptions.
 - Fixed a crashing bug on LINUX when setting remote window title -- usually when starting a script.
 - Fixed crashing bugs on LINUX when hitting xcb_lock exceptions.
-

Release v10.04 (20-August-2009)

LINUX

Highlights

- Eggplant preferences (aka defaults) are now stored under the Name **Eggplant** for both Linux and Mac platforms. Eggplant will continue to honor the old setting at **com.redstonesoftware.Eggplant** but the new **Eggplant** domain will take precedence.

- Fixed a number of bugs and other minor issues.
 - Includes Vine Server 3.11 which fixes problems with ZLibHex tile encoding.
-

User Interface

- Added the ability to reset script colorization to the default values.
 - Improved the Licensing Panel to allow specifying the Host Address for a Floating License Server.
-

Command Line Interface

- Added a ***GlobalResultsFolder*** *folderName* flag to request that all script results be written to a particular folder (or a particular folder relative to the suite).
-

Bug Fixes / Tweaks

- Fixed a hanging bug on LINUX when processing Rich Cursor updates from some VNC servers.
- Fixed a hanging bug on LINUX when handling an abnormal termination from some VNC servers.
- Fixed a crashing bug on LINUX when receiving the Alert Bell from VNC servers.
- Bulletproofing on LINUX to prevent exceptions in sub-threads from shutting down the application.
- Bulletproofing on LINUX to prevent a deadlock when accessing preferences from sub-threads.
- Fixed a bug on LINUX where the capture window could get progressively darker.
- Fixed a bug on LINUX where the script log window would not stay scrolled to the bottom when new log information was displayed.
- Fixed a bug on LINUX where scripts called via the command line interface using a relative script filename did not reference results and other scripts properly.
- Fixed a bug on LINUX with failures to lock the X-Windows graphic context (xcb-lock errors).

- Eggplant for LINUX now uses the modern FFI library (instead of FFCall library) for communication.
 - Fixed a bug where the name reported by the ConnectionInfo function would not change to reflect the display name specified in the Connection List.
 - Removed log warnings for missing fonts.
 - Added backtrace report output when Eggplant crashes.
-

Release v10.01 (27-April-2009)

Highlights

- Installer will now setup Eggplant to be launched from the Gnome->Applications menu system.
 - Fixed a number of bugs and other minor issues.
-

Bug Fixes / Tweaks

- Fixed a common problem where Eggplant wouldn't recognize changes in the "active script" resulting in an inability to record images in the current script.
 - Fixed a crash that could occur on making a connection (either from a script or the GUI panel)
 - Fixed a hang that could occur when clicking on the color well.
 - Fixed a number of problems with menu's not enabling/disabling or updating titles.
 - Fixed a problem reopening scripts/suites.
 - Fixed a problem with Eggplant not opening certain documents (Getting Started/Release Notes)
 - Fixed a problem customizing the Viewer Window Toolbar.
-

Release v10.00 (20-April-2009)

This initial release of Eggplant for Linux contains almost all of the functionality available in Eggplant for Mac. Below are the features that are excluded at this time. Please report any problems that you encounter to support@testplant.com.

Known Exclusions from the Mac version of Eggplant:

- **Text Image Generation** is only supported through **external** and **scripted** generators.
- **Color Panel** Does not currently support **Named Colors**.
- **ssh** connections ARE supported, but many LINUX versions restrict the ability to pass in the SSH password. We encourage you to set Public-Key Encryption to ensure that your SSH connections work.

*Copyright (C) 2002-2018 TestPlant, Inc – All Rights Reserved
US Patent #7,870,504. Other patents pending.
The eggPlant Logo is a trademark of TestPlant, Inc.*