Version 2018 Features

Updated: June 19, 2018

Upgrade Notes	
Area	**Feature**
All	**Adobe Flash replaced with HTML5** Adobe Flash is no longer required by any InetSoft product. All product functions that formerly relied on Adobe Flash have been upgraded to use HTML5. See particular product changes below.
Dashboards	**Dashboards updated to HTML5** Dashboards are now rendered using HTML5 rather than Adobe Flash. All dashboards now support annotation and end-user chart editing. The visual background grid of Visual Composer has been removed, and new features (snap-to-grid, rulers, numerical positioning) have been added to enable precise dashboard layout. Dashboard functionality is otherwise the same as the previous Flash-based version.
Data Worksheets	**Data Worksheets updated to HTML5** Visual Composer is now based on HTML5 rather than Adobe Flash. The Data Worksheet interface has been substantially redesigned to improve workflow. The new interface features two panels: Use the **top panel** to create and mashup data blocks via joins, unions, etc. Use the **bottom panel** to view the data within individual data blocks, and set data block properties such as filtering, grouping, and sorting. Both the web-based User Portal and desktop-based Style Studio use Visual Composer for Data Worksheet design.
Administration	**HTML5 flags removed** The server property " `force.html5`" and the related URL code ("`op=vs_html`") are no longer needed, and have been removed. Dashboards are rendered using HTML5 by default.
Administration	**SSO implementation changed** The recommended approach to implementing single sign-on (SSO) has been changed to utilize a security filter chain. See the "Advanced Server Integration" in the documentation for more details.
Administration	**AJP cannot be configured between proxy and Tomcat server** In version 2018, AJP can no longer be configured between your proxy and Tomcat server. If possible, upgrade your reverse proxy to Apache web server 2.4, and load the `mod_proxy_wstunnel` to enable websockets. (See https://httpd.apache.org/docs/2.4/mod/mod_proxy_wstunnel.html for more details.) If you must use Apache web server 2.2 or below for the reverse proxy, you must add a websocket-capable proxy to marshall the traffic. The websocket-capable proxy can be hardware-based (F5, etc.) or software-based (haproxy, etc.), and should directly route InetSoft requests to the InetSoft version 2018 server. Other requests should be routed to the existing Apache reverse proxy.

Administration	**Integration syntax modified** The syntax for integrating reports and dashboards via direct URL or IFrame has changed. See "Advanced Server Integration" in the documentation set for details on the new syntax.
Administration	**Load-balancing servlet has been removed** The load-balancing (proxy) servlet for clustering has been removed. Load balancing should generally be implemented by using a hardware asset such as F5. Contact InetSoft Technical Support for further guidance.

Administration	**Logging levels modified**
	Logging now uses SLF4J as the logging facade and Logback as the underlying implementation. The available logging levels for property `log.detail.level` in *sree.properties* are now 'debug', 'info', 'warn', and 'error'. Legacy settings will be automatically transformed. For example, `log.detail.level=WARNING` will be transformed to `log.detail.level=warn`. The following transformations are also applied automatically:
	`log.level.inetsoft.report=FINEST` to `log.level.inetsoft.report=debug`
	`log.level.inetsoft.graph=FINER` to `log.level.inetsoft.graph=debug`
	`log.level.inetsoft.uql=FINE` to `log.level.inetsoft.uql=debug`
	`log.level.inetsoft.sree=INFO` to `log.level.inetsoft.sree=info`
	`log.level.inetsoft.util=WARNING` to `log.level.inetsoft.util=warn`
	`log.level.inetsoft.analytic=SEVERE` to `log.level.inetsoft.analytic=error`
	
	The naming scheme for "sree.log" and "schedule.log" files has also been modified to include the IP address.

Administration	**CVS Archive is deprecated** The CVS archive feature for reports is deprecated and has been removed from the Enterprise Manager 'Archive' page.
Data Modeling	**OLAP Overlay has been deprecated** The OLAP overlay feature of data models has been deprecated, and the **Hierarchy** tab has been removed. Legacy users can set property `cube.hierarchy.enabled=true` in the *sree.properties* file to enable the **Hierarchy** tab.
InetSoft API	**API code modifications required** If you have created custom code that uses the *Tool.invokeThread()* method, you need to remove the *Tool.invokeThread()* wrapper and execute the wrapped code directly. If you have created custom code that uses the *KeyRing.setKey()* or *KeyRing.setDefaultKey()* methods, you should replace those calls with *inetsoft.util.ConfigurationContext.setHome()* . See the examples under "API Upgrade"at the end of this document.

Installation	**Installation locations have changed**
	The installation locations for the application and for the data directory have been changed in version 2018. The default application installation location is the following:
	Windows : *C:\Program Files\InetSoft2018*
	Mac : */Applications/InetSoft2018*
	Linux : *usr/local/InetSoft2018 or opt/InetSoft2018*
	The default data directory installation location is the following:
	Windows : *C:\ProgramData\InetSoft Technology\InetSoft\2018*
	Mac : */Library/Application Support/InetSoft Technology/InetSoft/2018/*
	Linux : */usr/share/InetSoft Technology/InetSoft/2018/*
	The default location of SREE Home is the *server-config* directory within the data directory.
Scripting	**Script editor has been updated**
	The Script Editor for Viewsheets, Data Worksheets, and Ad Hoc reports has been updated. The syntax auto-complete function can now be invoked by pressing **Ctrl-Space** on the keyboard. To view help for a function, place the cursor in the function name and press **Ctrl-O**. To open the full function documentation, press the "docs" link to the right the help text.
Scripting	**Component size is based on pixels**
	Because the Viewsheet grid has been removed in version 2018, if you previously use the component 'size' property (in script) to set the size of a Viewsheet component in terms of grid rows and columns, you will need to change these values to set the size based on pixels.
Scripting	**Checkbox component always returns array**
	In previous versions, `parameter.CheckBox1` returned a `null` value for no-selection and string value for single-selection. In version 2018, the return value is an array in all cases: The no-selection case returns an empty array, and the single-selection case returns a single-item array.

Minor Features	
Area	**Feature**
Ad Hoc Reporting	**Ad Hoc reporting interface redesigned**
	The interface of the Ad Hoc report design tool has been slightly revised. The highlight feature (conditional formatting) available in versions 12.1 and prior has been restored in this version. Functionality otherwise remains the same as the previous version.
Administration	**Portal theme can be customized**
	The User Portal color theme can be modified to suit your own needs. To customize the Portal, you will need to supply a color palette to InetSoft Technical Support, who will then provide you with the corresponding Portal CSS needed to create the theme. Contact Technical Support also for information about using custom font to define Portal button images.

Administration	**Database authentication provider added**
	In addition to the built-in InetSoft authentication provider and optional LDAP authentication provider, you can now also specify a database authentication provider. See the Administration documentation for details.
Administration	**CSS classes provided to style Chart brushing**
	The color of brushed Chart elements and un-brushed Chart elements can now be set using the following properties: `brush-highlight-color`, `brush-dim-color`.
Administration	**Local queries, local Worksheets, and embedded data sources listed as import dependencies**
	When a report is imported into the server environment using the Enterprise Manager 'Import Asset' page, the report's local queries, local Worksheets, and embedded data sources are now listed among the report dependencies, and can be included or excluded from the import.
Charts	**Chart trend lines can be enabled for individual measures**
	You can now select the individual measures on a chart for which trend lines should be displayed. This setting is available in the 'Chart Properties' (report) or 'Plot Properties' dialog box, and also via a new script function, `setTrendLineExcludedMeasures()`.
Dashboard Design	**Charts provide a 'Title' property**
	The 'Chart Properties' dialog box now provides a 'Title' property that allows you to place a title above the Chart. The title text can be sized and formatted as desired.
Dashboard Design	**Table column ordering enhanced**
	A dialog box is now provided for changing the order of columns in a table. Columns can also be reordered using drag-and-drop operations, as in previous versions.
Integration	**Viewsheet in IFrame supports hyperlinks to parent window**
	Hyperlinks within a Viewsheet can now pass data to the parent window by using cross-window messaging.
Reports	**CSV export delimiter options added**
	Options to specify the delimiter for CSV exports have been added to the 'Export Report' dialog box and to the Scheduler 'Deliver to Emails' feature.
User Portal	**Data Dictionary report and Viewsheet have been added**
	A new 'Data Dictionary' report and Viewsheet have been added to the default repository in the User Portal. The Data Dictionary allows you to search for any data resources, such as data models, queries, etc.

API Upgrade	
Example: Modifying Tool.invokeThread() Code	

invokeThread()	`// Replace the following code`
	`Tool.invokeThread(myThreadGroup, new Runnable() {`
	`  @Override`
	`  public void run() {`
	`    // wrapped code`
	`  }`
	`});`
	`// with "wrapped code"`
	
	`// Replace the following code`
	`MyObject result = Tool.invokeThread(myThreadGroup, new Callable<MyObject>() {`
	`  @Override`
	`  public MyObject call() throws Exception {`
	`    // wrapped code`
	`  }`
	`});`
	`// With the following:`
	`MyObject result = // result of "wrapped code"`

Example: Modifying KeyRing.setKey() or KeyRing.setDefaultKey() Code	

setKey()
setDefaultKey()
	`// Replace the following code:`
	`ThreadGroup myThreadGroup = new ThreadGroup("myThreadGroup");`
	`KeyRing.setKey(myThreadGroup, "/path/to/sree/home");`
	
	`// With this code:`
	`ConfigurationContext.setHome("/path/to/sree/home");`
	
	`// Replace the following code:`
	`KeyRing.setDefaultKey("/path/to/sree/home");`
	
	`// With this code:`
	`ConfigurationContext.setHome("/path/to/sree/home");`