DecSoft App Builder

History

2026.29 (10/25/2026)

* Fixed an introduced bug at the DecSoft App Builder CLI compiler which cause that the resources of the Apache Cordova or Capacitor apps are not properly created and therefore the final compilation was wrong.

* Updated the VueJS JavaScript library Pinia to the latest 4.1.0 version (debug and production) for the app's template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

* Updated the VueJS JavaScript library Core to the latest 3.5.44 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2026.28 (10/12/2026)

* Updated the VueJS JavaScript library Router to the latest 5.4.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Updated the NUglify tool to their latest version 1.23.4. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

2026.27 (09/21/2026)

* Updated the default Deployment target for iOS at the app Apache Cordova options to 15.0, which will be the minimum value for the next spring of 2027 according to Apple requirements.

* Updated the NUglify tool to their latest version 1.23.1. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

* Updated the VueJS JavaScript library Core to the latest 3.5.43 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2026.26 (09/10/2026)

* Updated the VueJS JavaScript library Router to the latest 5.3.1 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Updated the NUglify tool to their latest version 1.23.0. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

2026.25 (08/27/2026)

* Fix an introduced error which causes that even when we select the "Package bundle" at the Cordova Android Signing options, the "android_sign.json" will refer to create an "APK" instead of an "AAB" (bundle) file.

* Updated the VueJS JavaScript library Core to the latest 3.5.42 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Updated the VueJS JavaScript library Router to the latest 5.3.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2026.24 (08/22/2026)

* Enhanced the CSS of the DecSoft BlockUI plugin, that is, the container of the block UI stuff, in order to be compatible (better said, to look better) with the "edge to edge" Android devices.

* Enhanced the "app.setViewReadyForDeviceKeyboard()" and the "app.setDialogReadyForDeviceKeyboard()" methods for Capacitor apps, which now work exactly in the same way as for Cordova apps: especially better talking about app dialogs.

* Updated the NUglify tool to their latest version 1.22.4. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

2026.23 (08/13/2026)

* Now our Apache Cordova compiled apps will run in "edge to edge" mode (for devices who support it, there are no changes for devices who do not support it). The right enhancements to the app's CSS and some JS functions have been made in order to get this working as expected.

* Updated the VueJS JavaScript library Core to the latest 3.5.41 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Updated the VueJS JavaScript library Pinia to the latest 4.0.3 version (debug and production) for the app's template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2026.22 (08/01/2026)

* Add the Pilled property for the PushButton, Dropdown, Text input, Number input, Password input, Email input, Color input, Phone input, Date input, Time input, Search input, File input and Select input controls. The new Pilled is a

boolean property set to false by default, and, we can set to true to get a rounded pill border for the referred controls. The source code editor metadata and the product help has been updated accordingly and a new Pilled sample app has been added.

* Updated the VueJS JavaScript library Core to the latest 3.5.40 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Updated the NUglify tool to their latest version 1.22.3. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

* Updated the NUglify tool to their latest version 1.22.2. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

* Updated the NUglify tool to their latest version 1.22.1. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

2026.21 (07/15/2026)

* Updated the Font Awesome JavaScript library to the latest 7.3.1 version, which include a lot of changes, fixes and enhancements (6 new icons). The library has been updated in the apps template, the icons picker, the product about, welcome, AI tools chat, help, app plugins helps and "readme" file.

* Updated the VueJS JavaScript library Router to the latest 5.2.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Updated the VueJS JavaScript library Pinia to the latest 4.0.2 version (debug and production) for the app's template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

* Updated the VueJS JavaScript library Pinia to the latest 4.0.1 version (debug and production) for the app's template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2026.20 (07/14/2026)

* Updated the VueJS JavaScript library Pinia to the latest 4.0.0 version (debug and production) for the app's template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

* Enhanced the way used to search for installed web browsers for the Program options -> Debugger -> Browsers. Now we use the right Windows registry key in order to find the available web browsers installed on the system, to be ready to run the apps directly on it. Rather than searching for particular web browsers, examining the correct Windows registry entry allows us to discover a greater number of web browsers.

2026.19 (07/13/2026)

* Updated the NUglify tool to their latest version 1.22.0. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

2026.18 (07/11/2026)

* Updated the NUglify tool to their latest version 1.21.18. This tool is used by DecSoft App Builder to minimize the apps' JavaScript and CSS code, and the latest version of NUglify includes several changes, fixes and enhancements.

2026.17 (07/08/2026)

* This release of the product removes the full installer of Microsoft Runtime Edge from the product installer. Nowadays, including the installer of the Microsoft Runtime Edge is probably not required, since Windows 11 includes it out of the box, and Windows 10 includes it too with the right update. We keep, however, the installer in its "download flavor", just in case we need to use it. By doing this change, we reduce the DecSoft App Builder installer size from 270 MB to 64 MB.

2026.16 (06/26/2026)

* Update the Font Awesome JavaScript library to the latest 7.3.0 version, which include a lot of changes, fixes and enhancements (16 new icons and 8 new animations). The library has been updated in the apps template, the icons picker, the product about, welcome, AI tools chat, help, app plugins helps and "readme" file.

* Update the VueJS JavaScript library Core to the latest 3.5.39 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2026.15 (06/23/2026)

* Fix a bug which causes that we can add an external JavaScript file (from the app Files Manager) with an attribute "none" which is reflected "as is" in the final app index.html, which is wrong, since this "none" type will cause that the external JavaScript file is not retrieved as expected by the browser and then cannot be used by the app. Thanks Neil for reporting this error!

2026.14 (06/12/2026)

* This release of the product has been adapted in order to properly work with the new DecSoft Utils website. This means that the product is capable of reaching the appropriate URLs in order to search for updates, for example.

* Update the VueJS JavaScript library Core to the latest 3.5.38 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.5.37 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.5.36 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Add a third argument for options at the Apache Cordova Media Capture plugin for all the three methods: captureAudio, captureImage and captureVideo.

2026.13 (05/28/2026)

* Update the VueJS JavaScript library Core to the latest 3.5.35 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 5.1.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2026.12 (05/21/2026)

* Make some changes in order to get the app.setViewReadyForDeviceKeyboard() and app.setDialogReadyForDeviceKeyboard() methods working properly on Android if we build our apps using capacitor.

* Update the VueJS JavaScript library Core to the latest 3.5.34 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 5.0.7 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2026.11 (05/05/2026)

* This release of DecSoft App Builder is the first one which added another framework to create hybrid apps in addition to Apache Cordova. Yes! Now it's also possible to use Capacitor in order to create hybrid apps for Android and Apple iOS with DecSoft App Builder. A lot of work has been done, and, a lot of work remains, but, from this release you can choose between Apache Cordova and Capacitor to create hybrid apps with DecSoft App Builder.

* Add a "capacitor-app" attribute with a value of "true" on the HTML element of our apps, if our app is running in a Capacitor platform (Android, iOS). This

attribute can be useful to prepare some CSS rules specifically thinked to be used when the app is running in a Capacitor platform and not in a Cordova platform or a browser.

* Separate the Apache Cordova and Capacitor related files in the compiled app directory. When compile an app, the "www" files (for the browser) are placed like before, on the "compiled/www/" folder. If the app is intended to be compiled with Apache Cordova, then the Apache Cordova stuff is placed on "compiled/cordova/" folder, and, if the app is intended to be compiled with Capacitor, then the Capacitor stuff is placed on "compiled/capacitor/" folder. This means that both Apache Cordova and Capacitor can exists without conflicts in the same "compiled" directory.

* Do not require anymore to install Apache Cordova globally. Instead, Batch and Shell files now install Apache Cordova directly on the compiled app directory. Additionally, instead to check for the "cordova" command, now the BATCH files check if Node JS is installed, since this is what we need to continue installing Apache Cordova. Important: if you are using the Batch files options to add some Apache Cordova plugin, you need to change the plugin lines from "CALL cordova plugin add..." to "CALL NPX cordova plugin add...".

* Add the new "app.store.isBrowserApp", "app.store.isCordovaApp" and "app.store.isCapacitorApp" reactive variables. Note that these variables are defined as reactive variables so we can use also in HTML templates in addition to JavaScript code. You can take a look at the also new sample app "AppPlatforms" to see these variables in action on an HTML template.

* Set the Apache Cordova iOS "Deployment target" option to 13.0 by default: this is required to compile the app with XCode 26. You can of course continue using another deployment target if needed in your apps: if you see problems while compile (Package: cordova-ios-plugins unknow), try to change this option to 13.0

* Add a "cordova-app" attribute with a value of "true" on the HTML element of our apps, if our app is running in a Cordova platform (Android, iOS, Windows). This attribute can be useful to prepare some CSS rules specifically thinked to be used when the app is running in a Cordova platform and not in a Capacitor platform or a browser.

* Change the default "Position" property value of the NavBar control. In previous releases, that property value is set to "fixed-top", but, this property value can cause an unexpected position of the NavBar in modern Android versions. So, by default, now the propery values is set to an empty string, then the NavControl is positioned where we positioned it at designtime. It's our duty now to use another position (like "fixed-top") taking in consideration the possible non expected position in modern Android versions (working like a charm in browsers anyway).

* Add a "data-app-style" attribute to the root HTML element of the apps. This attribute can have a value "fixed" or "scaled", and, changes if we change the app style at runtime. This attribute can be useful to establish CSS rules for a specific app style.

* Add the missing app.style property reference for the source code editor. This app property exists and it's documented, but was not present for the

autocompletion and declarations of the source code editor.

* This release fixes an introduced problem which causes that we cannot use the F8 key (run the latest compilation of the app), the CTRL + F9 keys (compile the app), and the F9 key (compile and run the app), while we are writing code in the source code editor. This fix applies to the controls events editor, the controls properties editor, the app functions editor, and the app search results editor.

* Improve the product help / documentation by making these changes: The top "DecSoft App Builder" link goes now to the index of contents, instead of the DecSoft App Builder website. The search engine remembers now the search query, and also marks the matched words of the search query on the document.

* Simplify the app debugger by removing the list of "know resolutions". Now the app debugger is launched automatically following the app designed size.

* Add the new Tabs2 sample app. This sample is similar to the Tabs sample app, in the way that we use the same HTML markup and CSS classes from Bootstrap CSS to show a "tabs" component; however, this new sample app uses an app global reactive variable and the power of the HTML templates to create the tabs and show them dynamically.

* Update the VueJS JavaScript library Core to the latest 3.5.33 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 5.0.6 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Router to the latest 5.0.5 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2026.10 (04/08/2026)

* Fix a potential "access violation" error when clicking on an app resource or app function reference (to directly go to the resource or function) from the source code editor.

* Update the VueJS JavaScript library Core to the latest 3.5.32 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2026.9 (04/02/2026)

* When we introduce Monaco Editor as our source code editor, we must disallow moving or dragging out the app functions, app search, control events and control properties tabs, since the behaviour next was not the expected: the editor becomes hidden and unusable. This release recovers the ability to move the referred tabs without losing the source code editor.

2026.8 (04/01/2026)

* Replace the component that we use to deal with the Microsoft WebView2 control. With the new component, we are now using the latest API provided by Microsoft and will be able to offer various enhancements in the future related to it.

* Update the Monaco Editor to the latest version. Thanks to the replacement of the component to deal with the Microsoft WebView2 control, we can now keep the Monaco Editor well updated in DecSoft App Builder.

* Use the very latest release of the WebView2 installer (complete) which we use for the DecSoft App Builder installer. This release also fixed various minor errors and added other minor changes and enhancements.

* Update the VueJS JavaScript library Core to the latest 3.5.31 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2026.7 (03/19/2026)

* Set the default Android SDK target to Android 16 (API level 36). This fits with the latest release of Apache Cordova Android. Also add the ability to choose the Android SDK target to Android 17, 18, 19 and 20, for API levels 37, 38, 39, and 40, respectively.

* Update the VueJS JavaScript library Router to the latest 5.0.4 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Core to the latest 3.5.30 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.5.29 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2026.6 (02/23/2026)

* This release enhances how the app dialogs are internally managed. Not only fixing a bug in some specific scenario, which causes that a possible dialog is shown below an already showing dialog, so it's not really shown, but also performs other enhancements. Basically now we can show any number of dialogs, put it always above the previous ones, and never get any orphaned "backdrop" after hiding the dialogs, one by one or all of them at once.

* Update the VueJS JavaScript library Router to the latest 5.0.3 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2026.5 (02/11/2026)

* Update the VueJS JavaScript library Core to the latest 3.5.28 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the Font Awesome JavaScript library to the latest 7.2.0 version, which include a lot of changes, fixes and enhancements (new icons too). The library has been updated in the apps template, the icons picker, the product about, welcome, AI tools chat, help, app plugins helps and "readme" file.

2026.4 (02/02/2026)

* Update the VueJS JavaScript library Core to the latest 3.5.27 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 5.0.2 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2026.3 (01/18/2026)

* After 10 years, the jQuery JavaScript library team are happy to launch the jQuery v4.0.0 version, and, we are proud that this release update to this jQuery version for the Apps template, ReadMe file, Product help, Plugins helps, About page, Welcome page, Icons dialog and AI tools chat. Note that this jQuery version have a lot of fixes and enhancements, but also some breaking changes. The Apps template is ready for it, and, probably your own apps code too, however, maybe you need to check it. We recommend you to read this jQuery blog post: https://blog.jquery.com/2026/01/17/jquery-4-0-0/

* Make some changes in the LeafletMaps app sample to enhance the way in which we can deal with the maps when the where are using the "dark" app theme color in the app. Basically, in addition to the map tiles, we also take care now around other elements of the map interface.

2026.2 (01/03/2026)

* Update the VueJS JavaScript library Core to the latest 3.5.26 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 4.6.4 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2026.1 (01/01/2026)

* This is a major release of the product. If you purchase your license(s) before

01/01/2025, then you need to purchase an upgrade license (with a 50% off discount) in order to register this release. In another case, your license(s) will be updated and you can grab your new serial numbers from your DecSoft customer area. Please, contact us if you have any doubts!

* Update the VueJS JavaScript library Core to the latest 3.5.25 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2025.37 (11/09/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.24 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.5.23 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Pinia to the latest 3.0.4 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2025.36 (10/21/2025)

* Add "viewport-fit=cover" to the "viewport" metatag in the "index.html" of the apps. This fix a problem related with certain non expected margin between the status bar and the app views at iOS.

* Update the VueJS JavaScript library Router to the latest 4.6.2 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2025.35 (10/02/2025)

* Update the Font Awesome JavaScript library to the latest 7.1.0 version, which include a lot of changes, fixes and enhancements (new icons too). The library has been updated in the apps template, the icons picker, the product about, welcome, AI tools chat, help, app plugins helps and "readme" file.

* The Min, Max and Step properties of the Number input control accept now also float numbers in addition to integers at both design time and runtime. Fix also a bug related with the possible usage of float numbers in the value property of the Number and Range inputs controls.

2025.34 (09/27/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.22 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Implement (internally) the "hide" event of the app dialogs and the app modal alerts to avoid an accessibility warning in the browser console: "blocked aria-hidden on an element because its descendant retained focus." This has some other benefits in addition to preventing the warning in the browser console.

2025.33 (09/15/2025)

* Fix a bug in the app Files manager which causes that the first time we add a new script and use the "type" attribute for that script, this "type" is not saved and therefore is lost, and we need to edit the script in order to set the "type" attribute again. Now the behaviour is the expected: the "type" is saved when we add the new script.

* Update the Bootwatch app themes (cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, lux, brite, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) which now are based in the latest version 5.3.8 of the Bootstrap CSS, which is the version we already use in the default app theme. Update the two versions of the themes: LTR and RTL.

2025.32 (09/07/2025)

* Get the "show()" and "hide()" methods of the app dialogs to work as expected. Probably because there are the methods "app.showDialog()" and "app.hideDialog()" and we profusely used them, we forgot to make certain changes in the "show()" and "hide()" methods of the app dialogs, which caused it not to work as expected. This release makes the appropriate changes in order to get such methods working as expected too.

* Improve the product help search form by assert that the content to be searched (file names, titles, and contents) are in lower case, as well as the search query: this offers better matches than before. Other improvements have been made, for example, showing now the number of matches in every search result, like the number of matches in file names, titles, and contents.

* Change (internally) the value property type of the Range and Input Number controls to be Number instead of String. Nothing serious four our apps, but, with this we avoid some "check type fail" notices in the browser developer console, since for these controls the value property must be certainly of a Number type and not a String type.

2025.31 (09/04/2025)

* In addition to the link to the right product help topic, now, when we hover the mouse above an app user function in the IDE editor, we can see a link that, if pressed, will show that app user function in the app functions manager. So now we can go to the app users functions manager from the IDE editor, directly to the app user function under the mouse, and, this also works from the IDE editor completions list hints.

* In addition to the link to the right product help topic, now, when we are over an existing app resource from the IDE editor completions list, we will see a link that, if pressed, will show that app resource in the app resources manager. So now we can go to the app resources manager from the IDE editor, directly to the app resource that we are seeing in the completions list hints.

2025.30 (09/03/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.21 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the Font Awesome JavaScript library to the latest 7.0.1 version, which include a lot of changes, fixes and enhancements (new icons too). The library has been updated in the apps template, the icons picker, the product about, welcome, AI tools chat, help, app plugins helps and "readme" file.

* Add a Help button into the IDE code editor toolbar. When this button is pressed we will see the right help topic according to what we are editing: the help topic for the control event that we are editing, the help topic for the control property that we are editing, the app user functions help topic, the app search help topic, etc.

2025.29 (09/02/2025)

* This release is more or less a maintenance release of the product. We do a lot of internal enhancements in the Delphi library that we wrote for the integration of the Monaco Editor as our IDE editor, as well as fixing some bugs that we found there. No new features have been added, but we want that everything is updated in the latest published release, so here it is.

2025.28 (09/01/2025)

* Assert that, when the app debugger stops, the browser does not continue running the app in the background. Before this release, the browser continues running the app, and if there is some code which, for example, put something in the debugger console, that task continues running, which is for sure not the expected since the debugger has been stopped.

* Change the internal algorithm to determine the size of the debugger browser. We notice the previous algorithm does not work as expected if DecSoft App Builder is running on a Windows "scaled screen" greater than 100%. The changes that we made in this release are trying to work in a similar way no matter the scale that is used for the screen.

* Add the "Autoplay" property to the Carousel control. Before this property is available the Carousel always start cycling automatically, and, we can use the "pause" method at runtime to stop it, however, the "Autoplay" property, only available at designtime, we can determine if the Carousel cycling must start automatically or not. The property is set to True by default to keep compatibility with existing apps. The provided Carousel sample app has been also

modified to remove the "Dark mode", since this has been deprecated from time ago. Thanks Enel for the suggestion!

2025.27 (08/31/2025)

* The IDE editor for the CSS code (for example, the app style custom CSS) will help you now with a completion list which include not only the general CSS related stuff, but also the specific app related CSS stuff. This basically mean that you can easily find every app view, app dialog, app frame and every controls inside, in order to apply your desired custom CSS style for these elements.

* Add an "app-frame" class attribute for the app frames, just in the same way that we already have the "app-view" and the "app-dialog" classes for the app views and the app dialogs. These classes allow customizing the CSS style of all the app views and dialogs, and from now on also for all the app frames if we want. Also we rewrite enterely the "runtime selectors" help topic, in order to be more precise and correct some mistake around these.

2025.26 (08/30/2025)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.8 version for the program's welcome page, program's help, program's about, program's icon picker, AI tools chat, DecSoft app plugins helps, the apps template and the program's "readme" file. This version of Bootstrap CSS includes a lot of fixes and enhancements.

2025.25 (08/29/2025)

* We are very, very proud to announce that this new release of DecSoft App Builder starts the integration of the marvelous Monaco Editor, which will offer you the very best experience writing your app JavaScript, HTML and CSS code from now on. We published a new post in our blog so please visit our website or use the below link to know more, and, of course, update your DecSoft App Builder copy, so you can start enjoy Monaco Editor in DecSoft App Builder!

English: https://www.decsoftutils.com/blog/hello-monaco-editor/

Spanish: https://www.decsoftutils.com/blog/hello-monaco-editor/?lang=es

* This release also have other good enhancement related with the installation and the usage of DecSoft App Builder by various users in the same machine. Basically this is something that we do not take in consideration before, and, certainly can be useful. So we made the appropriate changes in the product and the product installer in order to allow various users in the same machine to work with DecSoft App Builder, using their own configuration and other user specific files like app themes, plugins and others. Thanks again Bojan for the report and help to solve this!

2025.24 (08/28/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.20 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.5.19 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Fix a bug which cause that the Quick Code List of the source editors do not reflect the appropriate changes if an app view, dialog or frame is renamed. Now the new view, dialog or frame name is well reflected in the Quick Code List of the source editors.

* Assert in the program "check requisites procedure" the existence of the "I18n" file that it's required. If we don't check this and the file did not exists the program will end with an exception message, not very funny. Thanks Bojan for the report!

2025.23 (08/19/2025)

* Take care about the possible line breaks in the prompt edit control of the AI Tool - Chat Conversation. Basically before this release the possible line breaks are not properly handled and therefore the user prompt did not appear in the chat as we expected.

2025.22 (08/14/2025)

* Yet more enhancements in the AI Tool - Chat Conversation. We continue enhancing this AI tool, this time particularly decorating some HTML elements in the AI responses. For example, we are adding rounded borders to some HTML elements, and in particular, we decorated the possible HTML tables with some useful CSS classes and made them responsive tables.

* Enhancements in the AI Tool - Code generation. We try to improve this AI tool in various ways. We don't append any more the responses, keeping the response editor with just the latest asked piece of code. We also enhanced the "final" prompt, making it more simple. Additionally, we are trying now to find the right syntax highlighter for the editor based on the user prompt and also the AI response, supporting syntax highlighters for JavaScript, PHP, CSS, HTML, XML and JSON.

2025.21 (08/13/2025)

* Yet more enhancements in the AI Tool - Chat Conversation. Apart from some stylish changes, now we are capable of showing break lines in the AI responses. This is quite important, since certainly AI models tend to place line breaks, so missing it gives us a very long paragraph of text, more difficult to read. But line breaks means more than just "line breaks", that is, this help when show certain content as HTML as we do: in an HTML context line breaks are more than simply line breaks in plain text. Additionally, we also enhanced the markdown to

HTML conversion for the AI responses, giving us more HTML elements (like tables) properly shown, or, another example: possible URLs (without markdown links, just URLs) are now also converted to HTML clickable links.

2025.20 (08/12/2025)

* Enhance the AI Tool - Chat Conversation in order to use an HTML canvas instead of the previous plain text editor. The usage of an HTML canvas has several benefits compared to the plain text editor. For example, some Ollama models tend to use several markdown syntax when responding, and now we are allowed to convert it to HTML, so we increase the legibility of the responses. For example, we are now capable of clicking on possible links in the responses, opening them in your default system browser. We can also now copy to the clipboard individual prompted questions, individual responses, and also the entire chat in text and HTML flavors. We can also repeat the questions by just clicking a button. We can now restart the chat too, which means delete the chat history and start it from scratch. We also avoid some glitches which can be see when using the plain text editor. In few words, the AI Tool Chat Conversation is now even more usable and user friendly.

2025.19 (08/09/2025)

* Change the default models for the OpenAI tools from gpt-4o to gpt-5-nano. This new model is faster and cheaper than the previous one. Remember anyway that you can specify the models to be used for the integrated AI tools, so, if you prefer a specific model, just establish it at the program options.

2025.18 (08/02/2025)

* Make some improvements in the code generation and chat conversation AI tools. Basically we improve how the AI response is placed in the respective editors, doing it in a more convenient way, in order to avoid some editor flickers and that not always we can see the very latest AI response on it (without manual scrolling).

2025.17 (08/01/2025)

* This release say goodbye to the venerable Open AI assistant and hello to the AI tools! This is a try to integrate the power of the AI into DecSoft App Builder, and, even more, now you can use Ollama AI in addition to Open AI: Ollama AI can be installed and run directly on your system! Get more information at the DecSoft blog here: https://www.decsoftutils.com/blog/hello-ai-tools

* There is an error which occur if we put any editor who support it into full screen and try to close the full screen mode using the keys combination CTRL + F4. Now basically we can use this keys combination, in addition to the already supported F11 and Escape to close the full screen mode of the supported editors.

* This is a major release of this product and require a new license (or upgrade license) if you purchase it before 08-01-2025. You can purchase a license from

our store or an upgrade license from your DecSoft customer area, with a 50% off discount. Also from your DecSoft customer area you can get the outdated releases of the product.

2025.16 (07/28/2025)

* This is a maintenance release which fixes an introduced very minor bug (but annoying) in the program options form, which basically shows us a wrong hint while moving the mouse around. Additionally, but not the expected, this release also fixes some memory leaks, not too very important, but certainly existing; therefore, fixable.

2025.15 (07/24/2025)

* Fix a bug related with the usage of the "sr-only" Bootstrap CSS class in the Carousel control. Instead of this class, the one to be used is now "visually-hidden". Additionally we made another minor enhancement on the Carousel control, in order to avoid the use of some non translatable English strings.

2025.14 (07/23/2025)

* In some previous release we move from the Open AI model "gpt-4o" to "gpt-4-preview" for the Code generation and Chat conversations features of the integrated Open AI assistant. This was a mistake, since we don't know that the "gpt-4-preview" model is near to being abandoned by Open AI. So we move back to the "gpt-4o" model, but, in order to avoid something like this happening again, we allow now to specify the models to be used by the Open AI assistant. This can also be good if some customers have access to some models that he considers better to be used, so now this is possible too.

* Update the VueJS JavaScript library Core to the latest 3.5.18 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2025.13 (07/22/2025)

* Update the Inno Setup program to the latest version (6.4.3). This is the software that we use for the creation of the installer of DecSoft App Builder, and, the latest Inno Setup version provide a lot of changes, fixes and enhancements.

* Update the Font Awesome JS library to the latest version (7.0.0) in the app template, the about page, the welcome page, the help pages, the app plugins help pages, the app icons picker dialog and the installer "readme" file. This new release of Font Awesome contains various fixes and enhancements, as well to 6 regular, 20 brand and 5 solid new icons, with a total of 2089 icons.

2025.12 (06/26/2025)

* Update the Bootwatch app themes (cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, lux, brite, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) which now are based in the latest version 5.3.7 of the Bootstrap CSS, which is the version we already use in the default app theme. Update the two versions of the themes: LTR and RTL.

2025.11 (06/19/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.17 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.7 version for the program's welcome page, program's help, program's about, program's icon picker, DecSoft app plugins helps, the apps template and the program's "readme" file. This version of Bootstrap CSS includes some fixes and enhancements.

2025.10 (06/14/2025)

* Include a new main toolbar icon to provide a fast access to the Open AI assistant. This release also includes another minor enhancements in the Open AI assistant.

* Improve the OpenAI assistant by using now the "gpt-4.5-preview" model for the Chat conversation feature, instead of the "gpt-4o" model.

* Improve the OpenAI assistant by using now the "gpt-4.5-preview" model for the Code completion feature, instead of the legacy "gpt-3.5-turbo-instruct" model.

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

* Update the VueJS JavaScript library Pinia to the latest 3.0.3 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2025.9 (05/29/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.16 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2025.8 (05/26/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.15 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2025.7 (05/15/2025)

* Update the VueJS JavaScript library Core to the latest 3.5.14 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.6 version for the program's welcome page, program's help, program's about, program's icon picker, DecSoft app plugins helps, the apps template and the program's "readme" file. This version of Bootstrap CSS includes some fixes and enhancements.

* Update the Bootwatch app themes (cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, lux, brite, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) which now are based in the latest version 5.3.6 of the Bootstrap CSS, which is the version we already use in the default app theme. Update the two versions of the themes: LTR and RTL.

2025.6 (04/26/2025)

* Update the VueJS JavaScript library Router to the latest 4.5.1 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Pinia to the latest 3.0.2 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2025.5 (04/06/2025)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.5 version for the program's welcome page, program's help, program's about, program's icon picker, DecSoft app plugins helps, the apps template and the program's "readme" file. This version of Bootstrap CSS includes some fixes and enhancements.

* Update the Bootwatch app themes (cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) which now are based in the latest version 5.3.5 of the Bootstrap CSS, which is the version we already use in the default app theme. Update the two versions of the themes: LTR and RTL. Also add the new Bootwatch app themes Lux and Brite.

2025.4 (04/04/2025)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.4 version for the program's welcome page, program's help, program's about, program's icon picker, DecSoft app plugins helps, the apps template and the program's "readme" file. This version of Bootstrap CSS includes a lot of changes, fixes and enhancements.

2025.3 (03/30/2025)

* Add Android 16 (API 36) to the available Android target SDK option. On the other hand, upgrade the target SDK default value to be Android 15 (API 35) instead of Android 14 (API 34), to follow the requirements of the new major release of Apache Cordova Android.

* Update the VueJS JavaScript library Pinia to the latest 3.0.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2025.2 (02/03/2025)

* Update the VueJS JavaScript library Pinia to the latest 2.3.1 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

* Since Microsoft decide to remove Wordpad from Microsoft Windows 11, we changed our RTF files for the history and the license of the product to be TXT. This implied some changes in the product in addition to the referred files.

2025.1 (01/01/2025)

* This is a major release of this product and require a new license (or upgrade license) if you purchase it before 01-01-2024. You can purchase a license from our store or an upgrade license from your DecSoft customer area, with a 50% off discount. Also from your DecSoft customer area you can get the outdated releases of the product.

2024.62 (12/24/2024)

* Added the new Slide event in the Carousel control, which fired immediately when the slide instance method is invoked. Inside this event we can access to the also new Carousel control properties fromIndex, toIndex and direction. Take a look at the product help and also at the enhanced Carousel app sample.

* Added the new Slid event in the Carousel control, which is fired when the carousel has completed its slide transition. Inside this event we can access to the also new Carousel control properties fromIndex, toIndex and direction. Take a look at the product help and also at the enhanced Carousel app sample.

* Update the VueJS JavaScript library Pinia to the latest 2.3.0 version (debug and production) for the apps template. This version of the VueJS JavaScript

library Pinia includes a lot of changes, fixes and enhancements.

2024.61 (12/03/2024)

* Our apps offers two events: JavascriptError and RenderError. These events are useful to handle and deal with possible errors in our apps' code. However, apparently we take a bad decision in the past, rely only on the referred events, and, therefore, hidding some possible errors in the browser's or the debugger console, except if we certainly use the referred events. We revert this situation in this release: the events are still available, but, if some error occur, we place in the browser's and the debugger console the information about the error, no matter if we use or not the referred app's events. Thanks Kenneth to report this behaviour!

* Update the VueJS JavaScript library Router to the latest 4.5.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2024.60 (11/22/2024)

* Update the Font Awesome JS library to the latest version (6.7.1) in the app template, the about page, the welcome page, the help pages, the app plugins help pages, the app icons picker dialog and the installer "readme" file. This new release of Font Awesome fix an error found in Safari / iOS browsers.

2024.59 (11/19/2024)

* Update the Font Awesome JS library to the latest version (6.7.0) in the app template, the about page, the welcome page, the help pages, the app plugins help pages, the app icons picker dialog and the installer "readme" file. This new release of Font Awesome contains various new icons available and some other minor changes, fixes and enhancements.

2024.58 (11/15/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.13 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Pinia to the latest 2.2.6 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Pinia to the latest 2.2.5 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2024.57 (10/19/2024)

* This is a maintenance release of the product which did not contain any changes except some minor internal fixes that we want to be available and ready from now on.

2024.56 (10/10/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.12 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.55 (10/09/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.11 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Fix a minor mistake with the position of the Open AI program options, which do not follow the convention of the other program options tabs. Now the referred controls follow the convention of the other program options tabs, which is the expected.

2024.54 (10/01/2024)

* Update the VueJS JavaScript library Pinia to the latest 2.2.4 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Pinia to the latest 2.2.3 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2024.53 (09/28/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.10 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.5.9 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.52 (09/22/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.8 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.51 (09/21/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.7 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.50 (09/17/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.6 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.49 (09/14/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.5 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 4.4.5 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2024.48 (09/11/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.4 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 4.4.4 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2024.47 (09/06/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.3 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.46 (09/05/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.2 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.45 (09/04/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.1 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.44 (09/03/2024)

* Update the VueJS JavaScript library Core to the latest 3.5.0 version (debug and production) for the apps' template. This release of VueJS Core is a major version and contains a lot of changes, fixes and enhancements, so our app's template.

2024.43 (08/28/2024)

* Fix a bug which causes that the DecSoft plugin LabelInput, LabelText, LabelSelect, LabelSelectEx and TypeAhead controls, don't keep the expected scaled height. The bug only occurs if we use the same name for the controls in different views. The solution was to use certain CSS classes instead of JavaScript code to keep the expected height of the controls inputs in their containers.

* Update the VueJS JavaScript library Pinia to the latest 2.2.2 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2024.42 (08/15/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.38 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.41 (08/08/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.37 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.40 (08/07/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.36 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 4.4.3 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Pinia to the latest 2.2.1 version (debug and production) for the apps template. This version of the VueJS JavaScript library Pinia includes a lot of changes, fixes and enhancements.

2024.39 (08/02/2024)

* Add the new Input event in the Text, Number, Password, Email, Color, Phone, Date, Time, Search, Number and Textarea controls. Also add this Input event in the LabelInput, LabelText and TypeAhead app plugins controls included out of the box. The product help (local and online) have been updated to describe these changes. Basically, the Input event is fired before the Change event (all of these controls have a Change event), since no need that the user changes the focus of the controls.

2024.38 (08/01/2024)

* Update the Delphi library that we use with the OpenAI assistant to the latest version. The latest version of this Delphi library have various changes, fixes and enhancements, so also our Open AI assistant.

* Update the VueJS JavaScript library Router to the latest 4.4.2 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Router to the latest 4.4.1 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Core to the latest 3.4.35 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.37 (07/26/2024)

* Update the VueJS JavaScript library Pinia to the latest 2.2.0 version (debug and production) for the apps template. Update the VueJS VueDemi JavaScript library (used by Pinia) to the latest 0.14.10 version (debug and production).

2024.36 (07/24/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.34 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.35 (07/21/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.33 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.34 (07/17/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.32 version (debug and production) for the apps' template. This release of VueJS Core has a lot of

changes, fixes and enhancements, so our apps' template.

2024.33 (07/16/2024)

* Update the Font Awesome JS library to the latest version (6.6.0) in the app template, the about page, the welcome page, the help pages, the app plugins help pages, the app icons picker dialog (which now includes 8 new brand icons and 5 new solid icons) and the installer "readme" file.

2024.32 (07/09/2024)

* Replace the Open AI GPT-4 model used for the Open AI assistant Chat by the Open AI GPT-4o (omni) model. This new model is faster and cheaper (about 50% cheaper) than the previously used Open AI GPT-4 model, so we get this benefits to our Open AI assistant Chat. 2024.31 (07/04/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.31 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.30 (07/01/2024)

* This is a major release of this product and require a new license (or upgrade license) if you purchase it before 07-01-2023. You can purchase a license from our store or an upgrade license from your DecSoft customer area, with a 50% off discount. Also from your DecSoft customer area you can get the outdated releases of the product.

2024.29 (06/22/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.30 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 4.4.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2024.28 (06/21/2024)

* Fix an error when initialize a Date input control without a default value. We are using the Moment JS library to try to initialize the input value, but, from sometime ago, we remove the Moment JS library in favour of the Luxon JS library. We fix the bug by using the Luxon JS library to initialize the Date input control if they have not a default value set at designtime: basically in this case we set the input value to be the current date.

* Fix an error when initialize a Time input control without a default value. We are using the Moment JS library to try to initialize the input value, but, from

sometime ago, we remove the Moment JS library in favour of the Luxon JS library. We fix the bug by using the Luxon JS library to initialize the Time input control if they have not a default value set at designtime: basically in this case we set the input value to be the current time.

* Enhance the style of the TypeAhead DecSoft control (from the DecSoft Controls App Plugin, which is included out of the box) in order to put the type ahead list of items above other possible elements: basically we positioned the list and set an appropriate z-index.

* Fix a mistake in the App Plugins development help / documentation related with the files that we include with the app plugins: the "File" tags that we can use must be self closed XML tags, in order to be valid: in other case, and, like the help refer before this fix, the possible unclose "File" tag, will cause that the plugin (XML) loading operation fail.

2024.27 (06/15/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.29 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.26 (06/14/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.28 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.25 (06/12/2024)

* Internally separate the app files definitions to app StyleSheets and app JavaScripts definitions. This allow to deal with these files in a separated way, which is useful for other changes that we introduce in this release (see below) and others that can may introduce in future releases.

* The app file picker dialog has been removed and we incorporate two dialogs instead: the app StyleSheets dialog and the app JavaScript dialogs. These dialogs are used in the app Files Manager to pick StyleSheets and JavaScripts files to be included in our apps. The help / documentation has been enhanced to describe these new dialogs.

* We can now specify the type attribute of the JavaScript files that we want to include in our apps: the default attribute is "none" (means no type attribute), but we can now choose also the "module", "importmap" and "speculationrules" type attributes for the included JavaScript files. We can choose this types from the new app JavaScript picker dialog. The app plugins can also set now the type attribute for the JavaScript files that they required.

* Update the VueJS JavaScript library Router to the latest 4.3.3 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

* Update the Inno Setup program to the latest version. This is the software that we use for the creation of the installer of DecSoft App Builder, and, the latest Inno Setup version provide a lot of changes, fixes and enhancements.

2024.24 (06/07/2024)

* Remove an inline z-index style on the toasts container (for the Toasts) since apparently this causes some problems in a customer's app, and, in fact, after trying it (by removing) there is apparently no problem to remove that z-index inline style. Thanks Mario for the error report!

2024.23 (05/28/2024)

* Make some changes to be ready with the latest release of Cordova Android (13.0.0): add the Android target SDK for Android 16 (API level 35). Set the Android target SDK for Android 15 (API level 34) by default for new apps. Use the new "SplashScreenBackgroundColor" preference key in the Cordova configuration file, which is now used for Cordova Android but also for Cordova iOS. The new Cordova Android release add other fixes and enhancements, for example, now the camera intent can be used in input File controls if we set the Capture property to "user" or "environment".

2024.22 (05/08/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.27 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.21 (04/29/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.26 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.20 (04/25/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.25 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.19 (04/24/2024)

* Remove the nonstandard "orient" attribute from the Range control. Use the "writing-mode" CSS property if the Range control orientation is set to "vertical". Use the "form-control" instead of the "form-range" CSS class from the Bootstrap CSS framework. These changes cause that the Range control in vertical orientation works as expected in all modern browsers and avoid a

console notice about the usage of certain nonstandard CSS property. Before these changes, the Range control in vertical orientation don't work as expected in the Firefox browser, and we get a notice in the console due to the usage of a nonstandard CSS property.

2024.18 (04/23/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.24 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.17 (04/17/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.23 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.16 (04/16/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.22 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.15 (03/28/2024)

* Update the Delphi library that we use with the OpenAI assistant to the latest version. We need to do some internal changes in order to use the latest version of the Delphi OpenAI library, and, move on to use the GPT4 model instead of the "GPT4-preview" one for the OpenAI assistant chat capability.

2024.14 (03/22/2024)

* Fix an introduced bug which causes that if we set (and it's set like that by default) the "latest" version for Cordova Android, Cordova iOS and Cordova Electron, the proper "latest" version is not included in the BAT and SHELL files. This can cause some problem (basically we need to edit manually the BAT and SHELL files) if we need specifically to use the "latest" version (and no other one) for our apps. From this release the "latest" version is properly added to the BAT and SHELL files, or, the version that we specify in the app options.

2024.13 (03/06/2024)

* Update the Font AwesomeJavaScript library to the latest 6.5.1 version. This include the Font Awesome which is included in our app templates and also the used by the DecSoft App Builder icons picker dialog. This new release of the library include several new icons ready to be used in our apps.

* Update the Luxon JavaScript library to the latest 3.4.4 version for the apps template. Luxon is a library for working with dates and times in JavaScript and it's included out of the box to be used in our apps. This new release of the library include various changes, fixes and enhancements.

* Update the js-base64 JavaScript library to the latest 3.7.7 version for the apps template. js-base64 is a library to encode and decode Base64 in JavaScript and it's included out of the box to be used in our apps. This new release of the library include various changes, fixes and enhancements.

2024.12 (03/01/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.21 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the Bootwatch app themes (cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) which now are based in the latest version 5.3.3 of the Bootstrap CSS, which is the version we already use in the default app theme. Update the two versions of the themes: LTR and RTL.

2024.11 (02/26/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.20 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Router to the latest 4.3.0 version (debug and production) for the apps template. This version of the VueJS JavaScript library Router includes a lot of changes, fixes and enhancements.

2024.10 (02/20/2024)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.3 version for the program's welcome page, program's help, program's about, DecSoft app plugins helps, the apps template and the program's "readme" file. This version of Bootstrap CSS includes a lot of changes, fixes and enhancements.

* Update the VueJS JavaScript library Core to the latest 3.4.19 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.9 (02/09/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.18 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.17 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.16 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.8 (02/06/2024)

* Update the NUglify binaries used by the Code Minimizer tool to the latest version. This new version of NUglify provide various changes, fixes and enhancements.

2024.7 (01/21/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.15 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.14 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.6 (01/15/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.13 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.12 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.5 (01/12/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.11 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.10 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.9 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.8 version (debug and production) for the apps' template. This release of VueJS Core has a lot of

changes, fixes and enhancements, so our apps' template.

2024.4 (01/09/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.7 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.6 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.3 (01/04/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.5 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.2 (01/03/2024)

* Update the VueJS JavaScript library Core to the latest 3.4.4 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2024.1 (01/01/2024)

* This is a major release of this product and require a new license (or upgrade license) if you purchase it before 01-01-2023. You can purchase a license from our store or an upgrade license from your DecSoft customer area, with a 50% off discount. Also from your DecSoft customer area you can get the outdated releases of the product.

* Update the VueJS JavaScript library Core to the latest 3.4.3 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.2 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.1 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Update the VueJS JavaScript library Core to the latest 3.4.0 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template. This is a major release of the VueJS JavaScript library Core, which contains a lot of changes, fixes and enhancements.

2023.71 (12/20/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.13 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Fix a rare bug in the Carousel control when use it in a mobile device in which we shown (in other view) the device keyboard. In this specific case, when go back to the view in which the Carousel control is shown, they don't appear as expected. Thanks Harald for the report!

2023.70 (12/10/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.11 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2023.69 (12/05/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.10 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

* Replace the "text-davinci-003" model for code generation in the Open AI assistant with the recommended model "gpt-3.5-turbo-instruct". The "text-davinci-003" model has been marked as deprecated and become not available in January 2024.

2023.68 (11/27/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.9 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2023.67 (11/13/2023)

* Enhance the Open AI assistant which now use the new Chat-GPT-4 model instead of the Chat-GPT-3.5 model, and, the new DALL-E 3 model instead of DALL-E 2 model. The new models offer better results, for example, the Chat-GPT-4 model training data is newest and the DALL-E 3 model (for images generation) produces better images.

2023.66 (11/07/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.8 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2023.65 (10/28/2023)

* This release includes support for the RTL (Right To Left) versions of all the available app themes: the default one, but also the cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr themes. The DecSoft App Builder automatically include the RTL version of the themes that you want to include with the app, and, automatically, the compiler will set the right theme according to the app text direction property established at designtime.

* Now when you call to the app.setAppTextDirection() method, the properly version of the current app theme (LTR or RTL) is automatically changed, according to the text direction that you specified when calling to that app method. In the same way, when you call to the app.setAppTheme() method, the properly version of the theme (LTR or RTL) is set, according to the current app.textDirection property.

* The Themes sample app has been enhanced in order to include an "RTL / LTR" button, which shows how we can changet at runtime the text direction of the app, and, therefore the properly theme version (LTR / RTL) automatically. The app help has been also updated in order to refer these release changes related with the new support for RTL themes.

* Add the new RTL languages sample app. This sample app shows how we can support languages which requires a different text direction (Left To Right / Right To Left). The app is designed by default with the Arabic language, which requires the RTL text direction, and, we use the app languages manager to add the English language, which requires the LTR text direction. You can simply see how easy is to support both LTR and RTL languages.

* Properly set / update the HTML tag "lang" attribute to the app language. By default, the DecSoft App Builder compiler set the referred tag's attribute with the app language, but, now we also set / update it when we call the app.translateView() method, so we keep the referred tag's attribute with the right language established in the app at runtime.

2023.64 (10/25/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.7 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2023.63 (10/22/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.6 version (debug and production) for the apps' template. This release of VueJS Core has a lot of changes, fixes and enhancements, so our apps' template.

2023.62 (10/19/2023)

* Fix a bug in the ImagePush control which causes a warning in the browser console related with certain duplicate attribute which is placed by mistake in the control HTML markup. Thanks Mario for the bug report!

* Update the VueJS JavaScript library Pinia to the latest 2.1.7 version (debug and production) for the apps template. Update the VueJS VueDemi JavaScript library (used by Pinia) to the latest 0.14.6 version (debug and production). Update the VueJS JavaScript library Router to the latest 4.2.5 version (debug and production) for the apps template.

* Assign the F7 shortcut key to the "Stop the app debugger" action. Previous to this release we can't stop the debugger using a shortcut key, but, we can launch the app in the debugger using the F8 (launch the last compilation of the app) and the F9 (save with debug information and then launch the app) shortcut keys, so, now we also have a shortcut key in order to stop the debugger if they are running.

2023.61 (10/07/2023)

* Fix a bug in the app.setDialogReadyForDeviceKeyboard() method, which is not working as expected, probably due to some latest change in the newest Bootstrap CSS framework. This method is important when we include in our app dialogs some inputs controls and deploy our apps in devices like Android and iOS. The method is intended to not stretch the dialog size when the devices' keyboard appears when press in the input controls. This release fix the bug so this method is working like expected. You can take a look at this method working in the Inputs sample app, for example.

2023.60 (10/04/2023)

* Enhance the way that we use to search for available browsers from the program debugger options. The browsers set from the program debugger options appears in the program toolbar and allows launching our apps in any of the available browsers by clicking in the right browser toolbar icon, in addition to the DecSoft App Builder internal debugger browser. We can always add browsers by selecting the browser executable path, but, DecSoft App Builder can try to automatically search for the 32 and 64 bits versions of the Opera, Vivaldi, Brave, Firefox, Chrome and Edge browsers.

* For some reason, when we want to compile the app with the Apache Cordova iOS platform from Windows (to get an XCode project ready for Mac OS), the "platforms/ios" folder is not properly updated (with the latest app changes) if the platform has been added previously. Now we remove the "platforms/ios" folder (if exists) so we force that the "platforms/ios" folder contains the latest changes that we may made in the app.

2023.59 (10/03/2023)

* Before this release, we offer the ability to customize the BAT and SHELL files for Apache Cordova generated by DecSoft App Builder. This is useful, for

example, if we want to add some custom Apache Cordova plugins in a customized way (beyond the plugins IDs). We offer a way to add specific code at the start of a BAT or SHELL file, after the add Cordova platform command is executed, and, at the end of the BAT or SHELL file. Conceptually the above is correct, however, we are not doing it in the best way as possible. The mistake was to add into the iOS BAT file (which allows us to create a XCode project for our app from Windows) the BAT start, after and end app options, but, this is not correct, since maybe we don't want to use the same plugins (or customizations) for Android than for the Apple iOS platform. A similar mistake occurs for the Apache Cordova Electron BAT and SHELL files that we are generating: we are adding into these files the BAT start, after and end app options, as well the SHELL start, after and end app options, but, again, maybe we don't want to use the same plugins for Apache Cordova Electron that the ones used for Android or Apple iOS. In this release we enhance the BAT and SHELL start, after and end app options, by offering a way to set it separately per every supported platform, and additionally, for the case of the Apple iOS BAT file (for Windows), we also offer now a way to customize that specific BAT file. Doing this we can now customize the BAT and SHELL files per every platform, so, we can use, for example, different Apache Cordova plugins per every platform.

* Make changes in the product help (local and online) to remove the BAT and SHELL app options from the screenshots, since these options are now integrated in the app options pages which correspond with the different supported Apache Cordova platforms. We also fix a bug which causes that the app don't link correctly to the right help page when we press the F1 key in various app options pages.

2023.58 (10/02/2023)

* Change the viewport metatag of the app index.html file, in order to be the best as possible for all the supported platforms, specifically for the Apple iOS platform, in which we note that with the viewport that we are using previously to this release, the app views are not rendered as best as it can be rendered with the new used viewport meta-tag.

2023.57 (10/01/2023)

* Update the Bootwatch app themes (cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) which now are based in the latest version 5.3.2 of the Bootstrap CSS, which is the version we already use in the default app theme.

2023.56 (09/28/2023)

* Previous to this release, the version for the Apache Cordova Electron platform was always "latest". This is the recommended value and is the version that we use for the Apache Cordova Android and the Apache Cordova iOS platforms too, however, for the Apache Cordova Android and Apache Cordova iOS platforms we can specify a different version from the app options. Well, in this release we can now also to specify a version for the Apache Cordova Electron platform from the

app options, in the same way as we can do it for the other platforms. By default, the version is "latest", which is the recommended value.

2023.55 (09/27/2023)

* This release drops support for the Apache Cordova Browser and Apache Cordova Windows platforms, since Apache Cordova dropped support for these platforms sometime ago. Remember that, for the Windows platform, we can use the Apache Cordova Electron platform in order to compile our apps as executables for Windows, or, if you prefer, you can use our DecSoft HTML Compiler to get your apps as executables for Windows.

* This release also drops support for various "sub platforms" of Apache Cordova, which since sometime ago have been dropped from Apache Cordova. We are talking about platforms like Amazon (Android), Tizen, Blackberry, etc. These only imply that the creation of splash screens and the config.xml file for Apache Cordova do not contain any more specific information for these already unsupported platforms.

* This release of DecSoft App Builder, talking about the Apache Cordova integration, puts things in a way that we are completely up-to-date with Apache Cordova, supporting all the platforms that Apache Cordova supports today, and not the dropped ones by Apache Cordova from sometime ago. Since we do not create any more stuff for the dropped platforms, the creation of icons, splash screens, config files, and BAT and SHELL files is now faster for the supported platforms.

* Now, the Apache Cordova version used by default for the Android and iOS platforms is the "latest" one. We recommend that, at least for new apps, we always use the "latest" version of Apache Cordova, so this is the option that the apps will have by default now: we can always use a specific Apache Cordova version by changing the version value from the app options and for the platform that we want.

2023.54 (09/14/2023)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.2 version for the program's welcome page, program's help, program's about, DecSoft app plugins helps, the apps template and the program's "readme" file. This version of Bootstrap CSS includes a lot of changes, fixes and enhancements.

2023.53 (09/13/2023)

* Update the Delphi library that we use with the OpenAI assistant to the latest version. In the latest version of the referred library, "[...] the API has been fully updated in accordance with the official [OpenAI] version as of September 13, 2023". The library also has other changes, fixes and enhancements, so our OpenAI assistant.

2023.52 (09/11/2023)

* Fix a bug in the command-line compiler. Even if we use the option "minimize" to minimize the app JavaScript and CSS code, the applangs.js file, the pluginscomps.js file, and the pluginscomps.css file are never minimized as expected. This error does not occur if we use the code minimizer tool from the IDE, when all the expected files are minimized, but from the command line, and even when we use the minimize option, the referred files are not minimized. This release fixes this bug, and therefore the code minimizer tool works as expected from both the IDE and the command-line compiler.

2023.51 (09/10/2023)

* In this release, we replace the Microsoft AjaxMinifier tool with the NUglify tool (https://github.com/trullock/NUglify). The Microsoft AjaxMinifier tool is a good tool, but it does not support JavaScript features like async functions. On the contrary, NUglify is fully ES2020 + ES2021 compliant, so it allows minimizing a larger scope of JavaScript code as well as CSS code. This is an internal change, meaning that the code minimizer tool available in the IDE doesn't change anything, but internally it now uses NUglify instead of AjaxMinifier.

2023.50 (09/09/2023)

* The code minimizer tool is still trying to minimize the content.js file previously used for Web Extensions, but we don't use this file from various releases ago, and this causes the code minimizer tool to fail or show an error since that file did not exist. This release fixes this bug, and therefore the minimizer tool runs without errors (at least this error), as is expected. Other very minor changes, fixes, and enhancements have been made.

2023.49 (09/01/2023)

* Add the Splitted property for the Dropdown control. This new property (which is also documented in the product help and has been added to the Editor Quick Code List) determines if the Dropdown control must appear split into two buttons or not. If the Dropdown control is split, then the control appears divided into two buttons, and the user must press the right one in order to open the associated menu. If the Dropdown is not split, then it looks like before this change, and, the user can just press the unique button in order to open the associated menu. There is a new Dropdown3 sample app that shows how to use the new Splitted property of the Dropdown control.

2023.48 (08/29/2023)

* Update the jQuery JavaScript library to the latest version (3.7.1) in the app template, the program about page, program welcome page, program help pages, progam icons picker dialog, the included app plugins help pages and the program "readme" file. This new version of jQuery contains various fix, changes and enhancements.

2023.47 (08/28/2023)

* If we start the program without an Open AI API key defined in the program options, and then we establish an Open AI API key from the program options, and then we try to use the integrated Open AI assistant, we will experience an error, which basically means that we did not establish an Open AI API key. Once the program restarts, the behavior is as expected, but this release fixes the referred-to wrong behavior, so we can start to use the Open AI Assistant once we set an Open AI API key from the program options, that is, without the need to restart the program at all.

2023.46 (08/27/2023)

* Add a new app template help topic page in the product local and online documentation (and properly link it in the Index of Contents page) that describes the usage of the "v-if", "v-show", "v-for" and "v-html" directives that we can use in our apps interface, like in the HTML property of the HTML controls. These app directives are quite useful and powerful, as you will know when you read about it, play with it and finally start to use it.

* Add the TemplateHtml sample app, which shows how to use app global reactive variables, which can contain HTML markup, in order to be properly shown in our app interface by using the "v-html" directive. You can see that if we use the "v-html" directive, the HTML markup is shown as expected, but, if we use the moustache {{variable}} way the HTML is not shown as expected but "as is" in the app interface.

* Add the TemplateFor sample app, which shows how to use app global reactive variables with the "v-for" directive in our app interface. The "v-for" directive is intended to render Array variables in the interface, no matter if we are talking about simple Arrays or complex Arrays (like Array of Objects). It's quite useful, and the "v-for" directive is used in various sample apps, like the HTML sample app.

* Add the TemplateShow sample app, which shows how to use app global reactive variables with the "v-show" directive in our app interface. The "v-show" directive allows you to show certain content or other content depending on the value (true or false) of some app global reactive variable. The "v-show" directive renders the content that is intended to be rendered even if the variable in the condition is "false", but hides the content from the DOM by using "display: none".

* Add the TemplateIf sample app, which shows how to use app global reactive variables with the "v-if" directive in our app interface. The "v-if" directive allows to show certain content or other content depending on the value (true or false) of some app global reactive variable. The "v-if" directive doesn't render the content that is intended to be rendered if the variable in the condition is "false".

2023.45 (08/26/2023)

* Improve the Open AI assistant by providing a multiline control for the prompt.

This allows us to prepare a more convenient prompt, especially in some circumstances, like when we want to translate certain more or less large texts that can include multiple lines. Make other minor improvements, like trimming the prompt before sending it to Open AI.

2023.44 (08/18/2023)

* Update the Font Awesome icons to the latest 4.4.2 version. This means the update of the app template, the icons picker dialog, the Welcome page, the About page, the product Help and the included app plugins helps. The latest version of Font Awesome icons includes various fixes, changes and enhancements, for example, include the new "X" icon for the new Twitter logo.

2023.43 (08/17/2023)

* Introduce a new Icon Picker dialog for the IDE. The new Icon Picker dialog enhance the previous one in several ways! We can now get listed all the free icons of the latest version of Font Awesome (regular, solid, etc.). We can now also search for icons, but, not using the CSS classes of the icons: on the contrary, we use now all the search strings prepared by Font Awesome per every icon. This helps a lot in order to find the appropriate icon that we want to use, by filtering the icons that do not match our search criteria. Additionally the new Icon Picker dialog allows us to to determine the size of the icon that we wanted. We can also now choose one of the available icons animations to be optionally used. And finally we can choose one of the available icon rotations to be optionally used.

* Remove the "Fonts" folder from the installation. The "Fonts" folder contains the Font Awesome font files for Windows used by the previous Icons Picker dialog, but, these fonts are no more needed, because the new Icon Picker dialog do not needed it. Additionally, there is a process to "register" the referred fonts in Windows that is no more required, so this process has been removed too.

2023.42 (08/15/2023)

* After successfully adding a new app function, we now select the new app function in the functions list instead of selecting the first one. This change can be considered a bug fix since now we can add a new function and, just after that, add some arguments to that function without selecting the added function in the functions list. This helps us avoid adding the new argument to the wrong function: the first one in the list, which is not what we wanted. We want to add the argument to the new function, so this is what happens now: by selecting the new function in the functions list, the new argument is added to the right function.

* Fix an exception error if we try to search for new product releases when there is no internet connection. Now in this specific case, we get the right and expected error message (no host found; check the internet) instead of a weird exception that does not offer clear information about what is happening.

2023.41 (08/08/2023)

* Add the AutoClose property in the Dropdown control. The AutoClose property can be used to determine the auto close behaviour of the Dropdown control. The property have four possible values: "true" (by default), means the dropdown menu will be closed when the user press the dropdown button, a dropdown menu item or ouside the button and the menu. "inside", means the dropdown menu will be closed when the user press the dropdown button or a dropdown menu item. "outside", means the dropdown menu will be closed when the user press the dropdown button or outside the button and the menu. "false", means the dropdown menu will be closed only when the user press the dropdown button.

* Add the new AutoClose property of the Dropdown control in the contextual help file, providing the appropriate description and samples of use. Add the AutoClose property of the Dropdown control in the Editor Quick Code List. Add a new app constant "app.dropdownAutoClose" which contains all the possible values for the AutoClose property of the Dropdown control. Document this new app constant in the right help topic and add into the Editor Quick List. Add a new Dropdown2 sample app, which show the usage of all the possible values of the AutoClose property of the Dropdown control.

2023.40 (08/07/2023)

* Now the BAT files generated by DecSoft App Builder intended to compile apps with Apache Cordova, check for the existence of Apache Cordova, and, advise the user if the "cordova" command cannot be found, informing them also about the right URL of the Apache Cordova website to be ready with the selected platform. This is particularly useful for newbie users, who can find the "command not found" message from the console a bit annoying when executing one of these BAT files if Apache Cordova is not properly installed. Now these users (and others too) are properly informed about what happened and are pointed to the right documentation.

2023.39 (07/31/2023)

* Mark the app.setVar() method as deprecated, that is, remove it from the Editor Quick Code List and the documentation, in favor of the new app.setReactiveVar() method. The app.setVar() continue to be available to use, but, it's recommended to move on to the new referred method to set app global reactive variables.

* Add the new app.setReactiveVar() method, which is the indicated to set app global reactive variables. This method replace the app.setVar() method, which continue to be available for compatibility purposes. Advanced users can also directly use the "app.store" property to set app global reactive variables like: app.store.MyVar = MyValue. Please take a look at the product help in order to understand the difference between app global reactive and app global non reactive variables.

* Add the new app.getReactiveVar() method, which can be used to retrieve the value of previously established app global reactive variables. Advanced users can perfectly use the app.store property to retrieve these variables like:

app.store.MyVar, however, note that this method offer the ability to establish a default value to be returned if the variable did not exists. Please take a look at the product help in order to understand the difference between app global reactive and app global non reactive variables.

* Add the new method app.unsetReactiveVar(), which can be used to delete an existing app global reactive variable, so we can save precious resources for the app. Please take a look at the product help in order to understand the difference between app global reactive and app global non reactive variables.

* Add the new method app.setNonReactiveVar(), which is the indicated to set app global non reactive variables. We may need a global variable for the app, but, not required to be shown in the app interface, so we can use this method to create this kind of global variables, instead of reactive global variables, which consume a bit more resources. Advanced users can also directly use the "window.store" property to set app global non reactive variables like: window.store.MyVar = MyValue. Please take a look at the product help in order to understand the difference between app global reactive and app global non reactive variables.

* Add the new app.getNonReactiveVar() method, which can be used to retrieve the value of previously established app global non reactive variables. Advanced users can perfectly use the window.store property to retrieve these variables like: window.store.MyVar, however, note that this method offer the ability to establish a default value to be returned if the variable did not exists. Please take a look at the product help in order to understand the difference between app global reactive and app global non reactive variables.

* Add the new method app.unsetNonReactiveVar(), which can be used to delete an existing app global non reactive variable, so we can save precious resources for the app. Please take a look at the product help in order to understand the difference between app global reactive and app global non reactive variables.

* Add the previous referred new methods into the editor Quick Code List, and, update the product help in order to incorporate these new methods and some usage samples. Modify various existing samples apps in order to use the new methods instead the app.setVar() one or create by ourself a window.store variable, which is not created by the app to be available for us to store the app global non reactive variables.

2023.38 (07/26/2023)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.1 version for the program's welcome page, program's help, program's about, DecSoft app plugins helps and the apps template. This version of Bootstrap CSS includes a lot of changes, fixes and enhancements.

2023.37 (07/22/2023)

* We discover a piece of code in the app JavaScript template which correspond with certain Web Extensions related code that we are not using anymore from some releases ago. Probably this piece of code don't cause any trouble, but, may be

yes, talking about the Google Chrome and may others, so, we decide to remove that piece of code and publish a new release to avoid any possible errors which can occur due to that piece of code.

2023.36 (07/20/2023)

* Add in the program behaviour options a new button (inside a Configuration groupbox container) that can be used to restart the apps dock panels layout, the program dock panels layout and the program configuration at the same time. By pressing this button what we do is to initialize the apps dock panels layout, the program dock panels layout and the program configuration (at the same time), like if we start DecSoft App Builder for the first time. A confirmation is needed and the program need to be restarted when this new button is pressed. The product help has been enhanced by describing this new feature.

* Add in the program behaviour options a new button (inside a Configuration groupbox container) that can be used to restart the program configuration. By pressing this button what we do is to initialize the program configuration, like if we start DecSoft App Builder for the first time. A confirmation is needed and the program need to be restarted when this new button is pressed. The product help has been enhanced by describing this new feature.

* Add in the program behaviour options a new button (inside a Configuration groupbox container) that can be used to restart the program dock panels layout. By pressing this button what we do is to initialize the program dock panels layout, like if we start DecSoft App Builder for the first time. A confirmation is needed and the program need to be restarted when this new button is pressed. The product help has been enhanced by describing this new feature.

* Add in the program behaviour options a new button (inside a Configuration groupbox container) that can be used to restart the apps dock panels layout. By pressing this button what we do is to initialize the apps dock panels layout, like if we start DecSoft App Builder for the first time. A confirmation is needed and the program need to be restarted when this new button is pressed. The product help has been enhanced by describing this new feature.

* Add a new optional argument "fresh" in the DecSoft App Builder CLI compiler. The IDE tool which can create app BAT files to be compiled with the CLI compiler has been enhanced to add this new argument too. This new optional argument of the DecSoft App Builder CLI compiled can be used to compile the app in a fresh way, by removing, if exists, the "__cache" folder. The product help has been enhanced by describing this new argument and provide the right new screenshots of the CLI compiler and the Create app BAT tool dialog.

2023.35 (07/13/2023)

* Fix a bug which cause that the order of the app functions arguments cannot be changed: can be changed in the arguments list control, however, that changes are not then reflected in the app function, in other words, the argument list control show the arguments in the order that we wants, but, they are not properly saved in the function definition.

* Update the VueJS JavaScript library Router to the latest 4.2.4 version (debug and production) for the apps template. Update the Leaflet Maps library to the latest version: we use this library for the LeafletMaps sample app, which show how to use it, but, also, how to use (in general) JavaScript libraries in our DecSoft App Builder apps.

2023.34 (07/05/2023)

* Update the VueJS JavaScript library Router to the latest 4.2.3 version (debug and production) for the apps template. Update the VueJS JavaScript library Pinia to the latest 2.1.4 version (debug and production) for the apps template. Update the VueJS VueDemi JavaScript library (used by Pinia) to the latest 0.14.5 version (debug and production).

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

2023.33 (06/29/2023)

* Fix a bug which causes that any of the controls inside one or more containers inside app views, dialogs and frames, cannot be properly translated, even if we provide the appropriate translation for it using the app Languages manager. Now it's possible to place any number of containers and even containers inside other containers and the behaviour is the expected one: all the controls can be properly translated.

2023.32 (06/28/2023)

* We start to use the manifest version 3 when preparing the manifest files for the apps to be deployed as Web Extensions for modern browsers. It's very important to note, that, with the introduction of the Web Extensions manifest version 3 and the impossibility to use the "unsafe-eval" in the CSP (Content Security Policy) of this manifest version (which DecSoft App Builder apps need in order to use the VueJS runtime compiler), the DecSoft App Builder apps deployed as Web Extensions, must run now inside a kind of sandbox. This means that our apps, running as Web Extensions, cannot access the browser's local storage nor interact with the browser's page's content. We know that this represent a lose of functionality comparing to when we use the manifest version 2, however, the version 2 of the manifest files will be deprecated soon, and, therefore, we have no other option than the referred above: yes, we lose functionality, but, at least, can continue to deploy apps who will use the manifest version 3, of course, if we can go ahead with the referred limitations. We remove from the app object the methods that allow us to communicate with Web Extensions content scripts, since those methods have no sense anymore. We also remove the support to add content and background scripts from the Files Manager for the same reason. We also remove the WebExtensions samples apps, since they are prepared to run with the manifest version 2. We also remove from the product help the stuff which is related with the previous support of Web Extensions with

manifest version 2. We also remove the app events that are intended to use with this kind of Web Extensions. We leave a special previous release of DecSoft App Builder for our customers who want to continue using the Web Extensions manifest version 2 and their related stuff. You can find this special release by login into your DecSoft customer area at our web site, and then navigate to the Releases tab -> DecSoft -> App Builder -> Web Extensions manifest version 2. We encourage you to be ready with the manifest version 3, since the version 2 of the manifest files will be deprecated soon (they are already deprecated at the Chrome Web Extensions store). It's a pain when we lose functionality in our product, however, we can take the option to remove the support for Web Extensions at all, and, on the contrary, we work hard in order to keep their support, even with the referred limitations.

2023.31 (06/05/2023)

* Update the Bootwatch app themes (cerulean, cosmo, cyborg, darkly, flatly, journal, litera, lumen, materia, minty, morph, pulse, quartz, sandstone, simplex, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) which now are based in the latest version 5.3.0 of the Bootstrap CSS framework (as well the DecSoft App Builder default theme) and support the "light" and "dark" modes. The "Themes" sample app has been also enhanced and now offers a button to change between the "light" and "dark" modes.

2023.30 (06/03/2023)

* This release is a major version of the product. If you purchase DecSoft App Builder at or before 2022-06-01, you can purchase an upgrade license (with a 50% off discount) from your DecSoft Customer Area at our website, in order to register this release and the future minor releases for another year. Your current license does not expire, but it's now outdated: you can download the right outdated release also from your DecSoft Customer Area at our website. If you purchase DecSoft App Builder after 2022-06-01, you can also log in to your DecSoft Customer Area in order to grab your updated license serial number. Please, don't hesitate to ask us if you have any questions. Thanks in advance for your support and trust in our work!

2023.29 (06/01/2023)

* Enhance the Font Awesome icons picker in various ways. First of all, now the icon picker dialog offers all the available icons for the version of Font Awesome that DecSoft App Builder includes (the latest one). This is because now we are getting the list of icons from the "icons metadata" offered by Font Awesome. On the other hand, the classes for the icons are now the latest ones: even when the previously classes works, for example, the class "fas" works for solid icons, now we use the latest recommended way, that is, the classes "fa-solid", "fa-regular" and "fa-brands" instead of "fas", "far" and "fab" classes.

2023.28 (05/30/2023)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.0 version for the program's welcome page, program's help, program's about, DecSoft app plugins helps and the apps template. This version of Bootstrap CSS includes a lot of changes, fixes and enhancements. 2023.27 (05/28/2023)

* With the recent update of Apache Cordova Android the minimum Android SDK must be version 24 (Android 7) and the target SDK must be version 33 (Android 13). This release of DecSoft App Builder apply this requirements for new apps in the app Android options, and also add in both minimum and target SDKs combobox the API level 34 for Android 14.

2023.26 (05/19/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.4 version (debug and production) for the apps template. Update the VueJS JavaScript library Router to the latest 4.2.1 version (debug and production) for the apps template. Update the VueJS JavaScript library Pinia to the latest 2.1.3 version (debug and production) for the apps template.

2023.25 (05/13/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.2 version (debug and production) for the apps template. This version of the VueJS library includes various changes, fixes and enhancements.

2023.24 (05/12/2023)

* Update the VueJS JavaScript library Core to the latest 3.3.1 version (debug and production) for the apps template. Update the VueJS JavaScript library Router to the latest 4.2.0 version (debug and production) for the apps template. Update the VueJS JavaScript library Pinia to the latest 2.0.36 version (debug and production) for the apps template. Update the Luxon JavaScript library to the latest 3.3.0 version for the apps template.

* Update the jQuery JavaScript library to the latest 3.7.0 version for the program's welcome page, program's help, program's about, DecSoft app plugins helps and the apps template. This version of the jQuery library includes a lot of changes, fixes and enhancements.

* Update the Font Awesome JavaScript library to the latest 6.4.0 version for the program's welcome page, program's help, program's about, DecSoft app plugins helps and the apps template. This version of the Font Awesome JavaScript library includes a lot of changes, fixes and enhancements.

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.0 (alpha3) version for the program's welcome page, program's help, program's about, DecSoft app plugins helps and the apps template. This version of Bootstrap CSS includes a lot of changes, fixes and enhancements.

* The DecSoft app plugin "Analytics" has been enhanced in order to use the Google Analytics v4. This is important, because, the "Universal Google

Analytics, used by the previous release of the plugin, will be stop to recollect information in July 2023. Additionally, the current version of the "Analytics" app plugin add a new method "sendEvent" which allows us to send custom events to our Google Analytics account.

* Add the new "boolean" argument type for the app functions arguments dialog. We already can use the existing "number" or the "mixed" one, because in fact these are only for information purpose, but, for this same reason, we consider a good thing to have a "boolean" type for the arguments of a function, in addition to the existing string, number, array, object and mixed ones.

2023.23 (04/05/2023)

* Improve the app search tool. In addition to app functions, app events, app plugins events and controls events, now the app search tool perform the search also over the app Style property, the app Metatags property, the HTML property of the HTML control, and, the Rows and Header properties of the Table control.

2023.22 (03/30/2023)

* Perform some improvements over the app.hideDialog() and the app.hideAlert() app methods, by performing a better check of the involved DOM elements and avoiding any possible error during the execution of the methods if the specified dialog name, for example, do not exist when the app.hideDialog() method is called.

* Add the new app.getVisibleDialogs() app method, which can be used in order to find the currently visible (shown) dialogs. This method returns an Array with the names of the visible dialogs. We can use this method also to count (by using the returned Array length property) the visible dialogs.

* Add the new app.isDialogVisible() app method, which can be used in order to find if a specific app dialog is visible (shown). We no need this method before call the app.hideDialog() method, however, can be useful by itself in order to find if certain specific dialog is shown or not at some moment.

2023.21 (03/29/2023)

* Improve the Open AI assistant by changing some texts of the interface, but, mainly, adding a new tab which allows us to generate images from Open AI. So we can now generate source code, maintain a chat conversation and generating images using Open AI directly from your favorite software, DecSoft App Builder! :-)

2023.20 (03/28/2023)

* This is a maintenience release of the product which includes a very minor change related with the latest additions, which probably don't cause any problem, however, we want to publish this release in order to prevent any possible related unexpected behaviour.

2023.19 (03/27/2023)

* Improve the recently added Open AI assistant in order to get the replies from the Open AI API in a more friendly way, that is, character by character, or word by word, instead of wait for the entire reply to arrive. Additionally, add a Cancel button to the assistant which allow us to cancel the reply of Open AI at any moment.

* Improve the recently added Open AI assistant in order to keep the history (not all, but, a convenient part of it) of the a chat, so the Open AI recognize our conversation and can provide better answers. For example, if you provide your name to Open AI, the API will remember who are you and can refer to you by your name later.

2023.18 (03/26/2023)

* This release of the product incorporates an Open AI assistant (you can see it in the Menu -> Tools -> Open AI assistant) in order to be use the Open AI API directly from the IDE. You must provide an Open AI API key in order to properly work. The Open AI assistant can be used to ask for some code solution, like a "JavaScript function to validate EMail addresses", and, also to maintain a chat with Open AI, in order to get some other kind of solutions, like translated texts and many more.

2023.17 (03/25/2023)

* Update the Bootstrap CSS framework (CSS and JavaScript) to the latest 5.3.0 (alpha2) version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The new version of the Bootstrap CSS framework includes a lot of changes, fixes and enhancements.

2023.16 (03/24/2023)

* Fix a bug in the app.hideDialog() method while try to use it with a non existing or wrong dialog name or not shown dialog. Before this release, if we call to this method with a non existing dialog name, or if a dialog exists but is not shown, an error occur and the app behaviour can be affected.

* Fix a bug in the app.hideDialogs() method while try to use it with non shown dialogs. Before this release, if we call to this method and some of the available dialogs are not shown (which is the expected) an error occur and the app behaviour can be affected.

2023.15 (03/21/2023)

* Some minor changes has been made in some DecSoft App Builder controls in order to avoid some warnings while debugging our apps using the new "builder.debug.min.js" file. The same minor changes has been made also in some of the DecSoft app plugins included out of the box with the product.

* This release of the product incorporates a "builder.debug.min.js" file which is used while we are debugging our apps (F9). This file contains the non production versions of the VueJS, VueJS Router and Pinia libraries, which offers to us valuable information (in the internal developer console or any other browser developer console) while debugging our apps. This file is automatically used now by our apps when we are debugging it. On the other hand, the "builder.min.js" file, which contains the production versions of the referred libraries, is the one used when we build our apps (CTRL + F9) in order to be published, for example. This improvement in the product can be very useful while debugging our apps, since the non production versions of the referred libraries offers to us valuable information while debugging our apps.

2023.14 (03/09/2023)

* Add the new app.formatStr() method. This method is quite useful since allows to format a string easily. For example, it's possible to call the method with a string like "My name is {0} and my last name is {1}" and get a returned formatted string like "My name is John and my last name is Doe".

* Apply a background color and text color according to the used theme in the DecSoft BlockUI app plugin. This change means that the block UI message appears with the right background color and text color no matter if we are using the "light" theme color mode or the "dark" color mode in the app.

* Update the jQuery JavaScript library to the latest version. This means that the jQuery JavaScript library used in the app template has been updated, but also the jQuery JavaScript library used in the product "readme" file, the product About page, the product Welcome page, the product help and the "out of the box" app plugins helps.

2023.13 (02/25/2023)

* Update VueJS to the latest version (3.2.47). This latest version of Vue includes lot of minor changes, fixes and enhancements. Also update the Pinia JavaScript library to the latest version (2.0.32). Make some other minor internal changes, fixes and enhancements over the DecSoft App Builder IDE.

2023.12 (02/19/2023)

* Fixed a bug in the library used to work with the project XML files which causes that we can't properly retrieve previously saved data (in CDATA nodes, used, for example, to store the JavaScript code of the project) containing two closing square brackets without spaces: "]]". Sometimes it's possible to "simply" place some space between the brackets, but, the better way to solve this is the way in which this release act: using a fixed XML library which now have no problems to deal with the referred two closing square brackets.

2023.11 (01/28/2023)

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

* By some kind of "tradition" (probably from the arrival of XHTML long time ago) we start to use trailing slashes in void HTML elements, like images, breaks, horizontal rules, inputs, etc. That trailing slashes in such elements are not certainly required (talking about HTML), and, in fact, if we validate the HTML markup using the W3C validator, for example, we get an "info" reminder us that that trailing slashes for void elements are not really required in HTML documents. Well. We remove from our mind that "tradition" and therefore remove the referred trailing slashes from the DecSoft website and all our products. In particular, in this product we remove the referred trailing slashes from the HTML of the welcome page, about page, help pages, plugins help pages, app template, the proper app controls and finally the HTML produced by the app compiler.

2023.10 (01/26/2023)

* Enhance the app cache in order to properly remove cache files when we edit an event or modify a property of a control which is placed inside an app Frame view. Before this change we need to manually delete the entire cache if we modify an event or a property of a control if that control is inside an app Frame, which is something very annoying. This release properly remove the app cache files in the referred cases.

2023.9 (01/25/2023)

* Fix a bug introduced in the previous release of the product which causes that we cannot change the Classes property of a Frame control at designtime. Even more, when we try to run the app after change the referred property, we get an unexpected error dialog message which avoid to run the app as expected.

2023.8 (01/24/2023)

* Due to a mistake AB do not expose at designtime (like expected) the "Hidden" and the "Classes" property in the Frame control. Additionally, the "Classes" property are not take in consideration at runtime. This release fix these problems by propertly expose the "Hidden" and the "Classes" property at designtime and by apply the classes (if any) at runtime.

2023.7 (01/23/2023)

* Unfortunately we need to remove the SwipeLeft and SwipeRight events from the Table and the HTML controls. These events interfere with the horizontal scroll of these controls, and, we consider a priority to support (out of the box) the horizontal scroll and not the SwipeLeft and SwipeRight events. The app and views SwipeLeft and SwipeRight events still availables, and, on the other hand, since

the Hammer JS library is included out of the box, if we need the SwipeLeft and SwipeRight events in the referred controls (but no need the horizontal scroll) it's always possible to attach that events on that controls "by hand".

2023.6 (01/22/2023)

* Apply the app options max width and max height only for the app views and not app dialogs. This avoid certain dialogs screen mismatch and let the dialogs to appears always as expected: horizontal and vertically centered (from some releases ago).

* Add the new Fullscreen property to the app dialogs. This new property allows to show the app dialogs always in full screen, or to choose when to show the dialogs in fullscreen: at small, medium, large, extra large and very extra large screens. This new property is available at designtime and also at runtime, and, a new app.dialogFullscreen constant has been introduced with all the possible values for the property. The Quick Code List editor and the documentation has been updated to help with this property at runtime and to document it.

2023.5 (01/18/2023)

* Modify the behaviour of the app.showAlert() method. Now, you can provide a kind for the alert but an empty title. If you provide a title for the alert, then the optional kind is applied to the alert's title. If you leave the alert title to an empty string, then the optional kind is applied to the alert's body. This is useful if we want an alert with some kind but without a title. Before this change the kind is only applied to the title but never to the alert's body.

2023.4 (01/16/2023)

* We are particularly happy with this release since we discover a bug related with the Designer Comment control and the control cache files. Apparently the Designer Comment control is saved in the app cache, but, first of all, this have no sense, since this control do not have any style nor JavaScript associated. However, in fact we discover that we are saving in the referred control cache some JavaScript code which correspond to another controls, and, this cause a problem with that control, in the sense that changes made to that control are not reflected in the app, until we manually remove all the cache, including the "phantom" Designer Comment control cache. This release solve this bug and now not only we don't save in the app cache the Designer Comment control, but, also fix the problem related with this control cache, as is referred above, so the whole app cache works better than never, avoiding to remove the cache manually in order to get it working as expected.

* Add the Database2 sample app. This new sample app shown just another approach to deal with the "notes" database which we use in the already existing Database sample app. Despiste is a second version of the sample, maybe can appear more easy for newies, since don't use a more or less HTML control markup like in the Database sample app, but a Table control, in order to show the existing notes and allow to edit it, delete it, etc.

2023.3 (01/15/2023)

* The TableStyle and HeaderStyle properties of the Table control can be now set to any of the app.kind.* constant values instead of only "dark" and "light". We remove the app.headerStyle constant from the app template and properly document the referred control properties, as well to modify the Quick Edit Code list and the Table and Table2 sample apps to see these new possible styles in action.

* Add the new HeaderSticked property to the Table control. This property is false by default but can be set to true so the table header (if any) remain sticked. Enhanced the product help and the Quick Edit Code list, as well to modify the Table and Table2 samples apps to include this new control property.

* Start to use the new object fit related CSS classes of the Bootstrap CSS framework instead of the directly apply the object fit style in the Carousel, Figure, Image and Video Player controls. Modify the Image and Figure sample apps, the documentation of the referred controls and the Quick Edit Code list accordingly too.

2023.2 (01/09/2023)

* Enhance the HTML markup of the Progress bar control. Now the Progress bar control has the expected height (set at designtime). Also a new Caption (translatable) property has been introduced in order to allows to place some text inside the progress bar. Modify the Progress sample app in order to show these new capabilities of the Progress bar control.

* Use the Bootstrap CSS recommended "rounded-pill" CSS class instead of the "badge-pill" CSS class in the controls who uses a badge in someway. Use the "text-bg-* " CSS classes instead of the "bg-* " CSS classes and "text-* " CSS classes in the controls who uses these "bg-* " and "text-* " combination of CSS classes. Replace the "text-muted" CSS class used in the Figure control with the "text-body-secondary" CSS class.

* Remove the "DarkStyle" property of the Carousel control since has been deprecated for Bootstrap CSS 5.3.0 due to the new color mode of the framework. Remove the "btn-close-white" CSS class of the Toasts since has been deprecated for Bootstrap CSS 5.3.0 due to the new color mode of the framework.

2023.1 (01/01/2023)

* After 41 releases in 2022, here is the first release of the product in 2023. This is a major release of the product, means if you purchase your license before 01-01-2022, you must purchase an update license (with a 50% off discount) or continue using your outdated license with the last outdated release of the product, that you can download from your DecSoft customer area. Whish you the best for you and your family in this new year!

2022.41 (12/28/2022)

* Add a "fallback" redirection in the app router in order to back to the app main view (the first in the app views list) when we try to navigate to a non existing app view. This is the expected behaviour (go to the app main view) if a non existing app view route is used internally in the app or if we type in a browser adddress bar a non existing app view.

* Due to a mistake the Frame control have not exposed his Name property at designtime, so we cannot change it if we wanted and the name remains as who is set by default from the IDE. This release fix this bug and the Frame control name can be changed like any other control of the app.

2022.40 (12/26/2022)

* Fix a bug related with the Navbar control item click and brand click events. If the Navbar control is placed in non the main app view, we go to the app main view when this event are fired. This bug has been fixed by using the event prevent default for both events.

* Fix a bug with the Navbar control: even when the brand click event JavaScript code is stored as expected in the project files, it's not retrieved when load, so, certainly we miss this event JavaScript code when the app is loaded.

2022.39 (12/25/2022)

* Modify the ColorScheme sample app. Now, instead of using another app theme to establish a "dark theme" in the app, we use the new app.setAppThemeColor() method in order to set the app theme color to "dark". This is a bit optional, I mean, we can continue using different themes if we wanted, but, now we can also use the "dark mode" introduced by the new release of the Bootstrap CSS framework, which is included in this release.

* Add the new app.themeColor constant. This constant can be useful to use along the new app.setAppThemeColor() and the app.getAppThemeColor() methods. This constant is an object with two properties: "light" and "dark", which are the available values for the app theme color. This new constant has been also documented in the product help.

* Add the new app.setAppThemeColor() app method. This method allows us to establish the app theme color at runtime. The new version of the Bootstrap CSS framework, which is included in this release, introducing support for the "dark mode" and we can establish the app theme color at designtime, but also at runtime by using this method. This new method has been also documented in the product help.

* Add the new app.getAppThemeColor() app method. This method allows us to get at runtime the currently established app theme color, which can be "light" or "dark". We can establish the app theme color at designtime, but also at runtime, so this method allows us to retrieve the currently established app theme color. This new method has been also documented in the product help.

* Add the new ThemeColor app option. This option allows us to establish at

designtime the app theme color, which can be "light" or "dark". This is intented to use the "dark mode" supported now by the latest version of the Bootstrap CSS framework, which has been also included in this new release of the product. This new app option has been also documented in the product help.

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The new version of the Bootstrap CSS framework introducing support for the "dark mode" and we start to use it in the program's help, welcome, about and plugins help pages. Additionally this release enhance the help code editors (to show JavaScript code samples) when the program is running in dark mode.

2022.38 (12/23/2022)

* Add certain Bootstrap CSS class (modal-dialog-centered) to the app dialogs HTML markup. This is good for both scaled and not scaled (fixed) apps, in order to provide a better screen position for the dialogs. In the case of not scaled (fixed) apps we see now the dialogs vertically centered, instead of at the top of the screen. And for scaled apps with a max width and a max height, what we see is that the dialog fit the screen size instead of see how the bottom of the dialog exceeds the screen size a bit.

2022.37 (12/22/2022)

* Make some internal changes in order to replace certain editor components by others in various app options tabs. Additionally, this release uses a much modern dialogs when we need to select a folder path, for example, when we select a folder from the app Files Manager dock panel. This much modern folder select dialog offer us a better way to specify folder paths when needed.

2022.36 (12/21/2022)

* Enhance the internal app debugger server in order to properly handle the MIME type for JavaScript files, providing an exactly "application/javascript" MIME type for these kind of files. This is useful, for example, if we are debugging an app which uses an IFrame to show a Construct 3 game inside, since these kind of projects load JavaScript files as "modules" and a "perfect" MIME type is required to work as expected.

* Update the jQuery JavaScript library to the latest version. This means that the jQuery JavaScript library used in the app template has been updated, but also the jQuery JavaScript library used in the product "readme" file, the product About page, the product Welcome page, the product help and the "out of the box" app plugins helps.

2022.35 (12/16/2022)

* Update the jQuery JavaScript library to the latest version. This means that the jQuery JavaScript library used in the app template has been updated, but also the jQuery JavaScript library used in the product "readme" file, the

product About page, the product Welcome page, the product help and the "out of the box" app plugins helps.

* Add the new Table3 sample app. Very similar to the already existing Table sample app, but, this new sample show you some more features that we can use in the very powerful Table control, like have specific buttons in the cells, add or removing records of the table's data, show a message in the table in case that there is no table's data and more and additionally allow to filter the table's data using a text input control.

* Add the new Table2 sample app. Very similar to the already existing Table sample app, but, this new sample show you some more features that we can use in the very powerful Table control, like have specific buttons in the cells, add or removing records of the table's data, show a message in the table in case that there is no table's data and more.

2022.34 (12/12/2022)

* Due to a stupid bandwidth starvation attack to our server, we are obligated to perform some changes in how the product updater works. This means two things: this release made the right changes in order to get the updater working as expected after the referred changes, and, unfortunatelly, the previous release of the product cannot be updated using the product updater. We apologize for the possible inconveniences. If you are trying to update the product to this release using the product updater, please, just download the latest release of the product from our website.

2022.33 (11/30/2022)

* Add the "kind" property to the Checkbox input controls. This "kind" property is similar to the same property that we can found in another controls like the Push buttons, and, refers to the Bootstrap CSS framework "kind" colors that we can use, like "primary", "success", "danger", etc. The Checkbox input control do not have a kind by default, but, we add it to consider it useful for our apps. By default the "kind" is "primary", which let the control with a very similar apparience than the default one (without any kind). A new CheckboxKind sample app has been added, and, the editor Quick Code List includes the "kind" property now as well this property has been properly documented in the product help in the Checkbox input control reference.

* Add the "kind" property to the Radio input controls. This "kind" property is similar to the same property that we can found in another controls like the Push buttons, and, refers to the Bootstrap CSS framework "kind" colors that we can use, like "primary", "success", "danger", etc. The Radio input control do not have a kind by default, but, we add it to consider it useful for our apps. By default the "kind" is "primary", which let the control with a very similar apparience than the default one (without any kind). A new RadioKind sample app has been added, and, the editor Quick Code List includes the "kind" property now as well this property has been properly documented in the product help in the Radio input control reference.

* Add the "kind" property to the Switch input controls. This "kind" property is

similar to the same property that we can found in another controls like the Push buttons, and, refers to the Bootstrap CSS framework "kind" colors that we can use, like "primary", "success", "danger", etc. The Switch input control do not have a kind by default, but, we add it to consider it useful for our apps. By default the "kind" is "primary", which let the control with a very similar apparience than the default one (without any kind). A new SwitchKind sample app has been added, and, the editor Quick Code List includes the "kind" property now as well this property has been properly documented in the product help in the Switch input control reference.

* This release includes also some minor changes, fixes and enhancements, for example, the Radio control font size in the designer has been increased to be the same than what we use for the Checkbox controls and the Switch controls. Also, we remove from the editor Quick Code List the Size property for the Checkbox, Radio and Switch controls, since this property do not really exists for these controls: if the list include it is by a mistake.

2022.32 (11/22/2022)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. Update also the VueJS Pinia library to the latest version.

2022.31 (11/11/2022)

* Update VueJS to the latest version (3.2.45). This latest version of Vue includes lot of minor changes, fixes and enhancements. Make some other minor internal changes, fixes and enhancements over the DecSoft App Builder IDE and enhance, improve some of the included app samples.

2022.30 (10/28/2022)

* Add a new program debugger option which allows to automatically open the integrated browser developer console window when we launch the app. This option is not enabled by default, but the browser developer console window is quite useful for debugging the apps, so anyone who want this behaviour can check this new option: of course it's also possible to open the developer console using the right debugger toolbar button. Additionally, the browser developer console window is not closed now when we stop the running app: this can be useful to continue reading the console information even after the app is stopped.

2022.29 (10/27/2022)

* Update the Vue Router library to the latest version (4.1.6). Fix an introduced bug which causes an error if we use a dialog in the app which is named using upper case letters, for example, a dialog like "mydialog" works as expected, but a dialog like "MyDialog" causes an error which has been fixed in this release.

2022.28 (10/24/2022)

* Continuing with the app template improvements, in this release we move the "dialogs" HTML markup inside the app HTML markup, because now the dialogs and other app components (alerts and toasts) are child componentes of the app. Before this release the "dialogs" are in fact another root Vue component , but, making it childs components of the app provide to us various benefits, and, not to mention that in fact the dialogs and the other referred components are in fact part of the app, so there is no real sense to separate it in another root Vue component.

* Add the new app Instanced event. This event is only for advanced users and allows us to use the introduced "appInstance" variable, which stores the componenet instance of the app, and, can be used to setup some configuration in the app instance, as well to add custom components to it, for example. This method has been documented in the product help so you can read about it if you are interested.

* Add the new Instanced sample app, only for advanced users. In this new sample app we use the recently added app Instance event (and other new related events, as you can see below) in order to add a custom component to our app instance, so we can use that component like any others in our apps. Please, again, carefully read the documentation of the app Instanced event if you are interested.

* Add the new app Created event. This event is intended only for advanced users and allows us , for example, to prepare the object instance of a possible custom component which we can define in the app Instanced event. You can take a look at the help of this new app event as well the new app Instance event and the new Instanced sample app.

* Add the new app view Created event. This event is intended only for advanced users and allows us, for example, to prepare the object instance of a possible custom component which we can define in the app Instanced event. In this case that component will be available only for the app view in which we instantiate it.

* Add the new app dialog Created event. This event is intended only for advanced users and allows us, for example, to prepare the object instance of a possible custom component which we can define in the app Instanced event. In this case that component will be available only for the app dialog in which we instantiate it.

2022.27 (10/23/2022)

* Make various changes, fixes and enhancements in the app JavaScript template. We are now creating an instance of the app instead of directly working with the app root component. This allows us to recuperate the missing "OnAppRenderError" event handler (loses when we recently introduce Vue 3.x), as well to get the ability to configure the app instance, and, maybe in the future allow the user to configure it also or adding some filters for the HTML templates. Why not allow the user to add filters right now? Well! In fact we can already use any of the existing app methods as filters and the app user functions (which you can define) can also be used as filters in the HTML templates: in fact this is the way that Vue recommends to introduce filters in the HTML templates. But anyway

having an app instance ready to use allows us to maybe introduce some good things in the future.

2022.26 (10/22/2022)

* Update VueJS to the latest version (3.2.41). Use a more convenient releases of the latest versions of the VueJS router and the Pinia libraries. Update the Luxon library to the latest version (3.0.4). Update the Font Awesome library to the latest version (6.2.0).

* Since we surrounding now our apps views templates between the ` quotes, if we want to use this quotes inside an HTML control, for example, then the right HTML entity (`) must be used instead. This release of AB replace the referred quotes by the right HTML entities automatically, so we can use the quotes "as is" inside HTML controls, for example.

* Fix a bug in the Frame control events handlers: the app frames do not have events itself, however, when we use a Frame control to insert an app frame inside an app view or app dialog, we can handle common events (Click, DblClick, etc.). Before this release we can't do it due to a bad definition of the place in which the Frame control is placed.

* Peform some minor changes, fixes and enhancements in the app JavaScript template and the app HTML template, moving certain elements into the build.js file and reviewing mainly the JavaScript template from the scratch and make some code refactoring.

2022.25 (10/21/2022)

* This is a very special release of DecSoft App Builder and we are really proud to present it to you. What's new? Well. Starting from this release your app will use the latest release of the VueJS framework version 3.x, in addition to the latest release of the Pinia library (which replaces the venerable VuEx), in fact, what the hell? We are already using the latest versions of the Bootstrap CSS framework, the jQuery library, the Hammer library, the Luxon library, the base64 library and the FontAwesome library! All the best for you! (this is a new entry in the blog of DecSoft).

* In the previous release of DecSoft App Builder we are already using the latest version of VueJS 2.x branch, but in this release we move on and start to use the branch 3 of VueJS. This implied a better apps performance, a better supported framework (even when the branch 2.x of VueJS is still well supported), and, we are really happy with this release, because, after a hard work, we made it completely compatible with your current apps (see below the little breaking changes), that is, you can continue using DecSoft App Builder to compile your apps and now your apps become powered by VueJS 3.x.

* Breaking changes: There are only a few breaking changes. First of all, for this release we recommend to uninstall your previous version of DecSoft App Builder (not update it), and remove the files that you can find in your User Documents folder, at least the "Template" and the "Plugins" folders which you will find inside your User Documents / DecSoft / AppBuilder. In addition to

this, the only breaking changes has been the renaming of the AudioPlayer and VideoPlayer controls's properties: "duration", "pause" and "error", which are now "videoDuration", "isPaused" and "lastError". In the case of the AudioPlayer control the new "duration" is "audioDuration". In addition to this, also the methods "pause" and "play" of these controls are now "pauseAudio", "playAudio", "pauseVide" and "playVideo".

* There are no more changes! That is... internally... well... a lot of changes have been made, and, you will notice that your apps now run faster (yes even faster than before), but also internally better structured. We change everything in the app template and the app plugins (which are included out of the box), for example, we start to use more appropriately Ecmascript 6 Javascript code, but, again, except what has been referred above, you will no notice anything, you apps will continue compiling without any change! Of course, if you experience any problem, don't hesitate to contact us or post in our DecSoft support forum: we are happy to help you.

* Another advantage by these latest changes in DecSoft App Builder is that your apps can work now without require an HTTP server, that is, you can simply open the app's index.html file directly in a modern browser and will works as expected. This is because the app views and app dialogs HTML markup are stored now in the same app JavaScript core file, instead of in separated HTML files: this not only make the app faster (since no need to load external HTML files) but also allows you to run the app without the need of an HTTP server.

2022.24 (10/14/2022)

* Update the CSS themes (cerulean, cosmo, cyborg, litera, lumen, minty, sandstone, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) provided by Bootwatch (bootswatch.com) to the latest version. These themes matches now the version of Bootstrap CSS that we are already using in the app default theme and other stuff like the welcome, about, product help and plugins help pages.

* Add thew darkly, flatly, journal, materia, morph, pulse, quartz and simplex themes from the Bootwatch (bootswatch.com) package. Modify the Themes sample app in order to incorporate these new themes and add also some other controls to the sample app view.

* Update the Render Javascript library (VueJS) to the latest release. This release of the Render library includes various changes, fixes and enhancements.

2022.23 (10/08/2022)

* The app.showDialog() method has been enhanced to (optional) receive two more arguments for a couple of callback functions to be executed when the dialog is shown and hidden, respectively. This callback functions are optional and completely independent from the Show and Hide events of the app dialog. For more information you can take a look at the product help, in particular to the app.showDialog() help reference.

* Internally use the {once: true} argument for the event listener who we use when an app alert (shown by the app.showAlert() method) event "hidePrevented" has been fired. This prevent that internally the referred event will be executed more times that what is expected: apparently this is not a big problem, but, with the {once: true} argument can be perfectly avoided and that is what we do for this release of the product.

2022.22 (10/06/2022)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages.

2022.21 (09/20/2022)

* Update the CSS themes (cerulean, cosmo, cyborg, litera, lumen, minty, sandstone, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) provided by Bootwatch (bootswatch.com) to the latest version. These themes matches now the version of Bootstrap CSS that we are already using in the app default theme and other stuff like the welcome, about, product help and plugins help pages.

2022.20 (09/07/2022)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages.

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

2022.19 (09/05/2022)

* Make some changes in the Cordova config.xml file specifically for the Cordova Electron platform. Previously to this release we use a couple of platform=electron nodes in the Cordova config.xml file and this cause that the electron settings file are not taken and used as expected. We reported this issue to the Cordova Electron team, but, fix the problem by use only one node for the platform=electron platform. Make also a minor change / addition into the default / sample electron settings which we can set from the app Cordova Electron options, in order to hide by default the app window main menu.

2022.18 (08/28/2022)

* Update the jQuery Javascript library to the latest version in the app template, the product About, Welcome and Help pages, and also in the help pages of the DecSoft app plugins. This is also a mantenience release of the product with other minor internal changes, fixes and enhancements.

2022.17 (08/23/2022)

* Update the Render Javascript library (VueJS) to the latest release. This release of the Render library includes various changes, fixes and enhancements. This is also a mantenience release of the product with other minor internal changes, fixes and enhancements.

2022.16 (08/21/2022)

* Update the Render Javascript library (VueJS) to the latest release. This release of the Render library includes various changes, fixes and enhancements. This is also a mantenience release of the product with other minor internal changes, fixes and enhancements.

2022.15 (08/13/2022)

* Enhance the HTML markup for the toasts elements that we can use with the app.showToast() function (see the Toast sample app) following the very latest version of the HTML markup of the Bootstrap CSS framework. Fix also a bug which occurs also with the toasts when we press the close buttons.

* Fix a mistake in the ColorScheme sample app by adding certain configuration in the Apache Cordova extra config XML app option for the cordova-plugin-android-dark-mode-support Cordova plugin, so the plugin can work as expected in the Android platform. We forget such configuration, which is referred in the plugin's documentation and therefore the sample app are not properly working in the Android platform.

2022.14 (08/05/2022)

* Fix a bug which causes that the established value for the splash screen background color are not well taken when set the value of the "AndroidWindowSplashScreenBackground" key in the config.xml for Apache Cordova. Thanks Mario for the report!

2022.13 (07/26/2022)

* Update the CSS themes (cerulean, cosmo, cyborg, litera, lumen, minty, sandstone, sketchy, slate, solar, spacelab, superhero, united, vapor, yeti and zephyr) provided by Bootwatch (bootswatch.com) to the latest version. These themes matches now the version of Bootstrap CSS that we are already using in the app default theme and other stuff like the welcome, about, product help and plugins help pages.

2022.12 (07/25/2022)

* Update the Bootstrap CSS framework to the latest version: update for the app's

template, but also the program's help, welcome, about and plugins help pages.

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

* Update the animate CSS library to the latest version. Update the Font Awesome icons (Javascript library and desktop fonts) to the latest version. Update the Luxon Javascript library to the latest version. Update the VueJS, VueEx and VueRouter libraries to the latest versions.

* Due to the changes in Apache Cordova related with the splash screen implementation for Android, we encourage the usage of the latest version of Apache Cordova for Android (11.0.0), which requires the Android API level 32 (Android 12 L) , which is now the default option for the target SDK. Cordova Android 11.0.0 requires this target SDK API level in order to properly work, so we also carefully recommend to update your app to this API level: if you start a new app, don't worry, because this is the default option now.

* DecSoft App Builder no more prepare specific splash screens for various devices of the Android platform, but uses the provided splash screen image "as is", or the app icon, if no splash screen are provided. This change is because the splash screens on Android works now in a different way, using the Apache Cordova core instead of the Splash Screen plugin: this is mainly to offer support for the new Android API for splash screens in Android 12, but, the Apache Cordova core offer backwards compatibility for previous versions of Android.

* Do not use include anymore the Apache Cordova Splash Screen plugin for the Android and the Apple iOS platforms. The latest version of Cordova for these platforms includes the Splash Screen implementation in the Apache Cordova core and therefore this plugin is no longer required. Some additional changes related with the app Splash Screen as you can see above in this history document.

* Add new values into the app options for the Cordova Android platform and their Minimun SDK combobox and target SDK combobox, in order to provide support for the Android API level 31 (Android 12), the Android API level 32 (Android 12 L) and the Android API level 33 (Android 13). Set the default value for the target SDK to the Android 12 L (API level 32).

* Remove the Adobe Phonegap options tab from the app options dock panel, since the Adobe Phonegap project is no longer supported by Adobe, and, on the other hand, we always recommended to use Apache Cordova instead of Adobe Phonegap. Probably you never use Adobe Phonegap (and can't do it anymore), so this change do not affect to your apps. Also remove the Adobe Phonegap references in the product help and from some stuff that we are still added into the Cordova config.xml file.

* This is a major release of DecSoft App Builder, means if you purchase the product before 07-25-2021, you must to purchase an upgrade license (with a 50% off discount), or download the outdated release to continue using your outdated license. If you purchase your license after the referred date, please, login

into your DecSoft customer area to grab your new license serial number. You can purchase an upgrade license as well to download the outdated release from your DecSoft customer area in our website.

2022.11 (06/13/2022)

* Starting with a new app we can see that the Cordova AdMob plugin which we used before doesn't compile well with the latest version of Apache Cordova. After an investigation we found a certain modern and well supported Apache Cordova AdMob plugin which compiles and works as expected with the latest Apache Cordova version: this release modifies the CordovaAdMob sample app in order to use this new Cordova AdMob plugin.

2022.10 (06/08/2022)

* Add the PlaysInline property to the VideoPlayer control, so we can set this property to true or false (by default) at designtime and therefore add the appropriate attribute "playsinline" for the video HTML tag of the VideoPlayer control. This attrtibute can be useful, for example, to avoid the videos to be played at fullscreen on iOS.

2022.9 (05/31/2022)

* Due to a mistake the app.cordova.geolocation.getCurrentPosition() method of the Geolocation Apache Cordova plugin (which is integrated in the product) cannot receive the optional "config" variable as his third argument, which is supported by this method of the Apache Cordova plugin. This release fix this bug. Thanks Mario for the report!

2022.8 (05/30/2022)

* Due to some unexpected changes in our hosting the updater tool of the product stop to working as always. This release do the appropriate changes in order to allow the updater tool to continue working as always, that is, with the ability to check if a new version exists, to download and show the history of the product and to download an existing new release.

* When we add an app plugin into the app designer (for example any of the available DecSoft app plugins included by the installation of the product) automatically the IDE change the appropriate app option in order to include the right app plugin files, so we no need to do this manually (something which is always possible on the other hand). This release of the product do the same also in the case that we add an app plugin into the designer INSIDE a Container control. Before this release, if we add an app plugin into a Container control, the IDE do not check automatically the appropriate app option in order to add the required app plugin files, so the app plugin control don't work as expected, until the required app plugin is checked in the app plugins options. In few words, in this release of the product the IDE take care about the right app plugin option when we add an app plugin control inside the app designer, even if we do it inside a Container control.

2022.7 (05/09/2022)

* Add the new LabelSelectEx sample app, which show us how to use and the aspect of the recently added LabelSelectEx control from the "DecSoft controls" app plugin. This sample app is similar thant the SelectEx sample app, but using the LabelSelectEx control instead.

* Add the new LabelSelectEx control to the "DecSoft controls" app plugin. This control is similar than the existing SelectEx control, also from the "DecSoft controls" app plugin" but offer us another style based in a "floating label" which appear inside the control, in a similar way than the recently added LabelInput app plugin control.

* Add the new LabelSelect sample app, which show us how to use and the aspect of the recently added LabelSelect control from the "DecSoft controls" app plugin. This sample app show how we can use the control and show us the different aspect by changing their size property.

* Add the new LabelSelect control to the "DecSoft controls" app plugin. This control is similar than the existing Select control, but offer us another style based in a "floating label" which appear inside the control, in a similar way than the recently added LabelInput app plugin control.

2022.6 (05/08/2022)

* Add the new LabelText sample app, which show us how to use and the aspect of the recently added LabelText control from the "DecSoft controls" app plugin. This sample app show how we can use the control and show us the different aspect by changing their size property.

* Add the new LabelText control to the "DecSoft controls" app plugin. This control is similar than the existing Textarea control, but offer us another style based in a "floating label" which appear inside the control, in a similar way than the recently added LabelInput app plugin control.

* Add the new LabelInput sample app, which show us how to use and the aspect of the recently added LabelInput control from the "DecSoft controls" app plugin. This sample app show how we can use the different types of inputs that the control offers by changing the control's type property.

* Add the new LabelInput control to the "DecSoft controls" app plugin. This control is similar than the existing input controls (Text, Number, EMail, Date, etc.), but offer us another style based in a "floating label" which appear inside the control. The LabelInput control can act like text, email, number, password, color, date, time and search inputs, just by changing the control's type property.

* Fix a bug in the TypeAhead and the SelectEx controls from the "DecSoft controls" app plugin, which causes that the referred controls do not scale as expected (like the other controls) when the app window is resized by the user.

* Fix some introduced problem (releases ago) that causes that the app plugins help are not properly linked when we show it as a contextual help, for example, by pressing the F1 key when we select one of the available app plugin controls or one of their properties.

* The app minimizer code tool proceed also now to minimize the Javascript code of the the pluginscomps.js and the CSS code of the pluginscomps.css files which are included in the app when we use one or more of the available app plugins.

* Fix a bug which occur when the IDE want to load certain number of images (smaller and larger) of the available app plugins. Not only the bug has been fixed, but also, the IDE provide now more information in case that a problem occur when load that images.

2022.5 (04/25/2022)

* Change the way in which the DecSoft App Builder compiler set the "self" variable inside the events of all the controls, including native visual and non visual controls as well the app plugins controls, and, this all inside app views, app dialogs and app frames. There is a reported issue through the support forum related with the HTTP control and the "self" variable which occurs if we changed to another app view while the server response is not yet completed, however, I find that the same issue can happen also in another controls like the Websocket, AudioPlayer, VideoPlayer and Timer controls, so I decide to rewrite how the compiler deal with the "self" variable of all the controls events in a more accurate way. Thanks Mario for the report!

* Update the Base64 Javascript library used in the app's code to the latest version. In addition to other changes, fixes and enhancements, this new version of the Base64 library avoid the usage of the eval() Javascript function in order to properly work.

2022.4 (04/22/2022)

* Replace the Moment.js library (https://github.com/moment/moment/) from the app's core by the Luxon.js library (https://github.com/moment/luxon). The Luxon.js library is the recomended one by the Moment.js team to replace it and offers a much modern way to deal with dates and times in Javascript. Additionally the Luxon.js library is 300 Kb smaller than Moment.js. If your app already uses Moment.js, just download it an include to your app as any other Javascript file from the app's Files Manager.

* Add the Poster property to the VideoPlayer control. The Poster property is translatable and allows to establish an optional image path or URL to be shown in the video control before it start to be played.

2022.3 (03/06/2022)

* Make some internal changes in the app's Javascript template in order to fix a problem to use the app.showAlert() function inside the app's Mounted and DomReady event, as well the views' Show event. Thanks Paolo for the error

report!

2022.2 (02/06/2022)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The themes provided by Bootwatch has been also updated to the latest release.

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

* Add a new item into the App menu which can be used to remove the app source cache folder. Sometime we can experiment problems with the app source cache, so we can now use this new item in order to remove the app source cache folder, then we can compile the app from the scratch.

* Re-add the appropriate icon images to the items of IDE tabs contextual menu. This is in fact an omission from certain release in the past, when we add large images to prepare the IDE for 4K monitors.

* Add the new ChartJS sample app. This sample app shows how we can use the ChartJS library to create charts for our apps. The ChartJS library is probably the best Javascript library to create charts using Javascript. Additionally this sample app is another sample app which show us how we can add entire folders using the app's Files manager, and then link to the required files individually also from the Files manager.

2022.1 (01/01/2022)

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

2021.65 (11/30/2021)

* Our hosting silently starts to automatically redirect our HTTP calls to HTTPS and this cause problems while check for new releases of the product, for example, so in this release we add the appropriate stuff and enhance our internal procedures in order to use HTTPS by default, avoiding the problems. Thanks Ade for the report!

2021.64 (11/22/2021)

* There is a problem (probably in the project's view design XML) in the Container sample app which causes that the compilation cache don't work as

expected. This release fix that problem in the referred Container sample app so now we can see the compilation cache working as expected, as well the app, and, in particular the "button1" behaviour. Thanks Mario for the report!

2021.63 (11/08/2021)

* Select the Android 11 (API 30) of the Android target SDK for new created apps. You can still use an earlier API version without problems, but for new apps the selected target SDK is the Android 11 (API 30) by default for new apps.

* Fix a bug which cause that the cordova camera plugin destination type DATA_URL cannot be show into the editor Quick Edit List. Now we can select this destination type like the other available destination types from the editor Quick Edit List.

* Modify the CordovaCamera sample app in order to use the certain method of the Cordova File plugin (app.cordova.file.getFileBlobFromFileUrl) when we use the destination type = FILE_URL and want to show the photo that we take from the camera.

2021.62 (11/07/2021)

* The Login sample app is not working as expected because the HTTP link URLs do not point to the expected backend scripts. This error is fixed now and therefore the Login sample app works as expected again. Other minor changes, fixes and enhancements has been made in this release of the product.

2021.61 (10/25/2021)

* Fix a bug which disable the ability to establish the tabindex value = -1 at designtime for the PushButton, TextInput (all of them), FileInput, SelectInput, MultiselectInput, TextareaInput, Checkbox, SwitchInput, RangeInput, Dropdown, RadioInput and ImagePush controls. Thanks Desmond for the report!

2021.60 (10/24/2021)

* First release of the product compiled in Microsoft Windows 11. Fixed the detection of the Microsoft Windows version in order to be used internally at some places in the product. Set the rounded corners in the GUI forms when using the dark theme of the product GUI. Test the product at Microsoft Windows 11.

2021.59 (10/06/2021)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The themes provided by Bootwatch has been also updated to the latest release.

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and

64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

2021.58 (09/10/2021)

* Remove from the BAT files for Android the Apache Cordova WhiteList plugin, which is depcreated for the current version of Apache Cordova (10.0.0). We encourage to use the latest version of Apache Cordova. If you need to use a previous version of Apache Cordova, you MUST indicate this plugin usage by adding this line "CALL cordova plugin add cordova-plugin-whitelist" in the app's options -> Apache Cordova -> Batch files -> After.

2021.57 (09/08/2021)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The themes provided by Bootwatch has been also updated to the latest release.

2021.56 (08/29/2021)

* Use the "data-bs-parent" attribute for the collapsable items in the Sidebar control: this prepare the submenus of the Sidebar control to be auto-collapsables when another submenu is shown in the Sidebar control. Thanks Desmond for the idea!

2021.55 (08/10/2021)

* Now the body HTML tag of our apps have a CSS class attribute with the lowercase app's theme name. This class exists from the scratch, that is, once the app is started, and, it's updated with the properly lowercase theme name when we use the "app.setAppTheme()" method. This class in the body HTML tag is quite useful to apply CSS styles specifically to a particular theme and not globaly. For example, this CSS rule: body.slate select {} apply only to the app when the "Slate" theme is set, and not in other case.

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

2021.54 (08/07/2021)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The themes provided by Bootwatch has been also updated to the latest release.

2021.53 (08/01/2021)

* Fix an bug if we try to access the app variable (properties, methods, etc.) from an app user function called from the app's Mounted event. Before this fixed error we can't access the app variable (properties, methods, etc.) from an app's user function if this function is called from the app's Mounted event.

* Fix an annoying bug which causes that the IDE main form caption change to the number of app user function's arguments total and index when we click into the arguments list of the user function: this information is useful for the IDE internal debugging purpose, but not in other case, like in a program release.

* Remove Apache Cordova StatusBar plugin:
"app.cordova.statusbar.styleBlackTranslucent" and
"app.cordova.statusbar.styleBlackOpaque" deprecated methods. We can use the already available method "app.cordova.statusbar.styleLightContent()" instead of the deprecated methods.

2021.52 (07/13/2021)

* Add the new "darkStyle" property for the Carousel control, which is set to "false" by default. We can set this property to "true" in order to use a dark style variant into the Carousel control for the controls, indicators and captions of the Carousel control. This can be useful if the carousel's images are of light colors, so the dark variant can offer a better visibility of the controls, indicators and captions in the Carousel control. The Carousel sample app includes now the ability to set this new property to to true and false and the help / documentation as well the editor Quick Code List refer to this new control's property.

* Due to some change in the latest version of the Bootstrap CSS framework, the Carousel control don't work as expected with the before used HTML markup. This release enhance the Carousel control HTML markup and put it working again as expected with the latest release of the Bootstrap CSS framework, which is what AB use. Thanks Stefan for the report!

* The VideoPlayer2 sample app use certain third party URL for the played video, and, apparently that third party URL stop to work like before: the sample app has been updated in order to use another third party URL for the video, which works like expected at the time of this AB release publication: a note has been placed in the sample app "comment" to advert the user that these kind of third party URLs can stop to work without advise, so another URL (probably the right one for your own app) can be used instead. Thanks Stefan for the report!

2021.51 (07/03/2021)

* Add the new "Package type" option for the app's Cordova -> Android -> Sign options. This package type option allows to choose between APK and App Bundle, so now we can generate AAB (Android App Bundle) files of our apps. The Google Play Store only allows to upload AAB files from august 2021, so we are ready now to genereate AAB files for our apps. Note that the AAB files are only created when you provided a signing configuration from the app's Cordova -> Android ->

Sign options, and choose the "App bundle" package type option. It's possible to continue creating and signing APK files if we want.

2021.50 (06/23/2021)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The themes provided by Bootwatch has been also updated to the latest release.

2021.49 (06/20/2021)

* Take new screenshots of various program dialogs and dock panels for the help file (local and online versions), in order to show the latest changes make to the program dialogs and dock panels: many changes make over the referred dialogs and dock panels who are not properly reflected in the screenshots / captures of the help / documentation.

* In a previous release, when we prepare the product to be ready for 4K monitors and larger Windows screen scale sizes, we lost various images / icons for various dock panels, like the program options, about, welcome page, etc., as well various app specific dock panels, like functions manager, files manager, views list, etc., etc. This release put this panels icons / images working again like before.

2021.48 (06/19/2021)

* Update the app's help (local and online versions) to document the recently added ColorSchemeChange app event as well the new app.getAppColorScheme() method, which can be useful, for example, to change the app's theme depending on the configured system color scheme for apps.

* Add the new ColorScheme sample app, which show how we can use the recently added ColorSchemeChange app event as well the app.getAppColorScheme() method, in order to change the app's theme accordingly to the configured system color scheme for apps.

* Add the new app's ColorSchemeChange event, which is fired when the system color scheme for apps is changed at runtime, that is, while the app is running. We can use this event, for example, to change the app's theme accordingly to the configured system color scheme for apps.

* Add the new app.getAppColorScheme() method, which can be used to find the current system color scheme for apps, which can be "light" or "dark". This method can be useful, for example, to determine the theme that we can set to the app, depending on the configured system color scheme for apps.

2021.47 (06/18/2021)

* Include and use Microsoft Ajax Minifier instead of Yahoo YUI Compressor for the app's code minimizer tool: the Microsoft Ajax Minifier program offer us

better results because can handle more modern Javascript code as well CSS code. Nothing change in terms of the app's code minimizer tool usage: you can continue using the app's code minimizer tool (from the IDE or from the Command Line Compiler) like before, but internally we use now the Microsoft Ajax Minifier program instead of the Yahoo YUI Compressor program.

2021.46 (06/11/2021)

* Add the "disablePictureInPicture" and the controlsList="nodownload" attributes to the Video Player control, in order to avoid the "picture in picture" behaviour as well the download options wich the browsers provides. Modify the Video* sample apps also to show how we can avoid the browser's contextual menu for the Video Player controls just by "prevent the default behaviour" using the ContextMenu event of the control.

* Add the controlsList="nodownload" to the Audio Player control, in order to avoid the download options wich the browsers provides. Modify the AudioPlayer sample app also to show how we can avoid the browser's contextual menu for the Audio Player controls just by "prevent the default behaviour" using the ContextMenu event of the control.

2021.45 (06/08/2021)

* Add the API 30 (for Android 11) in the Minimum SDK and Target SDK select controls at the app's options -> Apache Cordova -> Android.

* Update the Render Javascript library (VueJS) to the latest release. This release of the Render library includes various changes, fixes and enhancements.

2021.44 (06/01/2021)

* Update the Bootstrap CSS framework to the latest version: update for the app's template, but also the program's help, welcome, about and plugins help pages. The themes provided by Bootwatch has been also updated to the latest release.

* Update the Render Javascript library (VueJS) to the latest release. This release of the Render library includes various changes, fixes and enhancements. This release of DecSoft App Builder includes other minor internal changes, fixes and enhancements too.

2021.43 (05/22/2021)

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

2021.42 (05/12/2021)

* Remove the NATIVE_URI "destinationType" for the Cordova Camera plugin, since it's no more supported. Add also the "app.cordova.file.wkWebViewConvertFilePath" method to the Cordova File plugin, in order to convert file URLs (only in iOS) to URLs ready to be used in Image, Audio player, Video player controls, etc. The method as been also included in the Cordova Camera plugin as "app.cordova.camera.wkWebViewConvertFilePath", for the case that we need only one of these plugins. Update the program's help with this latest changes.

2021.41 (05/10/2021)

* Add the new AceEditor sample app. This sample app not only show how we can use the Ace editor (https://ace.c9.io/) in our apps, but, also how to use a thirdparty library (and all the required files) in our apps, in a similar way than the already existing LeafletMaps sample app show for other Javascript library.

* Update the Bootstrap CSS framework (incuding the CSS and Javascript) to the latest 5.0.0 version. Update also the Bootwatch themes for Bootstrap CSS for this latest version and incorporates a couple of new themes: Zephyr and Vapor. Update also the Bootstrap CSS framework for the program's welcome page, program's help and program's about, as well for the DecSoft out of the box app plugins help.

2021.40 (05/03/2021)

* Add the ability to select a splash screen image for the dark modes to be used in the Android and iOS platforms. In the same way that AB generates several splash screens for the Apache Cordova platforms, now we can choose (optionally) a splash image to be used for the dark modes of the Android and the iOS platforms: then AB generates several splash screens to be used in that dark modes in the referred platforms.

2021.39 (04/29/2021)

* The latest release introduce some enhancements in the Cordova's configuration file around the "WKWebView" option, but includes it in a wrong way! This release fix the bug and add the right stuff in the Cordova's configuration file. Thanks Mario for the error report!

2021.38 (04/26/2021)

* Add a new "Use the modern WKWebView" checkbox in the app's options, under the Apache Cordova -> iOS options, which is checked by default. With this option checked (remember, it's checked by default for all the apps) we place in the Apache Cordova configuration file the right stuff in order to use the modern WKWebView for our iOS apps, instead the old WebView, which is no more accepted by the Apple Store. In some cases you may want to use the old WebView, then just uncheck this option.

2021.37 (04/09/2021)

* Place a small text hint when entering an empty (not yet used) event of the app, app views, app dialogs, native visual and non visual controls and app plugins visual and non visual controls. The text hints inform us about the specific event that we are seeing and also give us information about possible keys to be used.

* Remove an annoying flicker in the main toolbar Save All and Save buttons (basically the buttons alternates between the enabled / disabled states without apparently any reason). These buttons work as expected without any problem, however, the flicker referred to can certainly be a bit annoying distracting us about what we are doing.

* Assign the proper images (we lost this in a previous release) for the popup / contextual menus of the app views list, the app dialogs list and app frames list.

2021.36 (04/04/2021)

* This is the second major release of DecSoft App Builder, means if you purchase the product before 04-04-2020, you must to purchase an upgrade license (with a 50% off discount), or download the outdated release to continue using your outdated license. If you purchase your license after the referred date, please, login into your DecSoft customer area to grab your new license serial number. You can purchase an upgrade license as well to download the outdated release from your DecSoft customer area in our website.

2021.35 (03/29/2021)

* Continue with the efforts to make the program GUI ready for high DPI (Dot Per Inches) screen configurations. This release try to fix the item height for the control properties inspector and the control styles inspectors, as well the item height for the editor Quick Code List. Thanks Philip for your help!

2021.34 (03/28/2021)

* In this product release we make several internal changes and enhancements in order to prepare the GUI (Graphic User Interface) to be ready for 4K monitors, and, in general, for screens with a DPI (Dot Per Inches) configuration larger than the used when designing the product, that is, the Windows default DPI configuration. The menu and toolbars icons, the dialogs and all the GUI forms and controls have been reviewed and tested to achieve this objective. We are particularly proud of this product release, because the work we expend on it and because the results that we finally can obtain.

* In a previous release of the product we lost the ability to list in the integrated contextual help the available app plugins, as well their help. This release of the product reincorporates the app plugins list in the integrated

contextual help. Additionally, the app plugins help button from the app options works again as expected: we lost this ability also in the past due to some changes.

* We experience certain very, very rare bugs due to the lack of initialization for certain variables related with some events of the Edge Runtime WebView. This product release fixes these rare bugs by properly initializing the appropriate variables before trying to use it.

2021.33 (03/14/2021)

* Fix a wrong behaviour which occur when we launch the app using one of the possible available external browsers buttons in the program's main toolbar: the expected behaviour is that the app become launched in the external browser, but, also in the internal debugger browser. Before this change the app is launched in the external browser, but, not always in the internal browser, as is expected.

* Close the app's debugger panel when stop the debugger process by clicking the Stop button of the program's main toolbar. Previously to this change the internal debugger HTTP server is deactivated when clicking the Stop button, but, the debugger panel are not closed.

* Remove some hints from the app's debugger panel to avoid some problem (which do not occur everytime, but can cause that the program hang during a time) caused by that hints when the app is launched in the debugger browser (which is inside the debugger panel).

2021.32 (03/11/2021)

* The hint text (with some useful help) which appear when we select a method, property, etc., from the Editor Quick Code List, stop to work appropriately in certain past release of the product, due to certain change / update with the related used component. This release put the referred hint working again as expected. You can get more information about the Editor Quick Code List in the product's help.

2021.31 (03/07/2021)

* Update the app's template Javascript core and CSS style core, the program's welcome page, the program's help and the app's DecSoft plugins help with the latest versions of the Bootstrap CSS framework (5.0 beta 2), the jQuery Javascript library (3.6.0) and the FontAwesome icons Javascript library (5.15.2).

2021.30 (02/25/2021)

* Fix an error when create the shell files for iOS if our app contains custom plugins indicated by the app's Cordova custom plugins IDs. The compiler are not properly prefixed the plugins in the shell files by the appropriate "cordova plugin add" command, so the plugins are not properly installed nor compiled when

we compile the app with Cordova on Mac OS.

2021.29 (02/22/2021)

* We discover that the program still uses some stuff related with the previous apps debugger based on the Microsoft Internet Explorer WebView (used before we move to the new Microsoft Edge Runtime WebView). This release removes the referred old stuff, which is in fact no more needed.

2021.28 (02/20/2021)

* The program Welcome Page has been redesigned in order to be more responsive for different sizes and now addtionaly show more lastest forum threads and latest forum posts as well the latest entries from the DecSoft blog. No other changes has been made for this release of the product.

2021.27 (02/11/2021)

* Add the new Time input control, similar to the Date, EMail and other inputs control, the Time input control allows to specific ask the user for a time value (hh:mm). The help and editor quick list has been updated also to reflect this new control.

* Prevent an access violation error (not a fatal error) when press twice too fast the main toolbar Debug app button. Made the same prevention for the Compile button, even when apparently the error do not appear for this specific button. Thanks Asley for the report!

2021.26 (01/24/2021)

* Update the app's themes from Bootswatch.com (14 themes in total) to they last release, which has been made in order to support the Bootstrap CSS framework version 5.

* Add a new combobox control into the app Debugger dock panel toolbar which allows to resize the debugger browser by choosing one of the available screen sizes: based in the devices combobox which the toolbar shown.

* Add a new combobox control into the app Debugger dock panel toolbar which allows to resize the debugger browser by choosing one of the available devices: right now all the Apple iPhone, iPad and iPod devices, which can be taken as a good representation of different screen sizes.

* Add a new button into the app Debugger dock panel toolbar which allows to easily rotate the debugger browser, that is, changing from portrait to landscape and vice versa.

2021.25 (01/16/2021)

* Update the Microsoft Edge Runtime installer to the latest version. This update includes all the supported flavors, that is, the full installer for both 32 and 64 bits. This installer are internally used by the DecSoft App Builder installer to properly prepare the IDE to use the Edge Runtime webview for the apps Debugger browser.

2021.24 (01/10/2021)

* Due to the use of the Bootstrap CSS framework version 5 the Database sample app don't work as expected, since rely in certain CSS classes no more available in the referred version of the Bootstrap CSS framework. This release fix this bug and now the Database sample app works as expected, basically allowing us to select and delete existing notes.

2021.23 (01/03/2021)

* The Table control allows now to use the style "dark" or "light" in the same way than we previosly can use for the Table's header. Now it's possible to combine these styles in both the header and the table's rows itself. Take a look at the Table sample app to see it in action.

* Replace the "text-white" CSS class used in the Toast controls for the close button when "kinds" are others than "light" and "link" and use the Bootstrap CSS framework version 5 introduced "btn-close-white" instead. Use also for the close button the new "ms-2" CSS class instead of the previosly (and removed for this version of Bootstrap CSS) "mt-2" CSS class. Take a look at the Toast sample app to see it in action.

* Replace the "badge-* kind* " CSS classes in Push button and Dropdown controls by the appropriate "bg-* kind* " CSS classes: this change is required to be ready with the Bootstrap CSS framework version 5, who has removed the "badge-* kind* " CSS classes in favor of the "bg-* kind* " CSS classes. Additionally we now automatically apply the "text-dark" CSS class when the badge kind (left or right) are set to one of the "light", "warning", "link" and "info" CSS classes. Take a look at the PushButtons sample app to see it in action.

* Despite what the help reference mentions about the "ImageFit" property of the Figure control, the point is that that property is not shown in the control's designtime inspector nor is available in the Quick code list nor is working as expected. This release fix this bug and therefore the "ImageFit" property of the Figure control is now working as expected. Take a look at the Figure sample app to see it in action.

2021.22 (01/02/2021)

* This is the first major release of DecSoft App Builder, means if you purchase the product before 01-01-2020, you must to purchase an upgrade license (with a 50% off discount), or download the outdated release to continue using your outdated license. If you purchase your license after the referred date, please, login into your DecSoft customer area to grab your new license serial number. You can purchase an upgrade license as well to download the outdated release

from your DecSoft customer area in our website.

* Update the Bootstrap CSS framework to the latest version. This requires a lot of changes in almost all the components of DecSoft App Builder, due to the incompatibilities of the version 4 of Bootstrap CSS framework and the version 5 of the framework, that is, the latest version. You no need to worry, because all of these changes has been made internally in DecSoft App Builder and you no need to modify your apps at all. However, if you find some problem with this new release, please, report the problem to us using our support forum and we will try to help you as quickly as possible.

* Due to a mistake we are adding some CSS style for the designer Comment control. This control do not exists at runtime, only at designtime, so, the CSS style is no needed at all. So fix this bug to avoid any CSS style for the designer Comment control.

2021.21 (12/24/2020)

* Enhance the CordovaPushNotifications sample app in order to show how to include an icon for the push notification in the payload for Android. This requires to include an app's image and certain configuration in the Cordova Extra XML option. The sample server's code has been also modified in order to use the right payload with the new icon for the push notification.

2021.20 (12/17/2020)

* Fix a bug in the CordovaPushNotifications and CordovaDialogs sample apps, which by a mistake using a wrong way to call the alert method of the Cordova dialogs plugin. Another bug not related with this has been also solved in the CordovaPushNotifications sample app. Thanks Paolo for the report!

* Avoid to draw a bevel in the designer for the Push button and the Dropdown controls and remove the gray line from the Navbar control. Certainly the bootstrap referred controls don't show any bevel at runtime, so, this change at designtime look a bit more fidelis to the controls at runtime. Thanks Asley for the idea!

2021.19 (11/26/2020)

* Add the Background related CSS rules to be available in the style inspector of the Push buttons, Dropdown buttons and Image push buttons controls. The idea is to allows to leave the "kind" property of these controls "emptied" and apply the Background related CSS rules to the controls instead. Thanks Asley for the idea!

2021.18 (11/25/2020)

* Add the Input event to the Range input control. Instead of the Change event, which is fired when the user ends to slide the Range input control, the Input event is fired while the user slide the Range input control. Thanks Asley for

the idea!

2021.17 (11/24/2020)

* Add the new WebExtension5 sample app, which is a modification of the WebExtension sample app that uses the ability to add styles to our WebExtensions apps and show a custom contextual menu when the user right click in the browser's active tab page.

* In the same way that we can use the app's Files manager to add scripts specifically to be used in WebExtensions apps, this release of the product add the ability to add also styles specifically to be used in WebExtensions apps.

* Add the LineHeight CSS rule to be available in the Style inspector of various controls like the PushButton and HTML content control. We can always use the app's style or stylesheet to add any CSS rule, but, a customer of us wants this CSS rule to be available to be set from the control style inspector.

2021.16 (11/23/2020)

* Move the component used for the code editor to another version, which solves certain bug which causes that the app freezes if we try to use certain (very rare, but, useful) CSS rules inside the code editor for the app's style and other CSS code editors. Thanks Asley for the report!

2021.15 (11/21/2020)

* The App options -> WebExtension help topic has been updated to refer the way that we can follow in order to extend the browser's Developer Tools from our WebExtension apps. A new WebExtension4 sample app has been also added: this sample app is a modification of the WebExtension sample app, which now also adds a new panel into the browser's Developer Tools. Thanks John for the idea!

* DecSoft App Builder (in addition to modern HTML apps, WebApps, Progressive WebApps and Hybrid apps) produces also WebExtension apps for the modern browsers and this release of the product incorporate some changes and enhancements to provide a way to use a characteristic of the WebExtension apps: the ability to extend the browser's Developer Tools. Thanks John for the idea!

2021.14 (11/13/2020)

* Add three new buttons to the debugger toolbar in order to zoom in, zoom out and reset the zoom of the debugger's browser. There is also a new debugger option which store the currently selected zoom, to be applied the next time we open the debugger's browser. Thanks Peter for the idea!

* Apply some special Bootstrap CSS override to ".custom-file" which fix certain weird behaviour detected with the File input control in some specific browsers like Chrome and Opera: https://github.com/twbs/bootstrap/issues/32133 Thanks Peter for the report!

* In a previous release of the product we add a new Sidebar direction property, available to be view and set from the designer object inspector. This release include this option also in the app's Sidebar options, among the other Sidebar related options.

2021.13 (11/06/2020)

* Add the new RowDblClick event in the Table control. This event is similar than the RowClick event, and allows to prepare some Javascript code to be executed when the user double click or double tap the Table's rows. See the product's help for more information.

* Add the new Grid sample app, which shows how we can use the power of Bootstrap CSS (for example, the Grid system) in our apps, thanks to the HTML control, which allows to add virtually any HTML markup that we wanted.

2021.12 (11/05/2020)

* Fix a bug which happens when we use app relative paths in the Files manager for specific content and background scripts for WebExtensions apps. The problem occur because the "app/files" relative paths must take in consideration the "www" folder, so the scripts paths must be "www/app/files" and not just "app/files". We can continue to use "app/files" paths in the Files manager: the DecSoft App Builder compiler prepend the "www" folder for this specific kind of paths in the WebExtension manifest file. Thanks John for the report!

2021.11 (11/04/2020)

* Enhance the Sidebar sample app by adding a new button to toggle the direction of the app sidebar, as well to indicate the current direction of the app sidebar.

* Previous to this release the app sidebar is automatically hidden when the user swipe left over the sidebar. With the addition of the new sidebar direction property, now we use the swipe left if the sidebar direction is left, and, the swipe right, if the sidebar direction is right.

* Add the new "app.sidebarSetDirection()" method. This method can be used to change the app sidebar direction at runtime. Document this new method in the product's help.

* Add the new "app.sidebar.direction" property for both designtime and runtime. This new property allows to establish the direction of the app sidebar, which can be "left" or "right". Document this new property in the product's help.

2021.10 (10/24/2020)

* Update the VueJS Javascript library to the latest version 2.6.12. Update the VueJS Router Javascript library to the latest version 3.4.7. Update the VueJS

Router Javascript library to the latest version 3.5.1. Update the Hammer Javascript library to the latest version 2.29.1.

2021.9 (10/23/2020)

* Apparently the app VolumeUpButton and VolumeDownButton events, even if there has no code to be executed inside, can cause that the default volume controls GUI feedback don't appear as expected in some Android devices. This release fix that by avoid to attach the referred app events if there is no code to be executed inside. Thanks Asley for the report!

2021.8 (10/20/2020)

* Update the Microsoft Edge Runtime installer to the latest version. This latest version of the installer has been published in different flavours for Windows 32 and 64 bits, so, DecSoft App Builder also include this two flavours and use the right one for the product installer programs for Windows 32 and 64 bits.

2021.7 (10/18/2020)

* Fix an introduced bug in Boostrap CSS 4.5.1 which causes that some kind of modal dialogs (like when use some flavour of the app.showAlert() method) are not properly hidden when click (or press the Escape key) in the modal dialog backdrop. The Bootstrap CSS team knows about this bug and will fix it in a next release of the framework. Thanks Kiri for the report!

2021.6 (10/17/2020)

* Update the Boostrap CSS and JS to the latest version. Update also the 14 Bootstrap CSS themes provided by bootswatch.com. Other minor changes, fixes and enhancements has been made for this new release of the product.

2021.5 (10/01/2020)

* Fix a bug which causes that the apps created with an unregistered version of the product cannot be properly compiled with Apache Cordova on Mac OS for the iOS platform. Additionally, now the default deployment target for iOS is 11.0 (required by the latest version of Apache Cordova for iOS). Thanks Carlos for your error report and for your help!

2021.4 (09/26/2020)

* We move our products from davidesperalta.com to decsoftutils.com and therefore some URLs has been changed into this release of DecSoft App Builder. No more other changes has been made for this release of the product.

2021.3 (09/24/2020)

* The debugger developer console window are not automatically opened anymore when the app is launched. Now we must press a new button in the debugger panel toolbar or just press the F12 key, in order to open the developer console window.

* Fix a bug in the Typeahead plugin's control, which doesn't maintain the expected height, comparing with other inputs controls. Thanks Desmond for the report!

2021.2 (09/23/2020)

* Fix a bug in the debugger browser / panel size in case we have previosly installed AB when install the previous release. Basically the program get confused and don't check the configuration values for the browser / panel size. This release fix this problem. Thanks Asley for the report!

2021.1 (09/21/2020)

* This DecSoft App Builder release is a great step into the future, since, like our current generation of DecSoft HTML Compiler (recently published), say goodbye to the venerable WebView based in Internet Explorer and say welcome to the Microsoft Edge WebView2.

* Due to the usage of the Microsoft Edge WebView2, the internal debugger is no more an external program, that is, the apps are launched now in a dock panel in the IDE, more easy and faster. We continue having the Developer Console at our service anyway.

* By the way, as an user of DecSoft App Builder, maybe you want to take a look at the new generation of our DecSoft HTML Compiler product, which now can perfectly compile and convert modern HTML apps (for example, created by DecSoft App Builder) into standalone executables for both Microsoft Windows 32 and 64 bits. Get more information and download the demo version of DecSoft HTML Compiler in our website!

2020.100 (08/25/2020)

* Fix a bug which causes that we can't place a Visual Plugin control inside a Container control. Also fix a problem when we try to paste a Visual Plugin control inside a Container using the Clipboard. Thanks Desmond for the report!

* This release of the product have other minor internal changes, fixes and enhancements, for example, in the Debugger program.

2020.99 (08/05/2020)

* Move up the default value of the Android Min SDK option from the API level 19 (Android 4.4) to the API level 22 (Android 5.1). This is a requirement in order to be ready to use the latest version of Apache Cordova (10.0.0). Other minor

changes, fixes and enhancements has been made on this release too.

2020.98 (08/01/2020)

* Fix a bug in the Welcome Page which causes that the random help topic link which appears every certain interval of time, do not properly update the help topic's URL, causing that, when click in the random help topic link, we always ends in the same help topic location: always the first help topic which has been shown. Other minor changes, fixes and enhancements has been made on this release too.

2020.97 (07/26/2020)

* The app Text Search tool reduces a bit the searching feedback in order to increase very, very much, the search speed, which is now almost instantanlly. The app Text Search tool is faster than never before due to the reduction of certain small feedback (basically "where we are" in the search) that the Text Search tool shown in the past.

* The app functions manager show different popup (contextual) menus for the functions list and the function's arguments list. Additionally the arguments list popup menu, as well the functions manager toolbar, includes now two new elements that allows to move up and down the function's arguments.

* This release of DecSoft App Builder (the IDE, the Debugger and the Command Line Interpreter programs) has been builded using the latest version of Embarcadero Delphi, which is the IDE in which DecSoft App Builder is developed. The latest version of the Embarcadero Delphi's compiler introduces several fixes, changes and enhancements, so we get the benefits of these improvements.

2020.96 (07/24/2020)

* Add a couple of new options in the Cordova -> Android and the Cordova -> iOS options tabs in order to allows to specify the app's Cordova Android version as well the app's Cordova iOS version. By default these options are empty and this mean that the Batch and Shell files prepared by DecSoft App Builder uses the default Cordova Android / Cordova iOS version. We can specify here the versions to be used, so the generated Batch and Shell files clearly requires such specific versions of Cordova Android / Cordova iOS in order to be used.

* The Cordova iOS deploy target option is now set to iOS 11.0 by default. This means that we can be ready to use in our apps the WKWebView engine provided by Cordova iOS 6.x. Cordova iOS 6.x, however, requires a minimum deploy target of 11.0. Read more about how to use the WKWebView in the DecSoft support forum.

2020.95 (07/23/2020)

* Just attach the "BackButton" app's event causes a problem when try to use the Android device's back button to move around the app browser's history or just to exit the app (if no browser's history exists). This release fix this problem

attaching the "BackButton" app's event only if we write some code to be executed in that event. Thanks Letitia!

2020.94 (07/20/2020)

* Add the BackgroundAudio sample app (number 68) in order to show a way of use the app.PlaySound() method and the returned Audio JavaScript object in order to play a background audio in the app, which is not stopped even if we go to one app's view to another.

* Enhance the app.playSound() method in order to return the instance of the Audio JavaScritp object, so we can use the Audio JavaScript properties, methods and events if needed. Also fix the way in which we inquire to the platform about what kind of audio's format is supported. 2020.93 (07/18/2020)

* Update the CEF4Delphi components, which include the binaries of Chromium 83.0.4103.106 for Microsoft Windows 32 bits and 64 bits. These components and the Chromium binaries are used by the DecSoft App Builder debugger program.

2020.92 (07/10/2020)

* The app sidebar is hidden by default when the user press over the app sidebar overlay. In the same way, the app sidebar is now also hidden when the user swipe left over the app sidebar or the app sidebar overlay. Some other minor changes, fixes and enhancements has been made in this release of the product. 2020.91 (06/22/2020)

* Fix a bug / mistake which causes that the app's "BackButton" event are not fired like expected, since it's attached before the Apache Cordova "deviceready" event. Move the "BackButton", "Pause", "Resume", "VolumeDownButton" and "VolumeUpButton" app's events to the "deviceready" event, so they can be fired as expected. Thanks Asley!

2020.90 (06/16/2020)

* Move up the app plugins section in the app plugins dock panel. Doing this we want to place the app plugins section in a more relevant position, since the importance of it, in order to activate or deactivate the app plugins.

* Fix a mistake related with the app's plugins and the ability to add entire folders: we must indicate the plugin's relative path using URL separators (/) instead of Windows path separators (\). The documentation has been updated accordingly.

2020.89 (06/14/2020)

* Add the new FeedReader sample app (number 67). This app show how easy is to load and parse a RSS feed using the jQuery library, which, as you know, is ready for us in the apps out of the box. The app also shown other things like how to use the "selectEx" control, the HTML template syntax and the app's user

functions.

* Make some internal changes to allows app's plugins to link scripts and styles using a plugin's relative paths. This can be useful, for example, in order to use scripts and style files placed inside certain folder which is also added by the plugin. This help a lot in order to use thirdparty libraries from app's plugins, since the library's files can remain in the expected position in the file system. This is something that we already can do it from our apps, that is, link to app's relative paths, and, exactly for the same purpose. 2020.88 (06/12/2020)

* Update the Animate.css library to the latest version (4.0.0). This version of the library, among other changes, fixes and enhancements, add to DecSoft App Builder this new "in" animations: fadeInTopLeft, fadeInTopRight, fadeInBottomLeft, fadeInBottomRight, lightSpeedInRight and lightSpeedInLeft... and this new "out" animations: fadeOutTopLeft, fadeOutTopRight, fadeOutBottomRight, fadeOutBottomLeft, lightSpeedOutRight, lightSpeedOutLeft, backOutDown, backOutLeft, backOutRight and backOutUp.

2020.87 (06/09/2020)

* Added the new CordovaMedia sample app (number 66), which shows how to use the recently integrated Apache Cordova Media plugin. This sample app show how to record and play an audio, and also how to upload a recorded audio to the app's server.

* Integrate the Apache Cordova Media plugin. This is an official Apache Cordova plugin which allows to record audio files from the user's devices without the usage of any external recording app, like the integrated Apache Cordova MediaCapture plugin.

* Add the new ResponseType property to the Http client control. This property is an empty string by default, but, can be set now to "blob", in order to deal with the server's response as a blob. This can be useful, for example, to download and store files in the user's device using the "app.cordova.file.writeFile" method. Take a look at the CordovaFile sample app to see this new property and the referred method in action.

* The Apache Cordova File plugin has been rewriten from the scratch, in order to have a more powerful way to deal with files and directories. Additionally the plugin add various new methods like "createDir", "dirExists", "removeDir" and others. The CordovaFile sample app has been also rewritten and show the usage of all of the plugin's methods. The help file has been also rewritten to describe all the plugin's methods accordingly.

2020.86 (06/08/2020)

* Add the new PushButtons sample app (number 65), in order to show the power of the Push button control, including the recently added Active and Outline properties, but also the usage of other useful properties like the kind, badges and icons related properties.

* Add the Outline property to the Push button controls. This property works together the Kind property, and allows to use the outline style in the Push buttons. This property can be a boolean "false" value (by default) to not use the outline style, or can be set to "true" in order to use the outline style.

* Add the Active property to the Push button controls. This property allows, for example, to use the Push buttons as toggle buttons, since we can show the buttons activated or deactivated. This property can be a boolean "false" value (by default) to not use the active style, or can be set to "true" in order to use the active style.

2020.85 (06/07/2020)

* Update the jQuery JavaScript library and the Moment JavaScript library to their latest versions, which include various changes, fixes and enhancements.

* Improve the apps backups creation by overwriting possible existing files and by prefixing the backup folder names with a UNIX timestamp. The backup folders names have now a human readable local datetime, but, also a machine readable timestamp, for some possible further processing of these backup folders names from the IDE.

2020.84 (06/06/2020)

* The IDE creates now a backup of the apps everytime we modify something in the app and then save it. This is optional and can be enable / disable from the program's options / behaviour / history. The created backup contains all the app project files, ready to be recovered "as is" if needed. The backups are placed in an "__history" folder, which is placed in the same folder than the app's main project file resides.

2020.83 (06/05/2020)

* The app's resources list inside the app's Languages manager includes now a search input which allow us to filter the app's resources list, in a similar way than almost all the list in the IDE, which also allow us to filter it.

* Some minor enhancements in the IDE's welcome page, for example, the forum search form can deal now with the IDE theme, redirecting with the right theme to be used in the DecSoft forum website.

* Update the Fontawesome JavaScript library to the latest version: this version includes lot of new icons ready to be used in our apps. Update also the Fontawesome desktop fonts, which are used in the Icons picker dialog, and, finally udpdate the available icons lists in the referred Icons picker dialog.

2020.82 (06/04/2020)

* Add the new LeafLetMaps sample app (number 64). This sample app show how we can use the Leaflet JavaScript library to place a map in our apps. Additionally,

this sample app also show another interesting technique: include the Leaflet library by include his complete directory, and, then link to the right JavaScript and Style files by app's relative paths. This technique allows to include libraries which are designed to locate certain files (for example, images) inside the directory in which the JavaScript and / or Style files are.

* Add the ability to work with PHP files in DecSoft App Builder, in the same way that we can already work with HTML, CSS, JavaScript, XML files and other formats / languages. New entries has been added in the main menu, the toolbar files / menu and the IDE tabs popup menu. Also we can drag and drop PHP files into the IDE window in order to be opened. * Fix a minor bug which causes that the New Script dialog (used in the app's Files manager) don't show correctly the recently added "Is content script" and "Is background script" checkboxes, only the first time that the program is run.

2020.81 (06/03/2020)

* The app's designer's controls (inputs, selects, images, etc., etc.) take now in consideration the IDE theme which is used, so, if we use the dark theme, the controls appear in a dark color in the app's designer.

2020.80 (05/31/2020)

* After lot of attempts, finally we can get working a very useful feature: until now you can customize the app's dock panels layout, moving around the IDE window to place it in the position that you wanted, however, when the IDE was restarted or a new app is created, the app's dock panels layout become to the original (installation) positions. Now the IDE save the app's dock panels layout when the app is closed, and, load the previously saved layout when you open or start new apps.

2020.79 (05/30/2020)

* Introducing the DecSoft App Builder interface dark theme. We can continue working in our apps all the night! Now we can use the "light" or the "dark" theme, as we wanted. And it's possible to establish different program's options, like the code editor general and gutter options, the designer options, etc., in a separate way for the "light" and "dark" themes.

* The help pages of the product has been improved with an input search in every page, which can be used to search for contents in all the help pages. The changes has been also made in the online version of the help at: https://www.getappbuilderhelp.com/

* Fix a bug working with the code editor files which can cause a double save confirmation dialog to appear. In this release we get only one of this save confirmation dialog.

2020.78 (05/26/2020)

* The internal debugger browser can now be resized using the keyboard's arrow keys, so we can easily and accurately set the debugger browser size in pixels. The internal debugger browser show now also the browser size in pixels for your convenience.

* Add the Home, Go back and Go forward tool buttons in the internal debugger program window. We can use that button to re-load the main app's view and go back and go forward in the navigation history, respectively.

* The internal debugger program window is closed now if we press the ESCAPE key. Also the debugger incorporates now an address input so we can see the current URL and can change it to another one, for example, to show a specific app's view.

* Update the CEF4Delphi component of the product's debugger program, which is our bridge for the Chromium browser binaries, which has been also updated.

* Update the product's installer program (Inno Setup) to the latest stable version. This version fix some bugs and made some changes and enhancements.

2020.77 (05/22/2020)

* Update the Bootstrap CSS framework to the latest version (4.5.0). This release remove some Bootstrap themes provided by a thirdparty provider who do not update it to the latest version of Boostrap CSS. This release, on the other hand, updates the provided Boostrap CSS themes by Bootwatch. (Remember that it's possible to add your own Bootstrap CSS themes or modification in DecSoft App Builder, as easy as add the CSS file in your Documents / DecSoft / AppBuilder / Themes, so they become available for your apps).

2020.76 (05/21/2020)

* The dialogs Show event are fired now in the "show" modal event instead of the "shown" one: this allows to made some possible changes in some dialog's controls in order to be ready to appear (changed) when the dialog is shown to the user. Before this change we can see a small glitch until the controls' changes are effectively made.

* Made some minor modifications and enhancements in the recently added app.showAlert() app method. Now the alerts with the optional header kind different than "none", "link" and "light", shown the header text in white color, instead of dark color (for "none", "link" and "light" kinds). The optional alert's header text font size has been decreasing a bit too.

2020.75 (05/20/2020)

* Add the new app.showAlert() and app.hideAlert() app methods, which can be used to show (and hide) from simple alerts to the user, to more complex ones: using buttons, kind headers, HTML bodies, very larger bodies, callback called when the alerts are closed by a button and more: please take a look at the also new Alert sample app (number 63 of DecSoft App Builder) and the new app methods help.

* Set internally the modal-dialog-scrollable class for all the app dialogs: this help us in order to a better positioning of the dialog in the app. Before apply this class the dialog can be positioned beyond the app view (vertically), and, even when we can fix this by apply some style to the dialog, the point is that with the referred class we no need to do it anymore, if we don't wanted to establish certain specific height for the dialog.

2020.74 (05/19/2020)

* The app's Files Manager picker JavaScript files dialog allows now to include and check JavaScript files in order to be included and used as "content" and "background" JavaScript files, to be available for the app when it's compiled as a Web Extension. This new feature is only for Web Extension apps.

2020.73 (05/16/2020)

* Do not save anymore the "ModifiedDate" entry in the main AB project XML file: this can cause some non desired issue when using version control programs with AB projects, and, since it's not really used, I decide to remove it from the main AB project XML file. Thanks Amin for the report!

* Additionally I discover that the "CreatedDate" entry in the main AB project XML file is not well set when the app is created: this release also fix this, and now the appropriate value is set in this "CreatedDate" entry in the main AB project XML file.

2020.72 (05/12/2020)

* Fix various bugs in relation with the new recently added feature App Search: basically we take care now about the cache of the app, views, dialogs, frames, functions, native controls and plugins controls events, in order to be properly removed if needed.

2020.71 (05/10/2020)

* This product release incorporates some other internal changes, fixes and enhancements.

* Introduce the App Search dock panel, which allows to search in the app, views, dialogs, frames, functions, native controls and plugins controls events, and modify the code directly in the search results editor.

* Avoid to add new app user functions with a name of an already existing app user function.

* Add the new WebExtension3 sample app (number 63), with an interesting sample of Web Extension app developed with DecSoft App Builder. Thanks to John Clarke for provide this sample app to be included in the product installation!

2020.70 (05/01/2020)

* Add the new VideoPlayer2 sample app, which shown the usage of the hls.js JavaScript library in order to reproduce a video using the HTTP Live Streaming protocol.

2020.69 (04/10/2020)

* The "Input" property of the Label controls doesn't works as expected when we want to associate it with an "File input" control. As you can see now in the Label control's help, the "Input" property of the Label control, in order to be associted with a "file1" File input control, must be suffixed by "-input", for example, must be "file1-input". Thanks Desmond for the error report!

2020.68 (03/28/2020)

* Fix a bug which causes that the TAB key don't work as expected in the code editor. Thanks Michael for your report!

* Add the new Open Chars background and foreground options in the program options -> Editor -> General. Now the code editor of DecSoft App Builder highlight the open and close chars (parentheses, brackets, etc.) for our convenience.

2020.67 (03/08/2020)

* Fix a bug which causes that the Placeholder property of the FIle control are not loaded properly and therefore become to the default value every time. Thanks Peter for the report!

2020.66 (03/08/2020)

* By a mistake the previous release of DecSoft App Builder add the jQuery library and other stuff in the WebExtensions manifest. We can't do that, since the content script of the WebExtensions are loaded in every browser's page, and, that page can already contain the jQuery library, which can cause undesired and unexpected results. We fix such manifest mistake and also rewrite the WebExtension and WebExtension2 sample apps, in order to avoid the usage of the jQuery library, in this case.

2020.65 (03/04/2020)

* Add the new WebExtension2 sample app. This is a bit more complex than the WebExtension sample app, and, demonstrates how we can catch the submitted forms in all the browsers' tabs, so we can upload that information to our WebExtension server, for example. Please, note that this is sample app is just for learning purposes: you must carefully inform to your users if you wanted to recollect any kind of data from him.

* Change the WebExtension sample app in order to be ready with the latest changes related with the apps deployed as WebExtensions for the modern browsers.

* Add a new WebExtBackground app event, which allows to place JavaScript code to be executed in the background, until our WebExtension app is loaded and enabled in the user browser. Please, take a look at this new app event at the product's documentation.

* The app WebExtContent event is no more "fired", but, allows to write code for the WebExtension's browser's tab Content script. It's quite easy to continue to receiving messages in the WebExtContent event just by attaching the right event handler (see the WebExtension and WebExtension2 samples apps for more details, as well the documentation of this app event), and, in addition, we can know more things (in addition to just receive messages) in the browser's tab content script.

* app.webExtensionSendMessage() is now app.webExtensionSendMessageToActiveTab(). A new app.webExtensionSendMessageToAllTabs() method has been added too. See the documentation for more information and details.

2020.64 (02/22/2020)

* Remove the scrollboxes used in the app options dock form and try to made it (all the options tabs) in order to be ready for use in some specific screen resolutions. Some customers report that he can reach some parts of the app options dialog: this release try to solve this. The same changes has been made in the program options dock form.

2020.63 (02/18/2020)

* Fix a bug in the app Files Manager when we move up or down the position of an script or style sheet previously added: due to this bug we get the styles are filled with the scripts, which is wrong, and, of course, causes problems when the app run. This release fix the bug and therefore we can now move up or down as expected both scripts and styles.

2020.62 (02/09/2020)

* Made some additions in the documentation around the "app.setVar()" method. Basically we clearly establish that this method is used to set global reactive variables to be used in all app views, app dialogs and app frames. We extend the method documentation to add information about the global reactive variables, and, clearly establish that may such global reactive variables are not needed, and we can use just global variables instead, avoiding the cost in resources to maintain the referred global reactive variables.

* Fix a bug in the app.js template which causes that the app KeyUp event are not properly fired. It's a typo in the event handler name which has been fixed.

* Fix a bug which causes that the filter editbox of the app resources listbox

become unuseful, since once we write any filter, the resource's editor take the focus, and therefore we can't continue writting in the filter editbox. This release fix that problem.

2020.61 (02/04/2020)

* Added the Hide event for the app views and app dialogs. This event is fired when some app view is hidden in order to show another app view, or when an app dialog is closed / hidden to the user. This can be a good event to stop some possible running Timer control in both app views and app dialogs.

* Despiste the documentation, the "self" variable is not available in the app dialogs dblclickHandler, mouseupHandler, mousedownHandler, mousemoveHandler, mouseenterHandler, mouseleaveHandler and contextmenuHandler event handlers. This release fix this bug and set the "self" variable in such event handlers.

2020.60 (02/01/2020)

* Set the appropriate (and expected) TAB order for the app user functions new argument dialog, which helps to add and edit the app user functions arguments.

* The items of the app resources listbox and also the items of the app user functions listbox appear now alphabetically sorted, probably better than just the creation order.

* Fix a bug wihch causes that the hints of the app user functions (based in the functions arguments) appear wrong (always appear all of them!) and therefore become unuseful.

2020.59 (01/31/2020)

* Fix a bug wihch causes that the code in the app's ContextMenu event and the device's battery related events are not properly saved and therefore loaded nor fired as expected. This release contains also other minor fixes, changes and enhancements.

2020.58 (01/21/2020)

* Now, when the internal debugger browser window is resized, what happen is that the app panel changes his dimensions, and not the developer console panel. This described behaviour is what we can expected, and not the contrary.

* Avoid the usage of the cache in the internal debugger browser. The cache can cause unexpected results, since if we change certain used image, we want to show it, and not a possible previous one cached by the browser.

2020.57 (01/20/2020)

* Fix a bug related with the app functions cache which can cause some unexpected

results: you can experience one of them when modify an app function but can't see that latest changes working when run the app. This release fix this cache related bug.

* If we don't place any code in the app VueJS Error or the app JavaScript error events, DecSoft App Builder places an "alert" which show the possible errors. Now that "alert" has been replaced with a "window.console.error", useful but not annoying.

2020.56 (01/19/2020)

* Fix an introduced bug which causes that the contextual menus of the app views, dialogs and frames lists, can't work as expected. This release also provide some other minor fixes, changes and enhancements.

* Add the splash screen pìxels width and height options, as well the splash background color options in the Apache Cordova splash screen related options. This new options are effective in the Apache Cordova "Browser" and "Electron" platforms.

2020.55 (01/18/2020)

* Avoid to open the internal debugger browser window if we launch the app from one of the external browsers toolbar icons. Before the previous release the internal debugger browser was placed docked or inside the IDE, but now the internal debugger browser window is an external program (for very good reasons) and we can and even must avoid to open it if what we wanted is to launch the app in an external browser.

2020.54 (01/17/2020)

* The internal debugger browser has been enhanced and it's now based in Chromium (derived from Google Chrome) instead of Microsoft Internet Explorer 11. This allows to use ECMAScript 6 and even ECMAScript 7 JavaScript code in our apps without "syntax errors". Additionally, the internal debugger browser shows now a developer console, similar than in Google Chrome and others browsers. Enjoy it!
* * Note that the support for ECMAScript 7 is very limited today (Only Google Chrome and Opera browsers) and that ECMAScript 6 is not supported by Internet Explorer 11. Our recomendation is to use ECMAScript 6 if needed, because is supported in all modern browsers, except Microsoft Internet Explorer 11. If you need to support IE11, your code must remain to be ECMAScript 5, which is supported by all modern browsers too.

2020.53 (01/13/2020)

* The "nodeIntegration" option for the Cordova Electron platform is now set to "true" by default. This means we can use Node JS modules in our Cordova Electron build apps. Note, however, that the "require" JavaScript symbol is not available: instead you must use "nodeRequire" symbol. This is the way in which we can integrate the Node JS modules, since the "require" symbol can be

established by other libraries like jQuery.

2020.52 (01/12/2020)

* Creates a "package.json" file for Apache Cordova. This file is never required before for platforms like Android nor iOS, however, it's required by the Cordova Electron platform, and, since the file is missing, we get problems when build our apps using the Cordova Electron platform. This release fixed that problem by creating a "package.json" file ready to be used.

* Add the "--nobuild" flag in the "run" Batch and Shell files for the Cordova Electron platform. Electron has the ability to preview the app without build it first, and, so, the referred flag is the best to be used in the "run" Batch and Shell files. Also remove the usage of the Android Debugger Bridge in the Cordova Electron Batch files: have no sense there.

2020.51 (01/11/2020)

* Add the window.setTimeout(), window.setInterval() and window.clearInterval() methods into the Quick Code list as well in the help documentation. This methods are useful to prepare some code to be executed after certain amount of time, or in certain interval of time. We add also into the documentation the reference for the window.alert(), window.console.info(), window.console.debug() and window.console.error() methods.

* Add the new ImageFit property to the Video Player control. This property act exactly in the same way than the ImageFit of the Image control, for example, and, allow to control the video apparence, no matter the level of app scale.

2020.50 (01/06/2020)

* Add the new AppQuery sample app (number 59), which show how to use the recently added "app.query" property, in order to pass variables to the app views.

* Add the new "app.query" property, which stores the possible URL arguments passed to the app views or just a "false" value if no URL has been passed.

* Enhance the Table sample app with a designer comment wich describes a bit for what the Table control is intended to use, and what to look in the Table sample app.

2020.49 (01/05/2020)

* Add the new IconsMenu sample app (number 58), which show various things, for example, the usage of the new Image Push control, but, also, a very good icons menu approach for our apps, with differents app views dedicated to the app menu.

* Add the new Image Push control. This control combine the Push button and the Image controls in a sole control which looks like a Push button, with an image

inside. In fact it's possible to use a Push control and place an IMG HTML tag in his "text" property, in order to get something similar to the Image Push control, but, using the Push control we can see the choosed image at designtime, for example.

2020.48 (01/04/2020)

* Update the VueJS, VueJS Ex, VueJS Router and the jQuery JavaScript libraries to their latest stable releases. Made other minor changes, fixes and enhancements.

* Due to a bug the debug certificate password for the Electron platform are not properly load and saved, and, therefore, not well writen in the build.json file for that platform. This release fix that bug and now the referred JSON file is prepared as expected.

* The app.resource() method don't fail now if the specified app language do not exists. Instead of fail with a JavaScript error, the method returns the specified app resource in the default app language.

* Fix a bug in the NavBar control which causes that we can't set the "background" property to "dark" at designtime. The "background" can be perfectly changed at runtime, but not at designtime. This fix solve the bug.

2020.47 (01/01/2020)

* Add the Inputs sample app (number 57) which show how we can place controls beyond the app view height, which can be useful, for example, to place lot of inputs controls in a sole app view, and not divided in various of them. Additionally, this sample app also show the usage of the "app.setViewReadyForDeviceKeyboard()" and the recently added "app.setDialogReadyForDeviceKeyboard()", to be ready with the devices' keyboards in platforms like Android and iOS.

* Add the new "app.setDialogReadyForDeviceKeyboard()" method, which can be used for the same thing than the "app.setViewReadyForDeviceKeyboard()" , but specifically designed to be used with app dialogs, instead of app views.

* Change the internal implementation of the "app.setViewReadyForDeviceKeyboard()" method, in order to run properly from even when the first app view (before the app DeviceReady event). Now the method attach a "deviceready" event so works as expected even if we place it in the first app view.

* We find a unaceptable behaviour in the Android (and probably the iOS platform, even when we don't test it) when we place controls beyond the app view height, due to the attachment of the app SwipeUp and app SwipeDown events, as well the app view SwipeUp and SwipeDown events. For the moment, we remove these events from the app, so the behaviour in the Android platform become as expected. Since the Hummer JavaScript library (used by these events) are available in AB "out of the box", you can continue use the referred events if you wanted, by attaching yourself to the view or the app window.

2020.46 (12/31/2019)

* Made the appropriate changes in the app.setViewReadyForDeviceKeyboard() method to fit his help description: by a mistake this method continues to require the view name argument, but, the help explicit that we no need to set any argument, since the method uses the current view or dialog automatically. Now the method works exactly as is described in the help.

2020.45 (12/29/2019)

* Add the Container sample app (number 56), which show how we can use a Container control and an HTML control in order to prepare some kind of "headered panel", which can contains other controls like Text Inputs. It's a sample of a posible controls combination.

* Add the JSONSource sample app (number 55), which show how we can use a JSON file included in the app from the app files manager, in order to feed a Dropdown control, even including some arbitrary items' properties to be used internally by us.

2020.44 (12/28/2019)

* Add the Themes sample app (number 54), which show how we can include themes in the app to be availables at runtime, how to fill a Select control with all the available app themes, and, finally, how to establish a theme by using the "app.setAppTheme()" method.

* Enhance the app.setAppTheme() method in order to return "false", if the theme to be established are not available (because has not be choosed from the app options), and, on the other hand, return the name of the established theme, as well to set the "app.theme" with the established theme.

2020.43 (12/26/2019)

* Add the new IFrame2 sample app (number 53) which show how we can send messages from the app to some used IFrame and also how to receive messages in the app sent by the used IFrame.

* Add the new app event Message, which is useful to receives messages from a call to window.postMessage() in another browsing context, like an IFrame, that we can use in our app. So we can establish a communication with the IFrame and viceversa.

* The app Resize event are not properly set due to a bug, which is fixed in this release (the mistake is that we are assigning the Online event code to the Resize event).

* Place the missing "defaultText" argument of the method "prompt" of the Cordova Dialogs plugin, in order to be available into the Quick Code List and method's

help "hints".

2020.42 (12/19/2019)

* Add the new GetJson sample app (number 52). This sample app shown how easy is to get a JSON response from a server's script and then just access the JSON object properties like any other JavaScript variable, just by using the HTTP control's "response" property.

2020.41 (12/16/2019)

* The icons and the splash screens images which DecSoft App Builder prepares for the different Apache Cordova platforms, are not taking in consideration the program's behaviour configuration that determines these images compression level. This causes that sometimes we can get an Apache Cordova error while get a release APK for Android of our apps. Now this problem is fixed, by applying a default compression level for the referred images, and, respecting the configuration values for such compression level.

2020.40 (12/11/2019)

* Add the new Clipboard sample app (number 51), in order to show how to use the recently added DecSoft Clipboard app plugin.

* Add the new BlockUI sample app (number 50), in order to show how to use the new DecSoft BlockUI app plugin.

* Add the new DecSoft BlockUI app plugin, which allows to block the user interface until some background task are made, for example. The app plugin's documentation is available, like other app plugins, in the program's help index of contents.

2020.39 (12/09/2019)

* Various changes and fixes around the VueJSError and JavascriptError app events. Basically the "app.event" variable are not well set as null in the JavascriptError event. Now this variable is set to null as expected. On the other hand, the compiler don't place the "default fallback" alerts in these events, if you provide some code to be executed inside these events, in other words, the compiler only place default alerts in the events if you compile the app in debug mode and you don't place any code to be executed in the events.

2020.38 (12/04/2019)

* The Cordova Dialogs plugin, included "out of the box" in DecSoft App Builder, has now the proper fallback for all their methods ("alert", "confirm", "prompt" and "beep"), so the plugin fallback properly when the app run in non Apache Cordova platforms.

* Fix the CordovaDialogs sample app, which are not propertly follow the Cordova Dialogs methods arguments order, and therefore don't work as expected.

2020.37 (11/30/2019)

* Made some minor bug fixes in a couple of samples apps.

* Add the new CordovaBarcode sample app, wich show how to use the phonegap-plugin-barcodescanner plugin for Apache Cordova to read barcodes (various formats are supported) from our apps, using the device cameras.

2020.36 (11/29/2019)

* Update the Boostrap CSS framework to the latest version. Also update the themes from Bootswatch and the themes from Theme Machine. Thanks Asley for the advise!

2020.35 (11/26/2019)

* Fix a bug which cause that the File control appear visible in the app views even if his hidden property is set to "true". Thanks Michael for the report.

* Add the Android 10 (API level 29) into the app options (Apache Cordova -> Android). Set this option to the default target SDK for Android apps.

2020.34 (11/25/2019)

* Add the new CordovaAdMob sample app. This is another sample app who show how to use an Apache Cordova plugin, different than the already included "out of the box" in the product. The sample show how to show ad banners, interstitials and videos.

* Enhance a bit the App languages manager by fill the controls and control's properties listboxes once an app view, dialog or frame is selected in their respective listboxes. This basically save us a couple of clicks.

2020.33 (11/22/2019)

* Add the new Typeahead and SelectEx sample apps, which show how to use the Typeahead and SelectEx controls, added by the DecSoft Controls app plugin.

* Add the new Typeahead control. This new control is added by the DecSoft Controls app plugin, which is also included in the installation of the product.

* Add the TabIndex property to the SelectEx control. This control is included in the DecSoft Controls app plugin, which is also included in the installation of the product.

2020.32 (11/21/2019)

* The Progress control do not present his Name property in the control inspector. This release fixed this bug. Thanks Asley for the report! Some other minor changes, fixes and enhancements has been made also for this release.

2020.31 (11/19/2019)

* Add the new DecSoft Clipboard app plugin, which provide us a way to check if the clipboard is ready to use in the current browser / platform, and, also a method to copy certain specified text into the clipboard.

* Made some changes around the recent files and favorite files. Now, when open a recent file, that file become the first in the recent list, which is probably what we expected. Before this change the recent file don't move the position in the recent list.

2020.30 (11/16/2019)

* Add the new "CordovaPushNotifications" sample app, which show how we can use the phonegap-plugin-push Apache Cordova plugin, in order to receive push notifications in our apps, which are deployed in platforms like Android and iOS.

* Add the new DecSoft Analytics app plugin, which can integrate Google Analytics into your app. This plugin, as Google Analytics, works in all platforms and can give us a very good idea about how our app is used by the users.

* Add a button in the app options / app plugins tab, which allows to open the app plugins help directly from that place. The IDE already link to the available plugins help in the index of contents of the product help, however, the referred button can be also a good thing.

2020.29 (11/13/2019)

* Fix a mistake in the Label control, which causes that the Classes cannot be properly set, since AB try to set a "classes" argument for the HTML element, but the right name is "class", not "classes". Thanks Asley for the report!

* Change the default settings for the Cordova Electron Cordova, in order to avoid the usage of the NodeJS modules. Use the NodeJS modules is possible, however, requires another changes in AB that we are not sure right now how to fight. It's possible to solve this (if you want to use NodeJS modules right now) by editing the app "index.html" and adding the stuff related in this article: https://electronjs.org/docs/faq#i-can-not-use-jqueryrequirejsmeteorangularjs-in-electron

2020.28 (11/12/2019)

* We are very proud to introduce the first DecSoft App Builder release which has support for thirdparty app plugins. In this new generation of DecSoft App

Builder, we want to go ahead, and, the app plugins can be perfectly integrated in the IDE, can offer global stuff (useful methods for the app, for example), and also components / controls, which play in the same league than the "out of the box" controls: can be edited using the controls properties and style inspector, and, of course can have events, ready to be edited in the IDE Javascript code editor, just like any other "out of the box" controls. Support thirdparty plugins become a hard work, but, we are happy with the final result, and, in fact DecSoft App Builder already incorporates the first plugin, provided by DecSoft, a "SelectEx" control, similar to the "Select" control which is already available, but, including certain useful feature, as you can see in the app plugin's help, which is also integrated in the IDE, as you can see in the new sections of the help Index of contents. You can get more information about the app plugins in the right help topics too: https://www.getappbuilderhelp.com/help/en-US/app-plugins.html https://www.getappbuilderhelp.com/help/en-US/app-plugins-dev.html There are lot of internal changes to made the app plugins possible. You can see now, for example, a new tab in the app options form, specially dedicated to the app plugins: you can get information of the plugins from that new app option form tab, as well activate and deactivate plugins, when needed (some of them can be auto-activated by the IDE, for example, if you use an app plugin's control, that plugin become activated, so everything is prepared when compile the app, in order to place the app plugin stuff available).

* Other minor changes, fixes and enhancements has been made also in this release, while we work in the app plugins integration.

2020.27 (11/09/2019)

* Fix a bug in the assigment of the app views events: by a mistake not all the app views events works as expected and this release fix this issue.

* Prepare the currently available Apache Cordova plugins stuff (basically the information that the plugins add into the code editor Quick list) to be translated in the future into another languages.

2020.26 (10/25/2019)

* Some minor changes, fixes and enhancements. Perform the default fade / slide animation for the apps dialogs, which appear too much better than leave it without these feature. Allow to edit the app languages from the App languages manager. We can now change both language codes and names. Also in the App languages manager, indicates the language of the original piece of code to be translated, and, the language that is expected in the translation, according to the currently editing app language.

2020.25 (10/21/2019)

* Add the App resources into the editor Quick list, which means that we can now easily see the available App resources and also insert the right "app.resource()" method in the code editor in order to get the desired App resource value.

* Fix various bugs and mistakes in the App files manager, which get confused when deal with Javascript and stylesheet files, causing a malfunction of the App files manager in various aspects. Now the App files manager works as expected.

* Add the new sample app "Pannellum", which show how to use the Pannellum library (the tour mode, in particular) in our apps. Pannellum is a lightweight, free, and open source panorama viewer for the web. Built using HTML5, CSS3, JavaScript, and WebGL, it is plug-in free. You can find more information at: https://pannellum.org/

2020.24 (10/19/2019)

* Introducing the Key Store files tool (IDE Main Menu -> Tools -> New Key Store file), which allows to create the Key Store files to be used to sign and additionally publish our Android apps in stores like Google Play. Document the tool in the program help file too.

2020.23 (10/17/2019)

* Add into the editor Quick list the information of the recently added app languages related methods. We forget to do this in the previous release. Made some minor changes in the app help also, to get a more usable list of the available app methods.

2020.22 (10/16/2019)

* Other lot of other minor changes, fixes and enhancements from the previous release of the product, no one to be expressly mentioned here, but a bunch of them.

* Add the new "app.getLanguageNameFromCode()" method, which returns a specific app language name from the specified language code.

* Add the new "app.getLanguageCodeFromName()" method, which returns a specific app language code from the specified language name.

* Add the new "app.getLanguagesCodes()" method, which returns the available app languages codes. This method can be useful, for example, to fill a Select control with the available app languages codes, to the user can choose one of them.

* Add the new "app.getLanguagesNames()" method, which returns the available app languages names. This method can be useful, for example, to fill a Select control with the available app languages names, so the user can choose one of them.

* Add the new "app.getLanguages()" method, which returns the available app languages in form of an Array of objects. Every object has two properties, "code", with a language code, and "name", with a language name.

* Add the new "LanguageName" option (and property at runtime) for the app. This "LanguageName" must match the already available "Language" option (and property at runtime), which stores the default or designing app language code. The "LanguageName" is useful not only to be used "as is", but, because it's also take in consideration when prepare the "applangs.js" Javascript stuff.

* Include the new App Resources and the App Languages managers in the produc help, as well the other new stuff that has been introduced in this product release.

* Include the new applangs.js Javascript file (which include the app languages resources, sidebar and interface languages) in the Minimize code tool, so this file is also minimized as well other app files like the main app.js.

* Cache the languages Javascript files generated by the App Resources and the App Languages managers. These files can be properly required everytime the app is run (so we can't wait for a build, for example), but caching the files certainly prepare it to be available causes a minimum impact when compile the app.

* Add "Search" inputs for the app views, dialogs, frames, controls and properties ListBoxes controls, in the App Languages manager. As usual in the DecSoft App Builder IDE, we can use these ListBoxes to filter what is show in the respective ListBox.

* The new "app.translateView()" method translate the current app view or app dialog (including the possible used app frames) into the specified "app.language", if that language is available and ready to be used. You can see this method in action in the also new "Languages" sample app.

* The new "app.resource()" method returns a specific resource value from the App Resources. This method is "language aware", that is, the method take in consideration the value of the also new "app.language" variable, so the resource is retrieved in the right language, if translated, or, the original value, if no translation is available.

* Introducing the new App Languages manager. The App Languages manager provides an easy to use way to translate our app interface into another languages. The App Languages manager offers a convenient editor to translate the app controls placed in app views, app dialogs and app frames. The App Languages manager also provides a way to translate the App Resources, and, finally, also provides a way to translate the optional app sidebar.

* Introducing the new App Resources manager. The App Resources manager give us the ability to establish one or more arbitrary Javascript resources (strings, objects, arrays, JSON) for the app to be available from the scratch when it's run. We can use the also new "app.resource()" method in order to get a specific app resource in both Javascript and HTML code (for example, to show a message in an alert, but, also to show a message inside an HTML control).

* Fix an introduced bug in the compiler message log that can cause that an app with hundreds of controls can take a while to be compiled, too much than the required one. This release fix that possible annoying problem.

2020.21 (10/11/2019)

* Fix an error that occur if we check one or more Apache Cordova plugin in the app options, and, after that, try to use the code editor. The problem occur because some expected files cannot be found, since the installation assistant are not including it. This problem has been solved in this release and the appropriate files are shipped. Thanks Phap Duy for the report!

2020.20 (10/10/2019)

* Made other minor fixes, changes and enhancements, including the fix of certain bug in the Frame sample app. Thanks Samuel for the report!

* Enhance the Android "run" Batch file in order to start the debug console just after our app has been launched in our attached Android device. Doing this we no need to open a new command line window and type the right command to see the debug console, but, the "run" BAT do it automatically for us. The debug console is quite important to debug the apps with an attached Android device, since offers to us the same information than the browsers's debug console offers: valuable information to debug our apps.

* Remove the SwipeUp and SwipeDown events from the HTML and Table controls. Unfortunately, these events block the scroll over the referred controls, but these controls are intended to place several contents, that is, they are commonly "scrollables". The app and the apps views and dialogs SwipeUp and SwipeDown events are maintained and can be used if needed.

2020.19 (10/06/2019)

* Add the new App options (local storage) app methods category, which include the new app.setOption(), app.getOption(), app.removeOption() and app.clearOptions() methods to deal with the app local storage: save, read, remove and clear, respectively. Thanks Peter for the suggestion!

2020.18 (10/05/2019)

* Hide the Navbar items menu after the user press in a Navbar item or subitem. Before this change the Navbar items menu still visible and therefore the Navbar toggle button must be used to hide the Navbar items menu.

* Move (link in the app index.html) the app user scripts (added by the app Files manager) before than the app.js main file, so there are availables in certain scenarios that we need to use it inside app.js.

* Add the new "itemIndex" and "itemSubindex" properties to the NavBar control. These properties are "-1" by default, but are filled with the right data inside the Navbar "ItemClick" event, and can help to identify the item or subitem which has been pressed by the user.

* Fix a bug in the app Files manager, that causes that we can't add scripts nor

stylesheet files, since the IDE offers to us a wrong open file dialog type, which disallow to choose the right files to be added.

* Change the arguments order of the Cordova Dialogs plugin methods. Basically the methods who accepts a title argument requires this argument before the dialog callbacks arguments, so we can set or not the callbacks but use titles, in a more easy way.

2020.17 (10/01/2019)

* Change the program graphics and icons to the new DecSoft graphics and icons. This is the very first public release of the new generation of DecSoft App Builder. This release is the first one available to be downloaded by anyone from the DecSoft website from today.

2020.16 ALPHA RELEASE (09/30/2019)

* Add the new Sidebar sample app, which show how to use the app sidebar control in various ways, from designtime to runtime.

* Document in the program help all the new stuff: app properties, methods, events, options, etc. Carefully read the help to learn about the new Sidebar component and more!

* Add the new Sidebar items dialog, which allows to establish at designtime both the items and subitems of the optional app sidebar, including all the default possible properties of that items and subitems, like their text, icon, kind, class, actived and disabled properties.

* Add the new app "SidebarItemClick" event, which is fired when the user click in the app sidebar items and subitems. Add the new app "SidebarHeaderClick" event, which is fired when the user click in the app sidebar optional header.

* Add the new "app.showSidebar()" method, which is intented to show the optional app sidebar to the user. Add also the new "app.hideSidebar()" method, that can be used to hide the app sidebar. Add the "app.sidebarIsVisible()" method, that can be used to check if the app sidebar is visible or not.

* Add all the new app sidebar options into the IDE designtime controls inspector for the app: we can directly use the IDE controls inspector to set all the app sidebar related options, including the items and subitems.

* Add the new app sidebar options tab in the app options dialog. This app options allows to establish the app sidebar options, including items and subitems, at designtime.

* Add the new app sidebar property, which stores the optional app sidebar options at runtime. The app sidebar property is an object that stores the app sidebar items and subitems, as well the header text, header kind, header align, image URL and more.

* Add the new app sidebar component. The app sidebar component allows to show a

menu with any number of items and subitems in all the app views. The app sidebar is optional, and, is completely customizable at both designtime and runtime.

* Add the new "app.randomNum()" method, which returns a random number from 0 (zero) to the specified max number. If no max number is specified, returns a random number from 0 to 100.

* Fix a mistake in the Figure control, which causes that only the Load and Error events can be fired, but not the others like Click, DblClick, MouseUp, etc. Now all the events are properly set and then properly fired.

* Fix a bug which causes that the "light" kind of the Push button cannot be properly set nor used at runtime: instead the "primary" kind is used. Now the "light" kind is used if we set it.

* If a Javascript error occur in the main "app.js" file, the DecSoft App Builder internal debugger, after show the error message, point us to the right line of the "app.js" file, so we can take a look at the error. We made some changes here in order to be sure that the editor with the "app.js" file fit the entire parent form, not partially, as somethings occurs (it's difficult to reproduce the problem). Thanks Samuel for the report!

2020.15 ALPHA RELEASE (09/27/2019)

* Add the ViewsTransitions sample app, that show how we can use the new TransitionName of the app views. Add also the ViewsAnimations sample app, that show how we can also use the out of the box available animations.

* Add the new TransitionName and TransitionMode properties to the app views. These properties allows to establish our custom CSS transitions to be applied to the app view, in addition to the out of the box provided animations that we can use.

* Fix a small problem with the Code Minimizer dialog in case of an error occur. Link also this dialog with the new IDE Code Minimizer help topic.

2020.14 ALPHA RELEASE (09/25/2019)

* The program help has been enhanced with the new toast and strings related methods: carefully read these methods help reference in order to properly learn how to use it.

* Add the recent added toast and strings related methods into the editor Quick code list, in order to be available for your convenience when write the required Javascript code.

* Add the new Toasts sample app, which show how to use the recently added toasts related methods, as well use the main app.showToast() method in various ways, show how to change the toasts notifications properties, etc.

* Add the new app.getToasts() method, which allows to retrieve all the currently showing toast notifications previously show by using the app.showToast() method.

* Add the new app.getToast(), which allows to retrieve some specific toast notification information, as well to change the toast properties like text, kind and many others, reflecting these changes automatically in the toast notification.

* Add the new app.hideToasts() method that can be used to hide all the currently showing toast notifications previously shown by using the app.showToast() method.

* Add the new app.hideToast() method that can be used to hide a specific toast notification previously show by using the app.showToast() method.

* Add the new app.showToast() method, which allows to show one or more toast notifications to the user. It's possible to show more than one notifications because they are vertically stocked. This method is quite powerful (carefully read their help reference), since can be called just with a toast notification text to be show, but also admits other useful arguments in order to control how the toast notification is show, if the toast notification is clicked or dismissed, etc.

* Add the new app.randomStr() method, which returns by default a random string of 10 character length, but can also be used to retrieve a random string of a specified length.

* Rename the "app.strSerialize()" method to "app.serialize()". Rename the "app.strUnserialize()" method to "app.unserialize()".

* Add the new "Always show Welcome page" checkbox in the program behaviour options. This option is checked by default and can be used to show the program Welcome page at start, no matter if we previously hide the Welcome page before the program is closed. If this option is unchecked, the program respect our decision and don't show the program Welcome page if we hide it. We always can show the program Welcome page from the Main Menu -> View -> Welcome page.

2020.13 ALPHA RELEASE (09/24/2019)

* Improve the compiler cache by removing the right cached files when we made some change in the controls properties inspector and the controls style inspector. Before this change the cache files are not removed in some specific properties changes, so we can't see the new properties values when compile the app and launch it, since the previous cache files are used. Thanks Samuel for the report!

* Allow to set some custom style for the NavBar control using the controls style inspector. This cannot be made before due to a mistake: certainly this control position cannot be set using CSS style (commonly), however, this don't means that we can't set another possible CSS rules for the control, so now it's possible to do it if needed.

2020.12 ALPHA RELEASE (09/23/2019)

* The app.setViewReadyForDeviceKeyboard() method has been changed in order to only apply the appropriate view's style if the app is running in a Cordova platform, for example, in an Android device. In other words, the method don't do nothing (the expected) if the app is running in a browser.

* Allow to establish the app MaxWidth and MaxHeight properties separately. This can be particularly useful to set an specific app max width, but not an app max height: these properties are intended to limit the auto scaled style of the app, so now we can limit the app width only, but leaving the app height occupying all the screen height.

* Fix a bug in the HTML markup of the Table control, that, between other things, causes that the table's width can't fit the screen with the expected 100%.

* Add the SwipeUp, SwipeRight, SwipeDown and SwipeLeft events for the app, the app views, the app dialogs, and, the HTML and the Table controls. These events are fired when the user made the up, right, down and left gestures using the mouse (browsers) or the finger (devices).

2020.11 ALPHA RELEASE (09/22/2019)

* Add the new RightBadgePilled property to the Push button and the Dropdown controls. This property determines the if the optional right badge property must be shown a bit more rounded and left and right padded.

* Add the new RightBadgeKind property to the Push button and the Dropdown controls. This property determines the style of the optional right badge property. Useful to use in combination with the Kind property of these controls.

* Add the new RightBadge property to the Push button and the Dropdown controls. This property allows to put certain text or number at the right of the button Text property.

* Add the new LeftBadgePilled property to the Push button and the Dropdown controls. This property determines the if the optional left badge property must be shown a bit more rounded and left and right padded.

* Add the new LeftBadgeKind property to the Push button and the Dropdown controls. This property determines the style of the optional left badge property. Useful to use in combination with the Kind property of these controls.

* Add the new LeftBadge property to the Push button and the Dropdown controls. This property allows to put certain text or number at the left of the button Text property.

2020.10 ALPHA RELEASE (09/21/2019)

* Fix the app Metatags property help link, so we are pointed now to the right help topic if press the F1 key inside this app property or their editor.

* Fix a bug which causes an Invalid Pointer Operation error when close the program, if previously we open the "shared.css" app file using the appropriate

option in the IDE.

2020.9 ALPHA RELEASE (09/20/2019)

* Add up to 43 "out animations" app constants values, which can be set (at both designtime and runtime) to the new app view "outAnimation" property, in order to perform certain animation when the app view is hide to show another one to the user.

* Add up to 37 "in animations" app constants values, which can be set (at both designtime and runtime) to the new app view "inAnimation" property, in order to perform certain animation when the app view is show to the user.

* Prepare a "share.css" stylesheet to be linked in the app after the app theme and the scaled or fixed styles. This "shared.css" stores the app's Custom Style and doing this change we avoid to duplicate that Custom Style in both scaled and fixed styles. Put in the Main Menu -> App an element to open this shared.css file, in the same way that we can already open the fixed.css and the scaled.css files.

* Made certain change in the configuration file for Apache Cordova in order to allow us to use the "edit-config" configuration tag inside the app Cordova -> Extra XML option. It's possible to use the "edit-config" before, but, not specifically to made changes in the AndroidManifest.xml file, or, at least some changes, like adding attributes to the "application" tag. Now it's completely possible to perform this kind of changes.

2020.8 ALPHA RELEASE (09/19/2019)

* Add the new app.strSerialize() method, which allows to serialized in a string an Array or Object variable. Add also their counterpart, the app.strUnserialize() method, which returns the Array or Object variable from a previously serialized string.

* Add the new app.strToBase64() method, which allows to get the Base64 representation of an string. Add also their counterpart, the app.base64ToStr() method, which returns the decoded text from a Base64 encoded string.

2020.7 ALPHA RELEASE (09/18/2019)

* Made certain changes in order to get the Web Extension apps working as expected. A new WebExtension sample app has been added, to see a Web Extension app working in the supported browsers, like Mozilla Firefox and Google Chrome.

2020.6 ALPHA RELEASE (09/17/2019)

* Add the new Database sample app, in order to show how easy is to implement a CRUD app using DecSoft App Builder. The Database sample is a small app, but, we can see how to show database records, how to update it, how to insert new records and how to delete it.

* Made some other fixes, changes and enhancements, like to properly link to the help from a couple of app user functions dialogs.

* Fix a bug which causes that the internal debugger browser shown annoying scrollbars when the app is launched. Other minor changes, fixes and enhancements has been made too in this release.

* Fix a bug which causes that the app dialogs cannot contain uppercase characters. The appropriates changes has been made in order to allow this uppercase characters in the app dialogs names.

* Fix a bug when try to deal with some "chrome" stuff (related with the app Web Extension capabilities) in the browser: the bug occur in Microsoft Edge and not in other browsers.

2020.5 ALPHA RELEASE (09/16/2019)

* Modify the Navbar sample app in order to use the new Navbar control related stuff. Update also the program help to reflect these changes.

* Allow the "icon" property to be used in the Navbar control items, in the same way that is available to be used with the Navbar subitems.

* Add the new Navbar items and subitems dialog editor, which allows to prepare the Navbar control items (and subitems) property at designtime.

2020.4 ALPHA RELEASE (09/15/2019)

* Add some new constants to the app (document it and add it to the editor Quick code list) related with the new Navbar control.

* Document the Navbar control in both the program's help and the editor Quick code list.

* Add a new Navbar sample app, which show how to feed the Navbar control with items and subitems, as well how to react to events and set other Navbar properties.

* Add the new Navbar control, which can some any number of items and subitems in a navigation bar placed at the top or the bottom of an app view.

2020.3 ALPHA RELEASE (09/14/2019)

* Made other some minor changes, fixes and enhancements.

* Fix a mistake in the Audio player sample, which causes an app error due to try to use a non existing control propety / variable.

* Add the sound related app methods: app.beep(), app.playSound() and app.stopSound(). Thanks Samuel for the idea!

* Add the string related app methods: app.lowerCase(), app.upperCase(), app.strLen(), app.trimStr(), app.strSearch(), app.splitStr(), app.subStr(), app.strReplace() and app.strReplaceAll(). Thanks Samuel for the idea!

2020.2 ALPHA RELEASE (09/13/2019)

* Fix the URL for the new releases to be downloaded, taking in consideration the version (32 bits or 64 bits) of the installed program.

* Prepare the right program manifest to made the DecSoft App Builder IDE DPI awareness. Thanks Samuel for the error report!

* Now is possible to use the mouse drag and drop to move the items (controls) of the Controls Tab Index dialog. Thanks Asley for the idea!

2020.1 ALPHA RELEASE (09/12/2019)

* DecSoft is very proud to present the new generation of our DecSoft App Builder IDE, that want to be your favorite IDE to create modern apps. This is the first public alpha release that our customers can try. Please, stay tuned for the publication of the first public stable release of the product. We hope that you find the product useful for your projects. Thanks very much for choosing DecSoft App Builder. Enjoy it!

Download and contact www.decsoftutils.com mailto:info@decsoftutils.com