General Improvements with GainSeeker version 8.7

Major new features in GainSeeker version 8.7

Python Script Editor redesigned for ease of use

Working with Python is now much more intuitive with the newly redesigned Python Script Editor.

Where previous versions of GainSeeker required that the Python script be comprised of one or more "Actions" - each with its own setup interface - the new Python Script Editor provides a code-centric interface for writing Python scripts.

The new editor is backward-compatible, so you can still run and edit your Action-based Python scripts from previous versions of GainSeeker. And if you prefer to continue creating Action-based scripts instead of the new "code-only" scripts, you can configure this in a new System-wide setting.
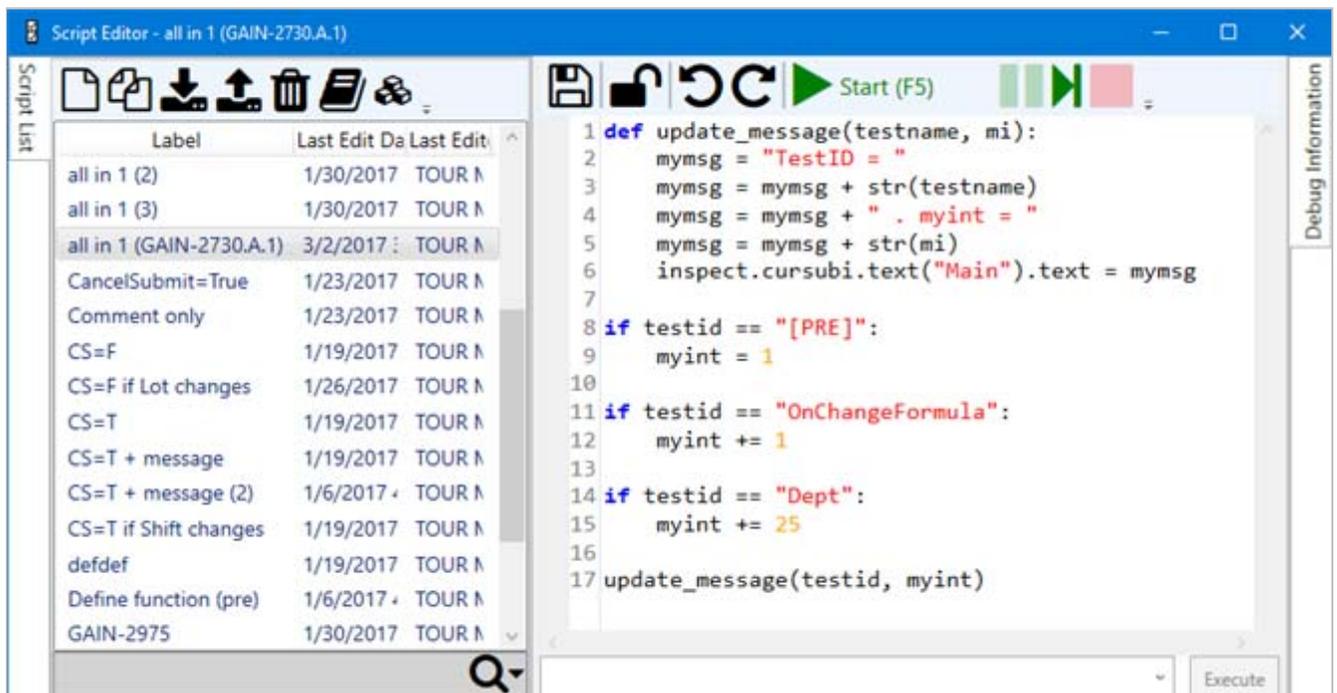
Client installation redesigned

The GainSeeker client installation has been redesigned to simplify the installation process and better support silent installations and installation logs.

In the updated client installer, it is only necessary to specify database connection information the first time the GainSeeker client is installed on your network. Subsequent client installations will automatically detect this information from the shared data path on the network.

Also, a new utility has been created for importing GainSeeker licenses.

Display SPC real-time failure colors on data tables and multiple data tables

A new setting uses real-time failure colors and priorities to indicate real-time failure data on SPC data tables and multiple data tables.

Dashboard can now automatically launch action when next check is due

For the Date Time dashboard control, a new **OverdueAction** property lets you configure an action to be automatically triggered each time this Date Time control changes to the Overdue color.

Edited data and SPC standards now display "Audit Trail" tab

When you edit an SPC data record, DMS data record, or SPC standard, any previous changes to that record are displayed on a new **Audit Trail** tab. It is formatted like the full Audit Trail utility and displays the old and new value for each change.

New rights to "View Audit Trail Tab - Data" and "View Audit Trail Tab - Data" let you determine who can see this **Audit Trail** tab.

Camera or webcam now supported for PC Collect data entry

PC Collect data entry now supports the use of an integrated or external camera or webcam for:

- scanning barcodes to enter Traceability values

- taking pictures for the sub-inspection

New module, GSConsole, can run Python scripts in Scheduled Tasks

In previous versions of GainSeeker, a Scheduled Task to launch the PC Collect module and run a standalone Python script sometimes failed.

For this reason, a new GainSeeker module - GSConsole - is now available for running a Python script in a Scheduled Task. It can use the same command line parameters as PC Collect, and it runs the script silently.

Checkbox test can now be optional

The Checkbox test now includes an **EntryRequired** property. When set to **False**, the inspector is not required to check the box for this test before submitting the sub-inspection.

A corresponding Python command `inspect.cursubi.checkbox(idx).required` has also been added.

Multiple Modules

- Some GainSeeker modules have been modified to load a very long list of standards significantly faster than in previous versions - sometimes up to 20 times faster - particularly in newer modules such as GainSeeker Charts, GainSeeker Inspections, PC Collect, Dynamic Reports, and GainSeeker Utility.

- When entering Limits for SPC standards, a graphical display now plots those limits in relation to each other. This helps to ensure that they are entered correctly.

GainSeeker Inspections

- A new **Save Traceability Values Per Inspection** setting determines whether the traceability values entered in the PC Collect module on a workstation will be remembered only within the current Inspection or across Inspections.

- The **Manage Python** button and menu item has been moved from the PC Collect module to the GainSeeker Inspections module. All editing and administration of standalone python scripts should now occur in GainSeeker Inspections.

  The **PC Collect** button and menu item has been removed from the GainSeeker Inspections module. The PC Collect module can still be launched from the LaunchPad module or from the Windows Start menu.

- In previous versions of GainSeeker, clicking the **Run** button on the Manage Inspections window would launch an instance of the PC Collect module to run the selected inspection. If you used this feature to run the inspection several times - for example, after each round of editing the inspection - this could result in multiple instances of the PC Collect module running on your workstation at the same time.

  Now this **Run** button launches the Inspection Viewer within the GainSeeker Inspections module. The Inspection will function the same as it would in the PC Collect module, but this prevents the unnecessary launching of multiple instances of the PC Collect module on your workstation.

- In previous versions of GainSeeker, the Inspection Editor settings **Display *n* tests on the editor screen** and **Display Python TestID at the beginning of the test label** were reset to default values each time GainSeeker was upgraded to a new version. These values are now set in a file that is used by GainSeeker version 8.7 and later, so your preferences will no longer be reset when you upgrade.

- Documentation has been updated to note that after upgrading from a previous version of GainSeeker, editing an inspection and closing it without making any changes may still display a prompt that asks if you wish to save changes. This is expected behavior and occurs because of new inspection properties in the newer GainSeeker version.

PC Collect

- The Pass/Fail/NA test has been updated so that when the inspector clicks the **N/A** button, a data record is stored with Sample Size = 0, NCU = 0, and Defects = 0 (if the Inspection property StoreAll is set to **True**, the default setting).

- When sending a real-time failure email, the **Compose Email** window will remember whether or not charts were attached to the previous email. This makes it easier for inspectors to send email the same way every time - with attached charts or without attached charts.

- The **Manage Python** button and menu item has been moved from the PC Collect module to the GainSeeker Inspections module. All editing and administration of standalone python scripts should now occur in GainSeeker Inspections.

- The **Manage Inspections** and **Manage Planned Inspections** buttons and menu items have been removed from the PC Collect module. All editing and administration of inspections and planned inspections should now occur in the GainSeeker Inspections module.

Python

- You can use the new `dashboard` commands to refresh or close the current dashboard or to open a dashboard or desktop.

  The `dashboard.refresh()` command can be especially useful when you want the script to first launch a data entry application and then refresh the dashboard as soon as the data entry application is closed.

- The new `disp.alert` command displays a popup message on a colored background in a flashing window. It provides the same basic functionality as the existing `disp.message` command but is more eye-catching to the end user.

- Three new `disp.bringtofront` commands let you bring a running program to the front. You can specify the GainSeeker module running the current script, another process name, or another process id.

- New Python commands `file.xls.lastrow`, `file.xls.lastcol`, `file.xls.rowcount`, and `file.xls.columncount` have been added for counting rows and columns in a file. The commands `file.xls.rowcnt` and `file.xls.columncnt` have been deprecated and are replaced by the new commands `file.xls.lastrow` and `file.xls.lastcol`.

- The new `inspect.shutdown()` command closes the PC Collect module and immediately stops execution of the script.

- The `inspect.cursubi.dmspart(idx).partnumber` command has been updated so that setting it to `None` will now clear the DMS Part Number test. This lets you force the operator to select a new DMS Part Number each time they enter a sub-inspection.

- The new `misc.executescript` command executes another Python script. This can take the place of copying and pasting the same code into multiple scripts, and changes to that code only need to happen in one script.

- The `sql.tables` command has been updated to return the names of more GainSeeker tables.

- In previous versions, the Python variable `cancelsubmit` was only evaluated when the inspection contained a Formula test with the **FormulaMode** property set to **Post** - even if this variable was set in a previous formula. GainSeeker has now been updated so that the presence of a "Post" script is not required for evaluation of this variable. Also, this variable is now reset to `False` each time a sub-inspection is opened.

- When an error occurs in a Python script during an inspection, the error message will now include the name of the script that was running when the error occurred.

GainSeeker Charts

- On a dashboard button, when you set up the **ClickAction** property, the options to load a dashboard or a desktop now have check boxes for both **Edit/Open** and **Keep Current Charts Open**.

- When opening a dashboard with changes, you can now change the number of standards or processes in each retrieval.

- A new setting allows you to hide the Cumulative line from the Pareto chart.

- A new command line parameter - `ExportType=EmailHTML` - allows you to automatically send an email that incorporates the charts into the body of the email but does not duplicate those charts as email attachments.

- When a dashboard includes a Dashboard Grid control, Big Data Analytics DMS control, or Big Data Analytics SPC control - all of which display information on a grid - sending the dashboard to HTML or Email now converts that grid to HTML and places this converted grid below the picture of the dashboard.

GainSeeker Utility

- When exporting data, a new **Set Export File** button on the Export Data Utility window makes it easy to export consecutive data sets to different files without having to close the Export Data Utility. In previous versions, you could only choose the next export filename when you opened the Export Data Utility.

Online User Guide

- A troubleshooting section has been added to the documentation for Python OPC commands.

- Documentation of the **SampleSize** property for the Checkbox test, Defect List test, and Pass/Fail/NA test has been expanded.

- Documentation on copying and pasting tests in a sub-inspection has been expanded.

- Documentation of the statistical calculations for the DMS statistics "Total *Sample size*" and "Total sample cost" was updated to make the terminology more consistent with the rest of the statistical documentation.

- Documentation of the statistical calculations for the SPC statistic "Data in Control" was updated for clarity.

- Documentation has been updated for the **Decimal Places**, **Exponential Notation**, and **Limits** settings for an SPC standard.