Changelog

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

[8.1.1] - 09-May-2025

Fixed

- [Regression since v8.1.0] Sound annotations do not play when double-clicked. (DOC-6897)

[8.1.0] - 10-Apr-2025

Added

- [Editor] Table data extractor tool (requires server-side SupportApi): finds and extracts tabular data from a region selected by the user. The data can be copied to clipboard or saved to a file in different formats including CSV, TSV, JSON, XLSX, XML or HTML. (DOC-5185)
- [Editor] [Wasm] Added support for redact, export to PNG, converting annotations to content and PDF linearize (fast Web view) to client-side (Wasm) SupportApi. (DOC-6388)
- Added support for the PDF sound action. (DOC-6738)

Changed

- [Editor] Improved the 'Text' property editor (always show the value, allows for easier editing). (DOC-6740)

Fixed

- Text selection is not visible in Safari Browser on IOS 18. (DOC-6822)

[8.0.6] - 2-Apr-2025

Added

- Option to send PDFs containing pages with different orientations to the browser print preview as a single job. (DOC-6854)

```
// Enable single print job mode to force all pages to be printed together:
var viewer = new DsPdfViewer("#root", {
  printSingleJobMode: true
});
```

Fixed

- [Wasm] Position of annotations changes when saving the document using Wasm support API. (DOC-6848)

[8.0.5] - 20-Mar-2025

Added

- Added support for cancelling the onBeforeOpen event. (DOC-6755)

Fixed

- [Regression since v8.0.4] Loading PDFs is very slow in FireFox browser. (DOC-6814)
- Typings of modules i18next and moment are missing in DsPdfViewer installation. (DOC-6778)
- The viewer zoom factor changes when the layout mode is changed. (DOC-6554)

- 'TypeError: Cannot read properties of null' can occur in specific scenarios. (DOC-6846)

[8.0.4] - 27-Feb-2025

Changed

- [Reply Tool] Enabled adding replies and statuses to locked annotations. (DOC-6773)

Fixed

- [Reply Tool] Fixed a permission check issue when deleting a reply using the Delete key. (DOC-6773)
- TypeError when clicking on the page with ReplyTool enabled. (DOC-6753)
- DsPdfViewer container element does not scroll if there is no PDF in the viewer. (DOC-6766)
- [API Docs] Some code examples formatting is incorrect. (DOC-6768, DOC-6769, DOC-6772)
- [iOS 17] Specific PDF fails to load on iOS 17.5 (loads fine on iOS 18.0 and above). (DOC-6724)

[8.0.3] - 09-Feb-2025

Added

- [Reply Tool] Automatic focusing on the clicked annotation in the replies list. Now, when you click on an annotation, the corresponding entry in the replies list will be focused and brought into view.
- [Reply Tool] Added new Reply Tool event bus events. (DOC-6732)
  - replytool-before-delete: triggered before an annotation is deleted using the Reply Tool. This event is cancelable, allowing prevention of deletion.
  - replytool-before-scrollintoview: triggered before the Reply Tool scrolls to a reply item's HTML element. This event is cancelable, enabling control over the scrolling behavior.
  - replytool-text-label-click: triggered when the user clicks a text label, before executing the associated Reply Tool actions (e.g., displaying the inline text editor for editable annotations). This event is cancelable, allowing customization or prevention of the default behavior.
  - replytool-item-click: triggered when the user clicks a list item in the Reply Tool, either a parent note item or a reply item. The event can be canceled by setting the cancel property of the event arguments object to true.
  - replytool-item-keydown: triggered when the user presses a key while a list item in the Reply Tool is focused (either a parent note item or a reply item). The event can be canceled by setting the cancel property of the event arguments object to true, preventing the default Reply Tool behavior.
  - replytool-placeholder-activate: triggered when the user activates the reply placeholder by clicking on it or pressing any key while it is focused. The event can be canceled by setting the cancel property of the event arguments object to true, preventing the default Reply Tool behavior (e.g., opening the reply editor).
  - replytool-author-label-click: triggered when the user clicks the author label, before executing the associated Reply Tool actions (e.g., opening the inline text editor for editable annotations). The event can be canceled to prevent the default behavior.
  - replytool-item-status-change: triggered when the user changes the status of a reply annotation. It can be canceled by setting args.cancel to true. The event also allows overriding the target annotation using args.annotation or modifying the status via args.status. Possible statuses: 'None', 'Accepted', 'Cancelled', 'Completed', 'Rejected'.
  - replytool-post-reply: triggered when the user adds a reply annotation via the "Post Reply" button or by pressing Ctrl+Enter after editing the reply text. The event can be canceled by setting args.cancel to true and allows modifying the reply annotation using args.annotation.

```
// Example:
viewer.eventBus.on("replytool-before-delete", function(args) {
    // Check if the entire annotation is being deleted (not just a comment)
    if (!args.removeCommentOnly) {
        // Confirm deletion with the user
```

```
        if (!confirm("Are you sure you want to delete the annotation with ID " + args.annotation.id + "?")) {
            // Cancel the deletion if the user declines
            args.cancel = true;
        }
    }
});
```

- Added annotationFilter option for Reply Tool. Allows customizing which annotations are displayed in the Reply Tool by providing a filter function.

```
// Example: show only text annotations in the Reply Tool
var viewer = new DsPdfViewer("#root", {
    replyTool: {
        annotationFilter: (annotation) => annotation.annotationType === 1 // AnnotationType.TEXT
    }
});
```

Changed

- JPN localization updated. (DOC-6727)
- [Reply Tool] Improved behavior for annotations without comments. (DOC-6718)
- The "Add a comment..." placeholder is now hidden unless the annotation is selected or focused.

Fixed

- Incorrect document display at high zoom values. (DOC-6728)
- [Reply Tool] Incorrect position of timestamp for the reply in the Reply Tool panel. (DOC-6716)
- [Reply Tool] The comment popup behaves incorrectly if the Reply Tool and other toolbars are used together. (DOC-6729)
- [Reply Tool] Issues with styling in Dark and Dark-Yellow Themes. (DOC-6737)
- [Reply Tool] Inconsistent behavior when deleting comments for annotations other than sticky notes. (DOC-6735)
- [Layers] Incorrect display of PDF layers in some cases. (DOC-6733)
- [Reply Tool] Issue with Reply Tool styling in Dark & Dark-Yellow Themes. (DOC-6737)
- [Reply Tool] The comment popup gets broken if reply tool and other toolbars are used together. (DOC-6729)

[8.0.1] - 25-Dec-2024

Added

- Added text selection capability to FreeText annotations. (DOC-6659)

Fixed

- Proximity search like [“My favorite” ONear(1) "is" ONear(5)"."] does not work as expected. (DOC-6579)
- Open search panel stops working when another document is picked from the document list. (DOC-6666)
- The same text can be replaced several times. (DOC-6674)

[8.0.0] - 03-Dec-2024

Added

- Added text replace feature: use Ctrl+H to open the search bar in replace mode, or click the expand button in the search bar to toggle between search and replace modes. (DOC-5200)
- Added display of timestamps in user comments. (DOC-6233)
- [Reply Tool] Added useRelativeDates and dateFormat options. (DOC-6586)

```
// Example 1: Disable relative dates and use absolute dates instead.
var viewer = new DsPdfViewer("#root", { replyTool: { useRelativeDates: false } });
```

```
// Example 2: Use a custom format for absolute dates.
var viewer = new DsPdfViewer("#root", { replyTool: { useRelativeDates: false, dateFormat: "yyyy.mm.dd
HH:MM" } });
```

- Added hidePopupTimestamp option. (DOC-6587)

Changed

- Enhanced proximity search with support for NEAR and ONEAR operators, added the ability to quote phrases for "full phrase" proximity search. (DOC-6390)

Fixed

- [CommentTimeStamp] Incorrect handling of future dates in relative date formatting. (DOC-6588)

[7.2.4] - 23-Oct-2024

Added

- Enabled two-finger scrolling while drawing ink annotations. (DOC-6537)

Fixed

- Save as SVG dialog message is not JPN localized. (DOC-6557)
- Using Ctrl+R to refresh the page is always handled by DsPdfViewer. (DOC-6548)

[7.2.3] - 02-Oct-2024

Changed

- JPN localization updated. (DOC-6514)

[7.2.2] - 19-Sep-2024

Changed

- The updateAnnotation and addAnnotation methods will now throw an error if the provided annotation object is improperly formed and cannot be used. (DOC-6506)
- [Documentation] The gcProps property has been moved from the base AnnotationBase class to the more specific WidgetAnnotation class. (DOC-6506)

Fixed

- The behavior is inconsistent when double-clicking a redact annotation. (DOC-6503)
- [iOS] Cannot scroll by swiping while entering ink annotations. (DOC-6501)

[7.2.1] - 28-Aug-2024

Added

- Added the ability to create field widgets with the same name, enabling such widgets to update their values simultaneously. (DOC-6461)
- Added new API method repaintTextLayer for efficiently repainting text selection and highlighting elements on visible pages.
- Added highlightTextSegment method to enable highlighting specific text segments on a page with customizable options for color and style.
- Added clearHighlightedSegments method to remove all custom text highlights from a specified page.
- Added `newDocumentFileName` option to specify the default file name when creating new documents in the viewer. This is distinct from `friendlyFileName`, which names existing documents.

Fixed

- [Wasm] Save API method did not recognize spaces as separators (commas worked fine). (DOC-6425)
- Unexpected content appears in a form field. (DOC-6420)
- Input widgets associated with the same field behave incorrectly. (DOC-6412, DOC-6415)
- Part of a combo text box's content is lost. (DOC-6356)
- [Form Editor] Cannot create multiple input fields with same name. (DOC-6416)
- In two pages side by side view, PageWidth zoom mode scales to single page width. (DOC-6476)
- Data using time format is not rendered correctly. (DOC-6482)
- Markup shows at incorrect locations in some browsers. (DOC-6474)

[7.2.0] - 07-Aug-2024

Added

- Added Wasm SupportApi option, it enables PDF editing features on the client without connecting to SupportApi/ DsPdf on a server.

```
// Example: use client-side Wasm SupportApi:
function loadPdfViewer(selector) {
  var viewer = new DsPdfViewer(selector, {
    supportApi: {
        implementation: new WasmSupportApi()
    }
  });
  viewer.addDefaultPanels();
  viewer.addAnnotationEditorPanel();
  viewer.addFormEditorPanel();
}
```

- Added registerFont method: registers a new @font-face style with the given font name and URL. Supported font formats include "ttf", "otf", "woff" and "woff2" ("woff2" is not supported by the Wasm SupportApi).

```
// Example 1: Registering a font with name and URL
viewer.registerFont('CustomFont', 'https://example.com/fonts/customfont.woff');
// Example 2: Registering a font with name, URL, and format
viewer.registerFont('AnotherFont', 'https://example.com/fonts/anotherfont.ttf', 'ttf');
```

- Added registerFallbackFont method: registers a fallback font that SupportApi will use when searching for fallback fonts. Supported font formats include "ttf", "otf", "woff", and "ttc". Font collections in "ttc" format are also supported.

```
// Example 1: Registering a fallback font with URL:
viewer.registerFallbackFont('https://example.com/fonts/fallbackfont.ttf');
// Example 2: Registering a fallback font with name and URL:
viewer.registerFallbackFont('SampleFont', 'https://example.com/fonts/SampleFont.woff');
```

- Added support for the standard "Zapf Dingbats" font, which can now be selected for text fields and free text annotations.
- Added a "Save As" button with the toolbar button key "save-as". Use the "Save As" button to export the PDF document to various formats (PDF, PNG, SVG). The old "Save" and "Save as images" buttons have been removed from the default toolbar layout.
- Added the ability to export a PDF document to SVG images.
- Added extended support for pageLayout mode based on initial viewer preferences specified in a PDF document. The following pageLayout values specified in the PDF correspond to pageDisplay values in the viewer:
  - TwoColumnLeft: sets pageDisplay to TwoPageScrolling.

  - TwoColumnRight: sets pageDisplay to TwoPageScrolling.
  - TwoPageLeft: sets pageDisplay to TwoPage.
  - TwoPageRight: sets pageDisplay to TwoPage.
  - OneColumn: sets pageDisplay to SinglePageScrolling.
  - SinglePage: sets pageDisplay to SinglePage.
- Added support for an ECMAScript action that can be performed before the field is formatted to display its value. (DOC-5643)
- Added support for two pages side by side view. (DOC-6063)

Changed

- The standard "Symbol" font now saves correctly and displays properly in most PDF viewers.
- [Toolbar] The "View Mode" button has been replaced with the "Page Display" dropdown menu. (DOC-6063)
- [Text Tools] Added the ability to create text markup annotations directly from selected text when text is selected.

Fixed

- [PDF Organizer] Page ranges were not created in some scenarios. (DOC-6345)

[7.1.4] - 23-Jul-2024

Added

- Added progress bar for long-running "Save as Images" operations, with the ability to cancel the current operations. (DOC-6301)
- JPN localization. (DOC-6297)

Fixed

- [Regression since v7.1.3] Saved PDF file name is different from the argument passed to the viewer.save() API. (DOC-6374)
- [macOS] Comment cannot be deleted with Delete key on macOS. (DOC-6152)
- Incorrect content of PDF while previewing on Ionic Modal. (DOC-6209)

[7.1.3] - 25-Jun-2024

Fixed

- Thumbnails are not shown when exiting full screen mode with 'NonFullScreenPageMode' set to 'UseThumbs'. (DOC-6293)
- [PDF.js vulnerability] Malicious JavaScript can run in PDF viewer's WebWorker via crafted PDF file. (DOC-6248)

Changed

- The required version of SupportApi has been changed to v7.1.3.

[7.1.2] - 27-May-2024

Fixed

- Cannot save a PDF if its name contains certain non-Latin characters. (DOC-6222)
- Localization doesn't work when the specified language is not all in lowercase. (DOC-6230)
- Full-screen mode incorrectly enabled for some PDFs without full-screen mode preference. (DOC-6243)

[7.1.1] - 23-Apr-2024

Fixed

- Cannot edit password-protected PDF when copying content is disabled. (DOC-6116)
- Comment added in the comment panel is not wrapped. (DOC-6134)
- Incorrect resize bar position. (DOC-6188)
- Auto-sized font appears larger than the required font size in some PDFs (additional fixes). (DOC-6029)
- [Collaboration] Only one page of a newly created PDF is shared. (DOC-4143)

[7.1.0] - 04-Apr-2024

Fixed

- Cannot submit comment by pressing the Enter kay on the 'Done' button. (DOC-6109)
- Issues with loading large files. (DOC-6080, DOC-6082)
- Edit mode is not deactivated when switching the viewer layout from annotation editor while the second toolbar is open. (DOC-6074)
- Searching for a text also highlights the space before the found text. (DOC-6142)

Changed

- Enhanced the behavior and appearance of the context menu, added the ability to add a note to selected text. (DOC-6100)
- Removed text markup menu that automatically popped up on text selection. The main context menu includes that menu as a submenu.
- Added the `showContextMenuOnSelection` option controlling the context menu behavior when text is selected.
  The option has the following values:
  - "Auto": Automatically determines whether to show the context menu based on the device type.
  - "On": Always shows the context menu when text is selected.
  - "Off": Never shows the context menu when text is selected.
    The default value is "Auto". On systems with mice, "Auto" behaves like "Off". On small devices (phones, tablets) without mice, "Auto" behaves like "On".
- [Reply Tool] Updated context menu texts. (DOC-6014)

Added

- Added support for rich media annotations (play embedded audio/video, modify rich media annotations). (DOC-5979)
- Reply Tool Improvements:
  - Added the ability to delete a comment item using the Delete key. Enabled navigation through comment items using the TAB and Arrow keys.
  - Added the ability to resize the right sidebar element.
  - The Reply Tool now activates automatically when a markup annotation or text comment is added via the context menu, focusing on the new comment in the list.

```
// Use autoExpandOnCommentAdd option to revert to previous behavior:
var viewer = new DsPdfViewer("#root", { replyTool: { autoExpandOnCommentAdd: false } });
```

  - The color for ReplyTool icons has been removed.

```
// Use useColoredIcons option to revert to previous behavior:
var viewer = new DsPdfViewer("#root", { replyTool: { useColoredIcons: true } });
```

- Split ISupportApi into base and multi-user parts, added the ability to specify a custom implementation for SupportApi that conforms to the ISupportApiBase interface. (DOC-5440)

```
// Example
var viewer = new DsPdfViewer("#root", {
    supportApi: {
        implementation: new CustomSupportApi()
    }
});
```

[7.0.4] - 20-Mar-2024

Fixed

- Incorrect appearance of a rotated stamp annotation. (DOC-6059)
- After calling dispose() method, viewer reserved keys still do not work correctly. (DOC-6060)
- Ellipsis is displayed in the floating search bar's settings when the system language is not English and the language option is not specified. (DOC-6067)

[7.0.3] - 19-Feb-2024

Added

- Added new option 'printResolution' (DOC-5995).
  The 'printResolution' option allows you to specify the resolution (in dots per inch - DPI) for printing purposes. When rendering and preparing a PDF documents for printing, this option determines the quality and detail of the generated images used in the printing process. A higher DPI value generally results in crisper and more detailed printed output but may also lead to larger memory usage and slower preview performance. The default value is 150 DPI.

```
  // Example: set the print resolution to 300 DPI:
  var viewer = new DsPdfViewer("#root", {
    printResolution: 300
});
```

Fixed

- In single page view mode, ItemArea for search results is NaN on Page 2. (DOC-6017)
- The search settings dialog is not closed after closing the floating search bar. (DOC-6021)
- Auto-sized font appears larger than the required font size in some PDFs. (DOC-6029)
- Unable to input "c" and "x" characters in CombTextField. (DOC-6035)

[7.0.2] - 25-Jan-2024

Fixed

- Floating text search window display issues in some locales. (DOC-5996)
- Incorrect display of some fonts. (DOC-5983)
- Searched text is not visible in the viewport. (DOC-5965)
- [SupportApi] Resources are not released when the viewer is disposed. (DOC-5733)
- Concurrency issue in the addAnnotation() method. (DOC-6010)
- [Search API] SearchResult.ItemArea is empty. (DOC-6000)

[7.0.1] - 04-Jan-2024

Added

- Added standardFontDataUrl option: the URL where the standard font files are located. Include the trailing slash. (DOC-5959)

```
// Example with relative URL:
var viewer = new DsPdfViewer("#root", { standardFontDataUrl: "resources/standard_fonts/" });
// Example with absolute URL:
var viewer = new DsPdfViewer("#root", { standardFontDataUrl: "http://localhost:8080/resources/
standard_fonts/" });
```

- Added Chinese localization. (DOC-5856)

```
// Example:
var viewer = new DsPdfViewer("#root", { language: "cn" });
```

Changed

- JPN localization updated. (DOC-5948)
- License messages updated. (DOC-5951, DOC-5953)

Fixed

- ComboBox: the first available option value displayed when field is empty. (DOC-5954)
- Console warnings about missing standard font files. (DOC-5969)
- Extra pages are added to the PDF in print preview when the HTML page contains elements with static positioning. (DOC-5895)
- Some incorrectly formed annotations are not shown. (DOC-5986)

[7.0.0] - 07-Dec-2023

Added

- Added DsPdfViewer class and "@mescius/dspdfviewer" package. Functionally DsPdfViewer is identical to GcPdfViewer. (DOC-5735)
- Added floating text search bar. (DOC-5406)
- Added localization resources and a localization example to the pdf viewer build.
- Added new option useFloatingSearchBar - enable a floating search bar instead of the sidebar search panel. Default value is true.

```
// To revert to using sidebar search panel instead of the floating search bar:
var viewer = new DsPdfViewer("#root", { useFloatingSearchBar: false });
```

Changed

- Some tooltips updated.

[4.2.5] - 22-Nov-2023

Fixed

- Cannot resize annotations and form fields in some PDFs. (DOC-5396)
- [Ionic Framework] Error "PDF Viewer Factory is not ready." occurs for Ionic Modal. (DOC-5826)
- [Search Panel] Text highlighting disappears when clicking on a document. (DOC-5838)
- [Editor] Problem when moving a stamp annotation to another page using the context menu. (DOC-5844)

Changed

- [Editor] Improved the floating bar containing the caret/hand options (added close button, reduced size).
- Enhanced the default HTML template index.html (added a description for SupportApi settings).

Added

- Localizable string resources and example for localization are now included in the package.

[4.2.4] - 27-Oct-2023

Fixed

- When a stamp annotation is moved to another page, the resulting position is incorrect. (DOC-5728)
- The annotation position is changed on saving changes after locking the annotations. (DOC-5732)
- [demos] Viewer API is not visible correctly with Window Dark Mode. (DOC-5721)

[4.2.3] - 28-Sep-2023

Fixed

- Clicking on a link in a PDF opens a blank page. (DOC-5709)
- [Windows Touch] Zooming by hand does not work. (DOC-5633)

[4.2.2] - 13-Sep-2023

Changed

- Localization resources updated. (DOC-5651)

Fixed

- [Regression] "Cannot find type definition" error when compiling a typescript enabled app. (DOC-5688)
- [Regression] Issues with displaying info messages from the viewer. (DOC-5690)
- Comments added by Adobe Acrobat Reader are not shown correctly when the renderInteractiveForms option is false. (DOC-5693)
- Annotations with AnnotationFlags.NoView are displayed. (DOC-5697)

[4.2.1] - 17-Aug-2023

Fixed

- [Editor] After saving a free text annotation with a custom font, the font in the saved PDF is incorrect. (DOC-5642)

Changed

- Localization resources updated. (DOC-5651)

Added

- [Editor] Ability to use custom fonts in saved PDFs, see CHANGELOG.md shipped with GrapeCity.Documents.Pdf.ViewerSupportApi for details. (DOC-5652)
- [Editor] Auto-adjusting font size of text boxes: if the font size is zero, it will be automatically adjusted to fit the content of the text box. (DOC-5558)

[4.2.0] - 01-Aug-2023

Fixed

- Content of some password-protected PDF documents is not displayed. (DOC-5623)
- The name of a document list item changed from lower to upper case. (DOC-5602)
- [iOS Desktop] PDF text cannot be selected on zooming the document. (DOC-5432)
- Miscellaneous UI improvements. (ARF-4244, DOC-5608, DOC-5638)

Added

- Added the ability to use an async function as a "beforeFormSubmit" handler. (DOC-5589)

```
// Example:
viewer.options.handlers  = {
  beforeFormSubmit: async function(args) {
    // Wait 5 seconds before submit:
    await new Promise(resolve => setTimeout(resolve, 5000));
    args.cancel = true;
    alert("5 seconds passed. Submit canceled.");
  }
};
```

- [Document List Panel] Added the ability to specify html content for document list items. (DOC-5367)
- [Collaboration] Added new methods: getSharedDocuments, openSharedDocumentByName, openSharedDocumentByIndex. (DOC-5457)

```
// Example 1: get the list of shared documents available to the current user:
var sharedDocuments = await viewer.getSharedDocuments();
// Example 2: open the first available shared document named "example.pdf":
viewer.openSharedDocumentByName("example.pdf");
// Example 3: open the second shared document in the list:
viewer.openSharedDocumentByIndex(1);
```

[4.1.5] - 14-Jul-2023

Fixed

- Image cache not functioning correctly when navigating from the first page. (DOC-5564)
- A signature with an appearance stream is not visible. (DOC-5547, DOC-5578)
- [iOS] [iOS Desktop mode] [Android] Pinch zoom in on a link redirects the page. (DOC-5560)
- [Editor] Free text annotation's text changes after repaint. (DOC-5573)
- Multiline text cannot be entered in free text & sticky note annotations. (DOC-5468)
  (Note that note that Adobe Acrobat does not support line breaks for plain text in text annotations. If line breaks are detected during editing, we automatically set the rich text flag for the text annotation.)
- When several lines of text are selected to create a markup annotation, a separate annotation is created for each line. (DOC-5503)
- Incorrect display on the second page when viewing a specific PDF. (DOC-5496)
- Incorrect handling of rich text when text markup is entered in a text annotation. (DOC-5551)
- Incorrect field names on calling the viewer.submitForm() method. (DOC-5553)
- [Editor] Content missing when opening a PDF on an IOS devices. (DOC-5407)
- [iOS] Incorrect context menu position when context menu is triggered by keyboard. (DOC-5555)
- [iOS] [Android] Pinch zoom in on a link is redirecting the page. (DOC-5560)

Changed

- Improved the behavior of free text annotations. (DOC-5368)

[4.1.4] - 20-Jun-2023

Fixed

- Improved handling of JavaScript code associated with widget events. (DOC-5475)
- [Windows Touchpad] In PDF Organizer, cannot dragged pages using the touchpad. (DOC-5465)

[4.1.3] - 31-May-2023

Fixed

- A typing error is thrown when adding stamps in code in an Angular app. (DOC-5433)

[4.1.2] - 08-May-2023

Added

- [Form editor] Added tooltip support for all fields, added "Field tooltip" property editor. (DOC-5384)
- Added addStamp method. (DOC-5376)

```
// Example: add graphical signature to the PDF using external image:
function addStampFromUrl(imageUrl, viewer) {
 fetch(imageUrl)
 .then(response => response.blob())
 .then(blob => blob.arrayBuffer())
 .then(arrayBuffer => {
       const fileId = new Date().getTime() + ".png";
       const fileName = fileId;
       const pageIndex = 0;
       const imageData = new Uint8Array(arrayBuffer);
       const rect = [0, 0, 200, 200];
       viewer.storage.setItem(fileId, imageData);
       viewer.addStamp(
                   imageData,
                   {
                       fileId,
                       fileName,
                       pageIndex,
                       rect,
                       select: false,
                       subject: "",
                       rotate: 0,
                       convertToContent: false
                   });
 });
}
addStampFromUrl("http://example.com/image.png", viewer);
```

- Added jsExecutionConfig option: optional execution configuration for JavaScript actions.

```
// Example 1: adjust jsCode before execution:
viewer.options.jsExecutionConfig = {
  before: function(args) {
      args.jsCode = args.jsCode.replace("app.alert", "app.showMessage");
  }
}
// Example 2: cancel JS execution:
viewer.options.jsExecutionConfig = {
  before: function(args) {
      args.cancel = true;
  }
}
// Example 3: execute long operation before JS execution and repaint visible pages after JS execution:
viewer.options.jsExecutionConfig = {
  before: function() {
      // Returned promise will be awaited before further JS execution.
      return new Promise(function(resolve) {
                setTimeout(resolve, 1000);
      });
  },
  after: function() {
      viewer.repaint();
  }
}
```

Changed

- [Editor] Updated property labels for field widgets: property "Name" renamed to "Field name", property "Value" renamed to "Field value", property "Export Value" renamed to "Choice value".

Fixed

- The second toolbar position is not updated in some cases. (DOC-5377)
- [iOS] When a signature is drawn by Apple Pencil, a lens is sometimes displayed. (DOC-5370)
- [iOS] Locking all fields does not lock some properties. (DOC-5299)
- [iOS] Several issues when using the PDF Organizer. (DOC-5344, DOC-5345)
- Stamp annotations in some PDFs incorrectly appear rotated. (DOC-5383)
- Cannot copy a layer name from the Layers panel. (DOC-5385)
- Cannot load a stamp generated from an image. (DOC-5379)
- JavaScript actions' sequence is incorrect for focus/blur events. (DOC-5389)
- Localization issues. (DOC-5359)
- [Editor] Incorrect popup annotation orientation is some cases. (DOC-5386)

[4.1.1] - 17-Apr-2023

Fixed

- [Regression] [Editor] 'Support API server version 6.1.0 is out of date' warning incorrectly shows when the viewer is connected to SupportApi v6.1.0. (DOC-5374)

[4.1.0] - 03-Apr-2023

Added

- PDF Organizer: new feature that allows users to rearrange, duplicate or remove pages of a PDF, or merge PDFs. (DOC-3914, DOC-3915, DOC-3916)
  To open the PDF Organizer dialog, in the top toolbar click 'Page Tools', and in the secondary toolbar click 'PDF Organizer'.
- Respect PDF's initial view settings (hide toolbars or menus, open with specific page layout, etc). (DOC-4667)
- Added ignoreInitialView option: set this option to true to ignore initial view settings specified in PDFs.

```
// Example:
viewer.options.ignoreInitialView = true;
```

- Added support for zoom-dependent optional content (layers). (DOC-3970, DOC-4108)
- Added the ability to specify a custom progress message during save action.

```
// Example: use your own progress title and message:
viewer.save("sample.pdf", { pages: "[angle:90]0,1-5", progressTitle: "Rotating", progressMessage: "Rotating
first page..." }, reload: true);
```

- Added new method resolvePageIndex(): resolves the page index using PDF page reference. (DOC-5214)

```
// Example:
const openAction = (await viewer.viewerPreferences).openAction;
if(openAction && openAction.dest) {
  const pageRef = openAction.dest[0];
  const targetPageIndex = await viewer.resolvePageIndex(pageRef);
}
```

- Added holdToPan action: press and hold the spacebar to temporarily enable the pan tool.

```
// Example 1: disable holdToPan when space is pressed:
viewer.options.shortcuts["32"] = () => {};
// Example 2: bind P to holdToPan action, keep Ctrl+P bound to print action:
viewer.options.shortcuts["P"] = [{ ctrl: true, tool: "print" }, { tool: "holdToPan" }];
```

Changed

- 'Layers' and 'StructureTree' panels added to the default sidebar layout, 'Articles' panel removed. The default set of sidebar panels is now as follows: 'Thumbnails', 'Search', 'Outline', 'Layers', 'StructureTree', 'Attachments'.

```
  // Example: restore the old default layout:
  var viewer = new GcPdfViewer("#root");
  viewer.addArticlesPanel();
  viewer.addThumbnailsPanel();
  viewer.addSearchPanel();
  viewer.addOutlinePanel();
  viewer.addAttachmentsPanel();
```

- [Editor] The printable flag is now set to true for all new annotations.
- Returning model type for the viewer.viewerPreferences property changed: all property names are now camelCased, new properties added: openAction, pageMode, pageLayout.
- Reduced the heights of dialogs' title bars. (DOC-5313)
- [Editor] Text annotation (sticky note) properties display improved. (DOC-3241)

[4.0.7] - 07-Apr-2023

Changed

- hideAnnotationPopups option improved: the option now allows specifying annotation types that won't show popups. (DOC-5283)
  Possible values are: ['Text', 'Link', 'Line', 'Square', 'Circle', 'Polygon', 'PolyLine', 'Ink', 'Popup', 'FileAttachment', 'Sound', 'Redact', 'Stamp'] or true or 'All' (true and 'All' have the same behavior).

```
// Example 1: hide popups for all annotations:
var viewer = new GcPdfViewer("#root", { hideAnnotationPopups: true });
// Example 2: hide popups for Redact, Circle and Square annotations:
var viewer = new GcPdfViewer("#root", { hideAnnotationPopups: ["Redact", "Circle", "Square"] });
```

- The save() method now allows specifying a range(s) of pages to save, changing page order or duplicating pages. (DOC-3916)

```
// Example 1: save specified pages only:
await viewer.save("test.pdf", { pages: "0, 3, 5-7" });
// Example 2: duplicate and save the first page:
await viewer.save("test.pdf", { pages: "0, 0" });
// Example 3: split PDF into two documents:
await viewer.save("test_part1.pdf", { pages: "0-3" });
await viewer.save("test_part2.pdf", { pages: "4-8" });
// Example 4: change page order:
await viewer.save("test_changed_order.pdf", { pages: "3, 2, 1, 0" });
```

- The saveAsImages() method now allows specifying the zoom factor.

```
// Example:
viewer.saveAsImages("sample.pdf", { zoomFactor: 1.5 });
```

Fixed

- [Editor] In a PDF with different page sizes the 'move to next page' option may not work correctly. (DOC-5217)

- [Editor] A locked stamp annotation can still be edited in GcPdfViewer. (DOC-5216)
- [Editor] Cannot undo changes if the editor layout is not activated. (DOC-5315)
- [Editor] A popup annotation may move unexpectedly. (DOC-5318)
- Cannot navigate between pages using PageUp/PageDown keys in single page view. (DOC-5335)
- The list of layers is not shown correctly in some cases. (DOC-5338)

[4.0.6] - 23-Mar-2023

Fixed

- [Editor] Move to next page does not work correctly with different page sizes. (DOC-5217)
- [Editor] The zoom value changes after finishing a drawing with the OK button. (DOC-5184)
- [Regression] Form fields that should be hidden by a checkbox are still visible after unchecking the checkbox. (DOC-5242)

[4.0.5] - 27-Feb-2023

Fixed

- In some cases the signer's name is not shown correctly in the viewer. (DOC-5145)
- signatureValue returns incorrect JP text for certain PDFs. (DOC-5136)
- [Demos] Downloaded sample cannot be run correctly. (DOC-5149)
- [Editor] Annotations cannot be placed correctly in specific PDFs. (DOC-5125)
- [Editor] Ink annotations' position are incorrect when zooming in/out. (DOC-5139)
- [Editor] A checkbox is not checked when setting its fieldValue same as the export value. (DOC-5151)
- [Editor] Line coordinates are incorrect when adding a line annotation in a specific PDF. (DOC-5144)
- [Editor] A 'Drag and drop error' message shows when trying to input a property value in some cases. (DOC-5169)
- [Editor] In some scenarios, a checkbox that is checked by code does not show as checked. (DOC-5152)
- [Editor] A checkbox's shown value is not synchronized with its state. (DOC-5113)
- [Editor] For checkboxes with the same name, the checked state shows incorrectly by Acrobat in some cases. (DOC-5155)

[4.0.4] - 06-Feb-2023

Added

- [Demos] Added new sample "Prevent editing a signed PDF".

Fixed

- Filling and saving a signed form invalidates the signature. (DOC-5090)
- Localization of digital signature verification. (DOC-5075)

[4.0.3] - 20-Jan-2023

Added

- [Editor] Added property editors for line annotation coordinates.

Fixed

- [Editor] Modifying 'End Y‘ of a line coordinates causes the 'Start Y' to change. (DOC-5040)
- [Editor] In some cases the size of a checkbox becomes incorrect. (DOC-5042)
- Checkbox widgets with the same name did not work correctly. (DOC-4874)
- Ink highlight is not visible in the viewer. (DOC-5048)
- Cannot redefine Ctrl+P keyboard shortcut. (DOC-5021)

- UpdateAnnotation event is not triggered on switching radio buttons. (DOC-5020)
- When a Japanese text is assigned to a ButtonAppearance.Caption, the text is garbled in the generated PDF. (DOC-5028)
- Japanese texts are garbled in the article panel. (DOC-5038)
- PDF content can be printed using native browser menu even when Print is disabled by the disableFeatures option. (DOC-5035)
- disableFeatures.Print does not hide the Print context menu item. (DOC-5029)
- [Editor] Several issues with Ink, Polyline and Polygon annotations and callout lines. (DOC-5014)
- [Editor] Line annotation in the saved PDF is not shown in Acrobat. (DOC-5014)
- [iOS] Unable to toggle checkbox state on touch devices. (DOC-5010)

[4.0.1] - 27-Dec-2022

Added

- Added the ability to override addAnnotation/updateAnnotation method parameters:

```
// Example:
const pageIndex = 0;
// Using overridden methods without pageIndex parameter:
annotation.pageIndex = pageIndex;
await viewer.addAnnotation(annotation);
await viewer.updateAnnotation(annotation);
// Using methods with pageIndex parameter:
await viewer.addAnnotation(pageIndex, annotation);
await viewer.updateAnnotation(pageIndex, annotation);
```

- Added the ability to save PDFs using incremental update or linearized mode. (DOC-4913)

```
// Example: save document using IncrementalUpdate mode.
viewer.save("test.pdf", { saveMode: "IncrementalUpdate" });
```

```
// Example: save document using Linearized mode.
viewer.save("test.pdf", {saveMode: "Linearized"});
```

- Added the ability to load an updated PDF document into the viewer without downloading it to the local system.

```
// Example: Save document and load the saved document into the viewer:
await viewer.save("test.pdf", { reload: true });
```

- Added electronic signature API. (DOC-4951)

```
// Example: Save document with signature:
viewer.save("test.pdf", { sign: { signatureField: "field1", signerName: "John Doe" } } );
```

Example of server side signing:

```
// Example: electronically sign PDFs with a .PFX certificate:
public void Configuration(IAppBuilder app) {
  GcPdfViewerController.Settings.Sign += _OnSign;
  // ...
}
private void _OnSign(object sender, SignEventArgs e)
{
    var signatureProperties = e.SignatureProperties;
    X509Certificate2 certificate = new X509Certificate2(System.IO.File.ReadAllBytes("certificate.pfx"),
"password",
      X509KeyStorageFlags.MachineKeySet | X509KeyStorageFlags.PersistKeySet |
```

```
X509KeyStorageFlags.Exportable);
    signatureProperties.SignatureBuilder = new Pkcs7SignatureBuilder()
    {
       CertificateChain = new X509Certificate2[] { certificate },
       HashAlgorithm = Security.OID.HashAlgorithms.SHA512,
       Format = Pkcs7SignatureBuilder.SignatureFormat.adbe_pkcs7_detached
    };
}
```

- Added signature checking API. (DOC-4962)

```
 // Example: check whether the current document is signed and show information about the signature:
 var viewer = GcPdfViewer.findControl("#root");
 const signatureInfo = await viewer.getSignatureInfo();
 if(signatureInfo.signed) {
   const signatureValue = signatureInfo.signedByFields[0].signatureValue;
   const signerName = signatureValue.name;
   const location = signatureValue.location;
   const signDate = viewer.pdfStringToDate(signatureValue.modificationDate);
   alert("The document was signed using digital signature. Signed by: " + signerName + ", location: " +
location + ", sign date: " + signDate.toString());
 } else {
   alert("The document is not signed.");
 }
```

- [demos] Added new sample "Sign document using digital signature".

Changed

- Default page view margins changed (added left/right margins).
- Japanese localization resources updated.
- Removed border highlight for article thread beads.\

```
// The following code can be used to revert to the previous article thread beads style:
GcPdfViewer.findControl("#root").addViewAreaStyle(".gc-thread-bead { box-shadow: rgba(255, 232, 0, 0.27) 1px
1px 1px 1px;}");
```

Fixed

- Support multiple quadrilateral regions for text markup annotations. (DOC-5008)
- Incorrect position when a stamp is pasted with zoom different from 100%. (DOC-5006)
- Incorrect highlight when document contains highlight annotations with multiple areas. (DOC-5003)
- Setting to disable the undo/redo operation using keyboard Ctrl+z/y. (DOC-4932 / DOC-4901)
- Polyline changed after saving current document. (DOC-4942)
- Ink annotation's position changed after save then load if the border width was set. (DOC-4941)
- The size of resizing rectangle around sound annotation changed after invoke setAnnotationBounds(). (DOC-4937)
- Cannot invoke signature tool "afterAdd" event if the tool's location was not set to "BottomRight". (DOC-4948)
- Checkboxes with the same name are checked together. (DOC-4874)
- AutoScroll in Single Page View. (DOC-4959)
- Some annotations are not added correctly. (DOC-4943)
- On performing the Undo operation, only the image gets deleted from the stamp Annotation. (DOC-4686)
- Cannot add annotation to the second page of a specific file. (DOC-4935)
- Incorrect selection rectangle position when the page is zoomed to a specific value. (DOC-5001)
- The toolbar button overlaps zoom input. (DOC-5002)

[4.0.0] - 06-Dec-2022

Added

- [Editor] Added support (toolbar items and context menu) for text markup annotations (highlight, underline, strikeout and squiggly). (DOC-4471)

```
// To disable showing text markup context menu:
var viewer = new GcPdfViewer(selector, {
       textMarkupContextMenu: false
});
// To disable showing the markup color selector:
var viewer = new GcPdfViewer(selector, {
  textMarkupContextMenu: { colors: [] }
});
// To change text markup context menu colors to "Red" and "Black":
var viewer = new GcPdfViewer(selector, {
       textMarkupContextMenu: { colors: [{value: "#ff0000", displayName: "Red"}, {value: "#000000",
displayName: "Black"}] }
});
```

- [Editor] Added the ability to resize line annotations. (DOC-4705)
- [Editor] [Graphical Signature] Added the ability to position a signature and drag it between pages using the mouse. (DOC-4732, DOC-4740)
  If the signTool.location option is not specified, the signature tool will now allow the user to drag the signature to the desired location, possibly on a different page.
- Added API documentation for the disableFeatures option. (DOC-3298, DOC-4747)
- Added "Ear" icon for sound annotations.
- Added a "Save current document as images" button to the toolbar. (DOC-4774)

Changed

- Text, sound and file attachment icons updated to look like Acrobat Reader icons.
- The file and sound annotations are no longer resizable. The resulting annotation size now depends on the selected icon type. (DOC-2373)
- The layout of the "Text" quick editing tools has been updated to include buttons for the new text markup annotations.

Fixed

- Annotation borders appear different in GcPdfViewer and Acrobat. (DOC-4804)
- Line annotation is not visible in some cases. (DOC-4769)
- [Editor] Incorrect icon is shown when adding several text annotations. (DOC-4856)
- [Editor] When converting annotations to content, some annotation icons are incorrect. (DOC-2373)
- [Editor] Once a stamp annotation is rotated, the image cannot be resized. (DOC-4671)
- [Editor] Can not add a stamp annotation to a rotated document. (DOC-4776)
- [Editor] After rotating an annotation it cannot be resized. (DOC-4664)
- [Editor] Error "operation is not valid due to the current state of the object" occurs when attaching a stamp to a PDF. (DOC-4860)

[3.2.5] - 08-Dec-2022

Fixed

- Cannot override the default undo key Ctrl Z. (DOC-4901, DOC-4932)
- Exception is thrown when deleting a page in the form Editor. (DOC-4930)
- The open file dialog is not shown when adding sound or file attachment annotation. (DOC-4925)

- Annotations' properties change after saving the document. (DOC-4924, DOC-4942, DOC-4941)

[3.2.4] - 04-Nov-2022

Added

- [Editor] [Graphical Signature] Added the ability to position a signature and drag it between pages using the mouse. (DOC-4732, DOC-4740) If the signTool.location option is not specified, the signature tool will now allow the user to drag the signature to the desired location, possibly on a different page. (DOC-4732, DOC-4740)

```
// To keep the old behavior:
// Variant 1:
var viewer = new GcPdfViewer("#root", {
       signTool: {
              location: "BottomRight"
       }
});
// Variant 2:
viewer.options.signTool = { location: "BottomRight" };
```

- [Editor] Added the ability to drag stamps between pages. (DOC-4724, DOC-4733)
- [API request] Added undoState property: contains the current state of the undo store. (DOC-4687)
  Note that this property is read-only, do not programmatically change the elements of the collection. Use the Undo API to modify the undoState.
  Available Undo API properties: hasUndoChanges, hasRedoChanges, undoIndex, undoCount
  Available Undo API methods: undoChanges(), redoChanges()

```
// Example:
viewer.eventBus.on("undostorechanged", function() {
    console.log("Undo State changed", viewer.undoState);
});
// Or:
viewer.eventBus.on("documentchanged", function() {
    console.log("Document changed", viewer.undoState);
});
```

Changed:

- [Graphical Signature] The default Graphical Signature tool behavior has changed. (DOC-4732)
  When the signTool.location option is not specified, the Graphical Signature will allow the user to move it to the desired location with the mouse.

Fixed

- [Forms] Cannot reset all fields' values using a push button. (DOC-4685)
- [Editor] PDF is saved incorrectly if it contains intersecting annotations and redacts (this fix also requires GrapeCity.Documents.Pdf.ViewerSupportApi v5.2.0.804 or later). (DOC-4722, DOC-4720)
- [Editor] Cannot cancel adding a stamp annotation using the ESC key. (DOC-4731)

[4.0.0-alpha1] - 21-Oct-2022

Added

- Added support for new annotations: Highlight, Underline, Squiggly, StrikeOut.
- Added the ability to save the PDF's pages as images. (DOC-4577) Added the saveAsImages method - saves the pages of the current PDF document as PNG images, zips the result images, and downloads the result zip archive.

```
// Example:
viewer.saveAsImages('test.zip');
```

- Added new "Save as images" toolbar button, the button key is "save-images". (DOC-4577) The new button is available on "annotation editor" and "form editor" toolbar.
- Added undoState property, contains current state of the undo store. (DOC-4687)

[3.2.3] - 11-Oct-2022

Fixed

- [SupportApi for WebForms] An error occurs when trying to share a document using the WebForms version of the SupportApi service. (DOC-4668)
- [Windows Touch] The pan tool does not work correctly. (DOC-4644)
- [Editor] The stamp annotation gets added as attachment when the attachment is added to the PDF. (DOC-4649)
- [Editor] The sign annotation gets rotated for rotated page. (DOC-4663)
- [Editor] The line annotation is not visible when the line is drawn at 0 or 90 degree. (DOC-4659)
- [Editor] [Regression] "Add Signature" dialog layout is incorrect when dialogLocation is set to "Right". (DOC-4513)
- [Editor] All properties of the annotation should be reset if the cancel flag is set to true in the onBeforeUpdateAnnotation event. (DOC-4139)
- [Editor] The cursor does not change to "cross-hair" style when activating the editor tool from the toolbar for the first time. (DOC-2977)
- The alignment property does not work with fields in JS actions. (DOC-4636)
- [iOS] It is hard to select native checkbox on iOS. (DOC-4517)

Added

- Added the ability to change SVG icons used by the viewer using the customIcons option.

```
// Example:
var viewer = new GcPdfViewer("#root", {
  customIcons: {
    'open': '<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="24" height="24" viewBox="0 0 24
24"><path d="M9.516 14.016q1.875 0 3.188-1.313t1.313-3.188-1.313-3.188-3.188-1.313-3.188 1.313-1.313 3.188
1.313 3.188 3.188 1.313zM15.516 14.016l4.969 4.969-1.5 1.5-4.969-4.969v-0.797l-0.281-0.281q-1.781 1.547-4.219
1.547-2.719 0-4.617-1.875t-1.898-4.594 1.898-4.617 4.617-1.898 4.594 1.898 1.875 4.617q0 0.984-0.469
2.227t-1.078 1.992l0.281 0.281h0.797z"></path></svg>',
    'search': '<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="24" height="24" viewBox="0 0 24
24"><path d="M9.516 14.016q1.875 0 3.188-1.313t1.313-3.188-1.313-3.188-3.188-1.313-3.188 1.313-1.313 3.188
1.313 3.188 3.188 1.313zM15.516 14.016l4.969 4.969-1.5 1.5-4.969-4.969v-0.797l-0.281-0.281q-1.781 1.547-4.219
1.547-2.719 0-4.617-1.875t-1.898-4.594 1.898-4.617 4.617-1.898 4.594 1.898 1.875 4.617q0 0.984-0.469
2.227t-1.078 1.992l0.281 0.281h0.797z"></path></svg>'
  }
});
```

[3.2.2] - 16-Sep-2022

Fixed

- [Editor] The default dash style is not used when editing an annotation with a dashed border. (DOC-4630)
- Text in a certain PDF is not rendered correctly. (DOC-4614, DOC-4616, DOC-4609)
- Text in some PDF documents cannot be selected. (DOC-4615)
- Text becomes invisible when a PDF document contains invalid path operators. (DOC-4617)
- [Android] [iOS] Zooming using fingers gesture is not smooth. (DOC-4550)

Added

- Method getPageLocation: returns the position of the page view relative to the browser window. (DOC-4602)

- Added the ability to set the selected image for the Stamp button. (DOC-4601)

```
// Example:
var viewer = new GcPdfViewer("#root", {
       stamp: {
    stampCategories:  false,
    selectedImageUrl: "https://example.com/download?file=images/image.jpg"
  }
});
```

[3.2.1] - 22-Aug-2022

Fixed

- [Editor] toolbarLayout.stickyBehavior interferes with adding stamp annotations. (DOC-4549)
- [Editor] Incorrect stamp annotations behavior when the Hide Annotations/Form fields option is enabled. (DOC-4532)
- Thumbnail navigation issue when two or more mice are connected to the system. (DOC-4553)
- [Regression] Stamp annotations are rendered incorrectly in some cases. (DOC-4554)

Changed

- The sidebar's pin button tooltip now updates according to the state of the pin.
- Minor css update for zoom input (font size reduced by 1px).
- Japanese localization resources updated.

[3.2.0] - 05-Aug-2022

Fixed

- [Form Filler] Date fields do not work correctly in form filler dialog. (DOC-4497)
- [iOS] The print preview is empty. (DOC-4515)
- [Windows Touch] Reply tool button does not respond to touch. (DOC-4508)
- [Windows Touch] Thumbnail pane does not work correctly. (DOC-4516)
- [iOS Desktop mode] Custom input types display is incorrect. (DOC-4504)
- [iOS Desktop mode] Read-only fields still show the dropdown menu. (DOC-4505)
- [Editor] "Draw polygon annotation" button can be pressed after canceling a draw operation, but does not work. (DOC-4499)

Changed

- Save and download buttons' tooltips updated. (DOC-4511)

Added

- Added the ability to change current user name from the UI. (DOC-4498)
- Added the ability to change the minimum/maximum zoom factor using the zoomOptions option.

```
  // Example:
  var viewer = new GcImageViewer("#root", {
    zoomOptions: {
      minZoom: 0.05,
      maxZoom: 5,
      dropdownZoomFactorValues: [0.05, 0.1, 0.3, 0.5, 0.7, 1, 1.5, 2, 3, 5]
    }
  });
```

[3.1.6] - 14-Jul-2022

Fixed

- [Regression] Some annotations are not visible in the viewer. (DOC-4430)
- The validateForm() method skips List Box fields. (DOC-4377)
- The getRenderedAnnotationBounds() method returns incorrect result if the document is rotated. (DOC-4276)
- [Android] Zoom is slow and not smooth on Android devices. (DOC-4253)
- [Android] Comb-Text displays incorrectly. (DOC-4146)
- [Editor] The OK button does not work after drawing a horizontal or a vertical line with Ink annotation. (DOC-4278)
- [Editor] The hasChanges property is true after saving the modified document. (DOC-4348)
- [Print] PDF is not printed correctly when PDF have both portrait and landscape pages. (DOC-4418)
- Line annotations with 0 width show in Acrobat but are invisible in GcPdfViewer. (DOC-4428)

[3.2.0-alpha1] - 24-Jun-2022

Fixed

- The contents in zoom textbox should be automatically selected when user mouse clicks on the zoom textbox. (DOC-4303)
- [Editor] hasChanges property is true even after saving the modified document. (DOC-4348)
- The validateForm method skips listbox field. (DOC-4377)

[3.2.0-alpha0] - 16-May-2022

Changed

- PDF.js engine updated. (DOC-3870)
- Request to make Zoom textbox editable. (DOC-1205)
  - Press Enter to apply the new value after editing.
  - Press ESC to reset the edited value.
  - You can use the up/down arrow keys to increase/decrease the input value.

Fixed

- PDF rendering problem #14462 (https://github.com/mozilla/pdf.js/issues/14462)

[3.1.5] - 09-Jun-2022

Fixed

- [Editor] Modified value of a Combobox cannot be saved if it is custom (requires SupportApi v5.1.0.793 or later). (DOC-4280)
- [Editor] When the orientation of a CombTextField is changed, borders between combs are not displayed correctly. (DOC-4322)

Added

- [Structure tree panel] Structure tree UI enhancements. Highlight the clicked text content. (DOC-3918)

[3.1.4] - 18-May-2022

Changed

- Documentation files updated.

[3.1.3] - 29-Apr-2022

Fixed

- [Regression] A checkbox with an attached JavaScript automatically unchecks when checked. (DOC-4244)
- [Editor] Cannot draw horizontal or vertical lines using ink annotation tool. (DOC-4245)
- [Editor] Undo works incorrectly when undoing rotating a stamp. (DOC-4237)
- [Editor] Incorrect appearance of stamp annotations after rotation. (DOC-4208, DOC-4225, DOC-4227)
- In some scenarios moving the mouse resizes the left panel. (DOC-4238)

Changed

- Updated Japanese localization strings.

Added

- Added resize handles to panels. (DOC-3171)

[3.1.2] - 12-Apr-2022

Fixed

- [Editor] Annotations and fields are added in incorrect orientation when the document is rotated. (DOC-3260)
- [Android] Zooming using pinch does not work. (DOC-3424)
- [Collaboration] Sharing a multi-page PDF is not working correctly. (DOC-4143)
- [Collaboration] The file name is incorrect in the "Manage Access" dialog. (DOC-4144)
- [JavaScript actions] Viewer properties are undefined in JavaScript actions. (DOC-4154)
- In some cases the visibility state of layers is incorrect after saving a PDF. (DOC-4067)

Changed

- Do not show the version mismatch warning if the connected SupporApi has version 0.0.0.0 (was built from sources).

Added

- New method invalidate: ensures that all visual child elements of the viewer are properly updated for layout.
- New property requiredSupportApiVersion: gets the required version of SupportApi that is compatible with the current version of GcPdfViewer.
- New property supportApiVersion: gets the connected version of SupportApi, if available.
- New property gcPdfVersion: gets the version of GcPdf library used by the connected SupportApi, if available.
- [Editor] Ability to rotate stamp and free text annotations using rotation handles. Pressing the Shift key snaps the rotation angle to a multiple of 90 degrees. (DOC-4058)
- [XFA forms] Added support for print, submit, reset, JavaScript actions, links. (DOC-3869)
- [Editor] Ability to persist the visibility state of optional content groups (layers) when saving the PDF. (DOC-3652)
- [Editor] Ability to change a widget's content orientation using the orientation property. (DOC-3260)
- [Editor] Sticky behavior for toolbar buttons: a button remains pressed after the editing operation is complete. (DOC-3913)
  Added stickyBehavior setting to toolbarLayout: an array with button keys that will have sticky behavior. Note that only annotation and form editor toolbar buttons can be made sticky.
  The complete list of buttons that can be made sticky: 'edit-sign-tool', 'edit-text', 'edit-free-text', 'edit-ink', 'edit-square', 'edit-circle', 'edit-line', 'edit-polyline', 'edit-polygon', 'edit-stamp', 'edit-file-attachment', 'edit-sound', 'edit-link', 'edit-redact', 'edit-widget-tx-field', 'edit-widget-tx-password', 'edit-widget-tx-text-area', 'edit-widget-btn-checkbox', 'edit-widget-btn-radio', 'edit-widget-btn-push', 'edit-widget-ch-combo', 'edit-widget-ch-list-box', 'edit-widget-tx-comb', 'edit-widget-btn-submit', 'edit-widget-btn-reset', 'edit-erase-field'

```
// Example: make square, circle, line and redact buttons sticky:
viewer.toolbarLayout.stickyBehavior = ["edit-square", "edit-circle", "edit-line", "edit-redact"];
```

- Option fieldsAppearance: added the ability to specify button fields render type. (DOC-3537)

```
// Example 1: Use platform-native styling for push buttons.
var viewer = new GcPdfViewer("#root", { fieldsAppearance: { pushButton: "Web" } });
// Example 2: Use predefined appearance stream for push buttons:
var viewer = new GcPdfViewer("#root", { fieldsAppearance: { pushButton: "Predefined" } });
```

- Ability to programmatically hide the left sidebar. (DOC-3981)

```
// Hide sidebar:
viewer.toggleSidebar(false);
// Alternative variant:
viewer.leftSidebar.hide();
```

- Ability to programmatically hide or show the toolbar. (DOC-3981)

```
// Example:
viewer.toggleToolbar(false);
```

- Option hideAnnotationPopups: use this option to hide all annotation popups. (DOC-3981)

```
// Example:
var viewer = new GcPdfViewer("#root", { hideAnnotationPopups: true });
```

- Ability to close the currently loaded document. (DOC-3981)

```
// Example:
await viewer.close();
```

- New events: onBeforeAddAnnotation, onAfterAddAnnotation, onBeforeUpdateAnnotation, onAfterUpdateAnnotation, onBeforeRemoveAnnotation, onAfterRemoveAnnotation. (DOC-3981)

```
// Example:
viewer.onBeforeAddAnnotation.register(function(args) { console.log(args); });
```

Events BeforeAddAnnotation, BeforeUpdateAnnotation and BeforeRemoveAnnotation are cancelable.

```
// Examples of canceling events:
viewer.getEvent("BeforeAddAnnotation").register(function(args) { args.cancel = true; });
viewer.getEvent("BeforeUpdateAnnotation").register(function(args) {
  args.cancel = true;
  viewer.repaint();
});
```

- Ability to listen to and trigger custom events. (DOC-3981)

```
// Example: listen to CustomEvent:
viewer.getEvent("CustomEvent").register(function(args) {
  console.log(args);
});
// Example: trigger CustomEvent:
viewer.triggerEvent("CustomEvent", { arg1: 1, arg2: 2});
```

[3.0.22] - 06-Apr-2022

Fixed

- [Editor] Arrow keys move annotations incorrectly when the document is rotated. (DOC-4129)
- [Editor] Stamp annotation disappears from page after printing the document. (DOC-4127)

[3.0.21] - 22-Mar-2022

Fixed

- [Regression] [Form Editor] Unable to move form fields using the arrow keys. (DOC-4115)
- [Form Editor] Text cursor moves when an annotation is moved using the arrow keys. (DOC-4123)
- [Regression] Missing "token" parameter when requesting a stamp image using GET method. (DOC-4080)
- In some cases the visibility state of layers is incorrect after loading or saving a PDF. (DOC-4068)

[3.0.20] - 03-Mar-2022

Fixed

- When renderInteractiveForms is false, checkbox values are not displayed. (DOC-4022)
- Highlight on the search text disappears when the document is zoomed in or out. (DOC-4028)
- [Editor] Custom image stamps are gone when the viewer is closed and recreated. (DOC-4023)

[3.0.19] - 25-Jan-2022

Added

- Added the ability to specify authentication or other HTTP headers in the open() method. (DOC-3998)

```
// Example: specify basic access authentication and custom headers:
viewer.open("http://example.com//pdfs/GetPdf?file=HelloWorld.pdf", {
  headers: {
    "Authorization": "Basic " + btoa(unescape(encodeURIComponent("USERNAME:PASSWORD"))),
    "CustomHeader": "Custom header value"
  }
});
```

- Added the ability to specify authentication or other HTTP headers in SupportApi requests. (DOC-3998)

```
// Example: specify basic access authentication and custom headers:
const viewer = new GcPdfViewer("#viewer", {
  supportApi: {
    apiUrl: "192.168.0.1/support-api",
    requestInit: {
      headers: {
        "Authorization": "Basic " + btoa(unescape(encodeURIComponent("USERNAME:PASSWORD"))),
        "CustomHeader": "Custom header value"
      }
    }
  }
});
```

Fixed

- [Editor] Text added in the free text annotation editor disappears on resizing the annotation. (DOC-3988)

[3.0.18] - 24-Jan-2022

Fixed

- Undo history should be cleared by the close() method. (DOC-3989)

[3.0.17] - 21-Jan-2022

Added

- [Form Editor] "Editable" property added to combo boxes. (DOC-3947)
- [Form Editor] New values added to "Tab order" property: "Annotations" and "Widgets". (DOC-3668)
- Added ability to close the current document. (DOC-3981)

```
// Usage example:
await viewer.close();
```

Fixed

- Incorrect tab cycle for an editable combo box. (DOC-3967)
- Tab order differs from Adobe Acrobat Reader in some cases. (DOC-3668)
- [Form Editor] Cannot reset the "Tab order" property to "Not specified". (DOC-3968)
- [Editor] The Backspace key does not work in the free text annotation editor. (DOC-3983)
- [Editor] If a new page is inserted and the document is saved, the tab order of following pages is incorrect. (DOC-3985)

[3.0.13] - 22-Dec-2021

Added

- Added support for editable combo boxes. (DOC-3947)
- [XFA forms] Added the ability to select/copy text content. (DOC-3917)
- [Editor] Remember last used editor values. (DOC-3925)
  New settings added to the "editorDefaults" option: "rememberLastValues" and "lastValueKeys". If "rememberLastValues" is set to true or undefined, the last used property values will be used as default values for new annotations. "lastValueKeys" specifies which properties will be remembered.

```
  // The default value of lastValueKeys:
  ["appearanceColor", "borderStyle", "color", "interiorColor", "backgroundColor", "borderColor", "opacity",
"textAlignment", "printableFlag", "open",
  "lineStart", "lineEnd", "markBorderColor", "markFillColor", "overlayFillColor", "overlayText",
"overlayTextJustification", "newWindow", "calloutLineEnd", "fontSize",
  "fontName", "name", "readOnly", "required"]
```

```
  // Example: turn remembering last used values off:
  var viewer = new GcPdfViewer("#root", { editorDefaults: { rememberLastValues: false } });
```

```
  // Example: remember only the borderStyle property:
  var viewer = new GcPdfViewer("#root", { editorDefaults: { rememberLastValues: true, lastValueKeys:
["borderStyle"] } });
```

- [Android] Added support for zooming using pinch gesture. (DOC-3424, ARD-2508)

Changed

- Japanese localization strings updated.

Fixed

- [iPad] PDF will not render after max zooming when iPad is running in Desktop mode. (DOC-3765)
- [iOS][Android] The main toolbar collapses when the secondary toolbar is clicked. (DOC-3843)
- [Editor] Method showSecondToolbar does not activate editor mode correctly. (DOC-3892)
- [Editor] Incorrect undo / redo behavior when editing ink annotations. (DOC-3924)
- Incorrect zoom controls behavior. (DOC-3929)

[3.0.10] - 25-Nov-2021

Changed

- **Breaking change**: All public APIs that used page numbers now use zero-based page indices. (DOC-3536)
- PDF.js library was updated from v2.0.943 to v2.10.377, see PDF.js Release Notes.
- Method goToPageNumber deprecated, use goToPage method or pageIndex property instead. (DOC-3536)
- By default, radio buttons and checkboxes now do not use predefined appearances from the PDF. Use the fieldsAppearance option to revert to the old behavior:

```
  var viewer = new GcPdfViewer("#root", { fieldsAppearance: { radioButton: "Predefined", checkBoxButton:
"Predefined" } });
```

- [SupportApi client] The ping() method has been deprecated and is no longer used; instead, the serverVersion() method is used.
- Properties SubmitForm/ResetForm renamed to submitForm/resetForm. (DOC-3379)

Added

- Added ability to edit document without switching to Annotation Editor or Form Editor modes. (DOC-3169)
- Added support for second horizontal toolbar.
- Added secondary editing toolbars to the main viewer toolbar: "Text tools", "Draw tools", "Attachments and stamps", "Form tools", "Page tools".
  Use secondToolbarLayout to control which toolbars are enabled.

```
// Example: specify custom second toolbar layout:
viewer.secondToolbarLayout = { "text-tools": ['edit-text', 'edit-free-text'] };
```

- Added API to display a custom second toolbar. (DOC-3170)

```
// Example: create custom second toolbar with key "custom-toolbar-key":
var React = viewer.getType("React");
var toolbarControls = [React.createElement("label", null, "Custom toolbar"),
React.createElement("button", { onClick: () => { alert("Execute action."); }, title: "Action title" },
"Action")];
// Register custom second toolbar for key "custom-toolbar-key":
viewer.options.secondToolbar = {
render: function(toolbarKey) {
   if(toolbarKey === "custom-toolbar-key")
     return toolbarControls;
   return null;
}
};
// Show custom second toolbar:
viewer.showSecondToolbar("custom-toolbar-key");
```

- Added Light and Dark themes. (DOC-3169)
- Support page content accessibility for tagged PDFs containing logical structure information for screen readers.
- Added ability to show the structure tree of tagged PDFs. (DOC-3131)

```
// Use addStructureTreePanel method to add the appropriate panel:
const viewer = new GcPdfViewer(selector);
viewer.addStructureTreePanel();
await viewer.open("sample.pdf");
```

```
// Use structureTree to access available structure tree data:
const viewer = new GcPdfViewer(selector);
await viewer.open("sample.pdf");
```

```
const structureTree = await viewer.structureTree;
if(structureTree) {
console.log(structureTree);
}
```

- Added goToPage method: navigate to the page with a specified 0-based index.

```
// Example: go to the first page:
viewer.goToPage(0);
```

- Added option maxCanvasPixels: maximum supported canvas size in pixels, i.e. width * height. Undefined or -1 means no limit. If the canvas scaling exceeds maxCanvasPixels, the CSS scaling is used instead of re-rendering the page to the canvas. (DOC-3765)
- Added the ability to use GET method to submit a form.

```
// Using GcPdf to create an ActionSubmitForm to submit a form using the GET method:
var actionSubmit = new ActionSubmitForm("/");
actionSubmit.SubmitFormat = ActionSubmitForm.SubmitFormatType.HtmlForm;
actionSubmit.HtmlFormFormat = ActionSubmitForm.HtmlFormFormatFlags.GetMethod;
var btnSubmit = new PushButtonField();
btnSubmit.Widget.Rect = new RectangleF(50, 100, 100, 50);
btnSubmit.Widget.ButtonAppearance.Caption = "Submit";
btnSubmit.Widget.Page = page;
btnSubmit.Widget.Events.Activate = actionSubmit;
```

- Action reset: added support for FieldNames, ExcludeSpecifiedFields and Next properties. (DOC-3379)
- Added new option fieldsAppearance - specifies how form fields are rendered. (DOC-3537)
  Use this option to customize rendering of form fields. Available appearance rendering types:
  - "Custom" - Default. The custom appearance has some improvements over the web appearance, for example you can specify background and border colors.
  - "Web" - Standard form field appearance using native platform styling. See https://developer.mozilla.org/en-US/docs/Web/CSS/appearance for details.
  - "Predefined" - Predefined appearance stream from PDF when available. If the appearance stream is not available, custom appearance will be used.

```
// Example 1: Use platform-native styling for radio and checkbox buttons.
var viewer = new GcPdfViewer("#root", { fieldsAppearance: { radioButton: "Web", checkBoxButton:
"Web" } });
```

```
// Example 2: Use predefined appearance stream for radio buttons:
var viewer = new GcPdfViewer("#root", { fieldsAppearance: { radioButton: "Predefined" } });
```

- Added enableXfa option: render XFA (XML Forms Architecture) forms if any; the default is true. (DOC-3681)

```
// Example: turn XFA forms off:
var viewer = new GcPdfViewer(selector, { enableXfa: false });
```

- Added requireTheme option. Use this option to apply a built-in CSS theme, this will inject the theme styles directly into the page head. Note that only a known built-in theme can be specified, otherwise the viewer will fail to load. Available built-in themes: "viewer", "dark", "dark-yellow", "gc-blue", "light", "light-blue". This option takes precedence over the "theme" option which can be used to specify a custom theme.

```
// Example:
var viewer = new GcPdfViewer(selector, {
     requireTheme: "light"
});
```

- Added onThemeChanged event: raised when the user changes the viewer theme.

```
// Example:
var viewer = new GcPdfViewer(selector, {
     requireTheme: localStorage.getItem('demo_theme') || 'viewer'
});
viewer.addDefaultPanels();
viewer.onThemeChanged.register(function(args) {
   localStorage.setItem('demo_theme', args.theme);
});
```

- Added onInitialized option: the onInitialized handler will be called immediately after the viewer is instantiated.

```
// Example:
var viewer = new GcPdfViewer("#root", {
  onInitialized: (viewer)=> {
    // put viewer initialization code here.
  }
});
```

Fixed

- Multiple bug fixes.

[2.2.19] - 05-Nov-2021

Fixed

- Tab order is inconsistent with Acrobat Reader when using Structure tab order. (DOC-3668)

[2.2.17] - 07-Oct-2021

Added

- Property optionalContentConfig: the optional content (layers) configuration.

```
// Example: hide the optional content group (layer) with id "13R":
const config = await viewer.optionalContentConfig;
config.setVisibility("13R", false);
viewer.repaint();
```

```
// Example: print information about available layers to the console:
const config = await viewer.optionalContentConfig;
console.table(config.getGroups());
```

[2.2.16] - 29-Sep-2021

Changed

- Minor changes for the Japanese release.

[2.2.15] - 15-Sep-2021

Added

- Layers panel: lists and enables users to show/hide individual PDF layers (optional content). (DOC-3539)

```
// Usage example:
viewer.addLayersPanel();
```

- New method openPanel(): opens a side panel.

```
// Usage example:
const layersPanelHandle = viewer.addLayersPanel();
viewer.open("house-plan.pdf").then(()=> {
  viewer.openPanel(layersPanelHandle);
});
```

- New method closePanel(): closes the side panel.

```
// Usage example:
viewer.closePanel();
```

- New method resetChanges(): resets the document to its original state, discarding all changes. (DOC-3608)

```
// Usage example:
await viewer.resetChanges();
```

- New method setPageRotation(pageIndex, rotation): enables users to rotate a specific page in the PDF. (DOC-3632) This method requires SupportApi. Valid values for rotation are 0, 90, 180, and 270 degrees.

```
// Example: set the first page rotation to 180 degrees:
await viewer.setPageRotation(0, 180);
```

- New method getPageRotation(pageIndex): gets the rotation value for a specified page.

```
// Example: get the first page's rotation (degrees):
var rotation = viewer.getPageRotation(0);
```

Fixed

- When an annotation is added in code and saved, its name changes. (DOC-3619)
- The state of the Signature Tool is not cleared after recreating the GcPdfViewer component. (DOC-3621)
- Console shows an error after calling viewer.newDocument(). (DOC-3622)
- Incorrect behavior if viewer.newDocument() is called immediately after opening a PDF. (DOC-3625)

[2.2.14] - 30-Aug-2021

Fixed

- [Editor] Size specified by the editorDefaults.rect property is not applied to target annotation. (DOC-3597)
- If the updateLayout method is called before opening a document, the list of predefined stamps is not loaded. (DOC-3578)
- Console shows multiple error messages when the mouse is moved over a text note annotation. (DOC-3579)
- [Reply Tool] The comment icon is inconsistent with the corresponding annotation. (DOC-3580)
- [Reply Tool] Comments are not loaded when the reply tool is added after opening the document. (DOC-3581)
- The document title is not updated after setting friendlyFileName. (DOC-3577)
- gcpdfviewer.worker.js cannot be loaded from an external CDN URL. (DOC-2804)
- [Android] Viewing file properties raises the Viewer display area, but not the Viewer tool area. (DOC-3529)
- [iOS 12] [Safari] [Editor] Cannot cancel selecting image for stamp annotation. (DOC-3518, DOC-3548)
- In some cases the setPageSize method throws an error. (DOC-3304)
- When the width of the browser is less than 820px, the viewer version cannot be shown. (DOC-3526)
- [Reply Tool] Only sticky notes annotation texts show up in the comments. (DOC-3270)
- [Reply Tool] Replies to text annotations are not deleted correctly. (DOC-2409)
- [Reply Tool] Current user cannot add replies if viewer.options.replyTool.allowAddReplyOtherUser is false. (DOC-3574)

- [Annotation Editor] A sound annotation stops working after editing. (DOC-3569)
- Setting friendlyFileName does not affect the downloaded PDF name. (DOC-3575)

[2.2.11] - 09-Aug-2021

Added

- Added support for predefined stamps. (DOC-3130)
  Use the new stamp option to configure the settings.

```
// Example 1: add two sets of custom stamps with captions 'Okay' and 'Not okay' to the stamps drop-down:
var viewer = new GcPdfViewer("#root", {
  stamp: {
    stampCategories: [
      { name: 'Okay', stampImageUrls: ['http://example.com/stamps/ok.png',
         'http://example.com/stamps/agree.png', 'http://example.com/stamps/fine.png'] },
      { name: 'Not okay', stampImageUrls: ['http://example.com/stamps/stamps/notok.png',
        'http://example.com/stamps/disagree.png', 'http://example.com/stamps/noway.png'] },
    ]
  }
});
```

```
// Example 2: hide the predefined stamps drop-down:
var viewer = new GcPdfViewer("#root", {
  stamp: {
    stampCategories: false
  }
});
```

```
// Example 3: specify image resolution for custom stamps (if unspecified, 72dpi is used):
var viewer = new GcPdfViewer("#root", {
  stamp: {
      dpi: 144,
    stampCategories: [
      { name: 'Stamps', stampImageUrls: ['stamp1.png', 'stamp2.png', 'stamp3.png'] }
    ]
  }
});
```

- Added disableFeatures option. (DOC-3298)
  This option allows disabling certain features (e.g. due to security considerations). Features that can be disabled: 'JavaScript' | 'AllAttachments' | 'FileAttachments' | 'SoundAttachments' | 'DragAndDrop' | 'SubmitForm' | 'Print'.

```
// Example: disable DragAndDrop operations, JavaScript actions, all attachments:
var viewer = new GcPdfViewer("#root", { disableFeatures: ['DragAndDrop', 'JavaScript',
'AllAttachments'] } );
```

- [Editor] Support opacity for annotations. (DOC-2954)
- [Editor] Added font family support for text fields and free text annotations. (DOC-3132)
- [Editor] Added ability to automatically convert the Signature Tool's stamp to content. (DOC-3152)

```
// Usage example:
var viewer = new GcPdfViewer("#root", signTool: { convertToContent: true });
```

- [Editor] Added method setPageSize: allows specifying custom page size for the newPage method. (DOC-2751)

```
// Example: set new page size for the first page:
viewer.setPageSize(0, { width: 300, height: 500 } );
```

- [Form Editor] Added ability to modify a page's tab order. (DOC-3168)
- Added support for Row and Column annotations tab order. (DOC-2797)
- Implemented required validation for comb fields. (DOC-2739)

Changed

- Show a warning message if the connected SupportApi version is out of date.
- [Editor] Improved selection box cursor styles. (DOC-3146)
- [Editor] The floating toolbar is hidden when any editor tool is activated.

Fixed

- Multiple bug fixes.

[2.1.21] - 5-Jul-2021

Fixed

- [Editor] The 'hide annotation' button does not hide annotations. (DOC-3277)
- Combobox dropdown is hidden behind the next page. (DOC-3300)
- Incorrect display of annotations in a rotated PDF. (DOC-3303)
- [Editor] The editor collapses when clicking a thread bead annotation. (Editing thread bead annotations is not supported yet, so they should not be listed by the editor at all.) (DOC-3352)
- [Form Editor] Saving the PDF after changing the fields' tab order produces corrupted PDF. (DOC-3294)
- [Form Editor] Cannot focus on fileds after setting tab order. (DOC-3292)
- [Editor] Cannot save the PDF after modifying tab order. (DOC-3295)
- [Editor] Font name is missing after saving and reloading a PDF. (DOC-3279)
- [Editor] Inconsistent behavior after setting annotation opacity. (DOC-3264, DOC-3265)
- [Editor] No default font name for free text annotations and text fields. (DOC-3273)
- [Editor] The 'hide annotation' button does not hide annotations. (DOC-3277)
- Combobox dropdown is hidden behind the next page. (DOC-3300)
- Incorrect display of annotations in a rotated PDF. (DOC-3303)

Changed

- [Redact annotations] 'Opacity' property label has been changed to 'Fill Opacity'. (DOC-3266)

[2.1.18] - 09-Jun-2021

Added

- Highlight the current document in the Document List panel. (DOC-3243)

Fixed

- Incorrect display of some annotation positions when the annotation size is small. (DOC-3203)
- Incorrect display of comb-text fields. (DOC-3199)
- Signature field's popup still shows when editor mode has changed to view mode. (DOC-1777)
- [Documentation] GcPdfSearcher class is missing. (DOC-3205)
- PolyLine annotations are not hidden when the "Hide annotations/form fields" tool is checked. (DOC-3272)
- The viewer does not initialized correctly when the viewer script is loaded after the DOM has been loaded. (DOC-3283)
- [Editor] The attached image disappears after the undo operation. (DOC-3204)
- [Editor] The appearance of comb text field has a slight glitch after setting back color. (DOC-3194)
- [Editor] Popup does not appear for audio annotation after editing. (DOC-3193)

- [Editor] It doesn't focus on the new added textNote if adding a new textNote annotation with Context Menu. (DOC-3247)
- [Editor] The textNote annotation is not selected after I set any status. (DOC-3246)
- [Editor] The sound annotation exceeds the boundary of the resize handle. (DOC-3195)
- [Editor] Circle Annotations are not resizing on the third page in GcPdfViewer. (DOC-3250)
- PolyLine annotations are not hidden when the "Hide annotations/form fields" is checked. (DOC-3272)
- Viewer is not initialized correctly when the viewer script is loaded after the DOM has been loaded. (DOC-3283)
- [Editor] Fixed multiple issues when editing annotations. (DOC-3204, DOC-3194, DOC-3193, DOC-3247, DOC-3246, DOC-3195, DOC-3250)

[2.1.17] - 20-May-2021

Fixed

- The position of resize handle is incorrect for redact annotation. (DOC-3122)
- The default icon of file attachment annotation is not displayed. (DOC-3118)
- Incorrect X conversion for XYZ destination. (DOC-3137)
- Text position of FreeText annotation is incorrect after saving. (DOC-3141)
- The status icon of a TextNote annotation is not displayed. (DOC-3113)
- The graphical signature stamp disappears after printing. (DOC-3160)
- [Editor] Border style in editorDefaults is ignored. (DOC-3176)
- [Editor] Incorrect display when resizing items. (DOC-3183)
- [Editor] Errors while reordering fields or annotations. (DOC-3155)
- [Editor] Unable to select other annotations after adding new PolyLine annotation. (DOC-3145)
- [Android] The menu does not pop up on a long button press. (DOC-3071)
- [Localization] "Paperclip" icon name is not localized. (DOC-3163)

Added

- Added support for Row and Column annotation tab order. (DOC-2797)
- Implemented required validation for comb fields. (DOC-2739)

[2.1.14] - 23-Apr-2021

Added

- [iOS] Provided an additional UI that can be used to open the file selection dialog on iOS device. (DOC-2878)
- [Viewer] Added the ability to open PDF files with Drag and Drop operation.
- [Editor] Added graphical signature tool. (DOC-2270)
- [Editor] Added stamp annotation support (allows adding images as stamp annotations; images can be converted to content). (DOC-2612)
- [Editor] Added the ability to lock annotations and fields for editing using the 'locked' property. (DOC-2642)

```
// Example:
var viewer = new GcPdfViewer('#root', { supportApi: { apiUrl: 'api/pdf-viewer', webSocketUrl: false } });
viewer.addDefaultPanels();
viewer.addAnnotationEditorPanel();
viewer.addFormEditorPanel();
viewer.addReplyTool();
viewer.onAfterOpen.register(()=>{
  // Lock all text annotations after document open:
  const resultArr = await viewer.findAnnotation(1, // 1 - AnnotationTypeCode.TEXT
    { findField: 'annotationType',
      pageNumberConstraint: 1, findAll: true });
    viewer.updateAnnotations(0, resultArr.map((data)=> { data.annotation.locked = true; return
data.annotation; }));
```

```
});
// Open Annotations.pdf
viewer.open('Annotations.pdf');
```

- [Editor] Added the ability to move objects to next or previous page using the context menu.
- [iOS] [Android] Support for filling PDF forms on phone or tablet. (DOC-2867)
- Link annotations support (DOC-1847):
  - added the ability to create a link annotation using the context menu for the selected text context.
  - added the ability to create a link annotation over any other annotation using the context menu.

```
Explicit destination types description:
  FitV = FitBV    // fit page height
  FitH = FitBH    // fit page width
  Fit = FitB      // fit page
  FitR            // Scroll and zoom rectangle into view.
  XYZ             // Scroll to X, Y and apply Zoom
```

- [Editor] Added the ability to use the Shift key to maintain the aspect ratio of the annotation box during resize.
- JS Actions: GcPdfViewer's methods are now available for JS actions in the PDF.

```
// An example of a JS action to show the signature dialog:
app.showSignTool();
```

- Added new methods:

```
lockAnnotation    // Lock annotation for editing.
unlockAnnotation  // Unlock annotation for editing.
getPageSize       // Returns the page size. By default returns unscaled size,
                  // pass true for the includeScale argument if you want to get the scaled value.
getPageRotation   // Get page view box rotation value.
```

- Added new option: coordinatesPrecision.

```
// Annotation coordinates rounding precision.
// Used by Annotation and Form editors.
// Default is 1 (round fraction part).
// Usage example:
// Change the default rounding precision to 0.001:
var viewer = new GcPdfViewer("#root", { coordinatesPrecision: 0.001 } );
```

- [Editor] Added the ability to change the default size for resize/move selection box handles. (DOC-2888) Use the editorDefaults options to tune resizeHandleSize and moveHandleSize settings. Default value is 8 pixels for resizeHandleSize, and 14 pixels for moveHandleSize.

```
// Sample code:
var viewer = new GcPdfViewer("#root", {
     editorDefaults: {
     resizeHandleSize: 20,
     moveHandleSize: 40,
     dotHandleSize = 20
 },
 supportApi: { apiUrl: 'support-api/gc-pdf-viewer', webSocketUrl: false }
});
```

- [Editor] Added an additional floating bar that contains Pan and Text selection tools. The Floating bar is visible by default in editor mode. You can hide the floating bar with the editorDefaults.hideFloatingBar setting.

```
// Sample code:
var viewer = new GcPdfViewer("#root", {
```

```
    editorDefaults: {
        hideFloatingBar: true
    },
    supportApi: { apiUrl: 'support-api/gc-pdf-viewer', webSocketUrl: false }
  });
```

- The About dialog box now displays the SupportApi version when SupportApi is available.
- Implemented auto-generated export value for a radio button in the same radio group. (DOC-2742)
- Push buttons: added support for MouseUp/MouseDown event actions. (DOC-2826)
- Added the ability to perform custom validation using the validateForm method. (DOC-2256)

```
  // Usage example:
  // Validate all form fields, each field must have the value "YES" or "NO".
  viewer.validateForm((fieldValue, field) => { return (fieldValue === "YES" || fieldValue === "NO") ? true :
"Possible value is YES or NO."; });
```

Changed

- [Android/Chrome] Swipe down does not refresh the page when finger is over the scrolling area.
- [iOS] Default zoom mode for iOS devices changed to PageWidth. (DOC-2938)

Fixed

- [iOS] [Android] Fixed multiple issues that occurred on mobile devices.
- Miscellaneous bug fixes.
- Inline text editor cannot be activated by double-clicking. (DOC-2982)
  - Limitation: inline text editing is not supported on iOS.

[2.0.24] - 20-Mar-2021

Fixed

- [JS Action, regression] app.popUpMenu() is not working correctly. (DOC-2903)
- [Form Filler] The form filler button is disabled in some cases. (DOC-2823)
- [Form Filler] The form filler does not work if the value of a radio field is set to null. (DOC-2824)
- Issues with filling PDF forms on touch devices. (DOC-2867)
- [Safari] Text field margins are too wide. (DOC-2897)

[2.0.21] - 04-Mar-2021

Fixed

- Default theme colors show momentarily before the specified theme is applied. (DOC-2810)
- Screen blinking when adding / removing or updating annotation. (DOC-2785)
- JS Action: isBoxChecked returns an incorrect value in some cases. (DOC-2803)
- [Editor] Some annotation property values are lost after saving and reloading a PDF. (DOC-2806, DOC-2809)

Changed

- A form that fails validation or has empty required fields now cannot be submitted. Instead, the first failed field is focused and highlighted, with a tooltip indicating the error. (DOC-2753)
- Improved the styles of the password dialog box.

Added

- New API method setAnnotationBounds: programmatically set the position and/or the size of an annotation.

```
// Example: move an annotation to the top left corner:
viewer.setAnnotationBounds('1R', {x: 0, y: 0});
```

```
// Example: move an annotation to the bottom left corner:
viewer.setAnnotationBounds('1R', {x: 0, y: 0}, 'BottomLeft');
// Example: set an annotation size to 40 x 40 points:
viewer.setAnnotationBounds('1R', {w: 40, h: 40});
// Example: set an annotation position to x=50, y=50 (origin top/left) and size to 40 x 40 points:
viewer.setAnnotationBounds('1R', {x: 50, y: 50, w: 40, h: 40});
```

Changed

- The repaint method now accepts optional indicesToRepaint argument. Usage example:

```
// Redraw content and annotations for pages with indexes 0 and 3:
viewer.repaint([0, 3]);
```

[2.0.18] - 22-Feb-2021

Fixed

- [Editor] On applying a redact annotation its bounds are displayed incorrectly. (DOC-1976)
- [Editor] File size of a file attachment is still shown after the file has been removed. (DOC-2784)

[2.0.17] - 10-Feb-2021

Fixed

- [Editor] Invalid selection rectangle if a page contains x / y offsets. (DOC-2767)
- [Editor] The confirmation dialog does not appear even though the active document has been modified. (DOC-2744)
- [Editor] Clicking the cancel button in item properties does not cancel the changes. (DOC-2743)
- [Editor] Incorrect download PDF file name if the file has been modified. (DOC-2690)

[2.0.15] - 25-Jan-2021

Fixed

- Cannot check a radio button if appearance state name contains spaces. (DOC-2721)
- Compatibility problems when using GcPdfViewer with Wijmo controls. (DOC-2670)
- Autofocus does not work in FireFox. (DOC-2696)
- Incorrect download PDF file name. (DOC-2690)
- After navigating to an article thread, another thread cannot be activated by clicking on the thread bead. (DOC-2701)
- [SupportApi] In some scenarios submitting a filled form causes out of memory error. (DOC-2718)
- [Search] Issues when using word, wildcard and proximity options. (DOC-2682, DOC-2689, DOC-2722)
- [Editor] Incorrect display of read-only CombText fields. (DOC-2733)
- [Editor] Incorrect callout line endings. (DOC-2723)
- [Editor] Author and subject are not saved for FreeText annotations. (DOC-2686)
- [Editor] Callout line is not adjusted when a FreeText annotation is copied to another location. (DOC-2710)
- [Editor] An error occurs when setting a FreeText annotation's BackColor via API. (DOC-2707)
- [Editor] No default border is shown for newly added FreeText annotations. (DOC-2702)
- [Editor] Tooltip is not shown when hovering the mouse over an annotation after editing. (DOC-2703)
- [Editor] Error occurs when copying a Polygon/Polyline. (DOC-2715)

[2.0.13] - 07-Jan-2021

Fixed

- [Editor] User name is set to StickyNotes annotations only. (DOC-2632)
- [Search] Search results are incorrect if the search text contains different styles. (DOC-2578)

- Copying rich text and paste to notepad, there will be a space. (DOC-2678)

[2.0.12] - 31-Dec-2020

Fixed

- In some cases text copied to clipboard is incorrect. (DOC-2678)
- Invalid request to viewer.css file. (DOC-2668)
- Error when disposing the viewer. (DOC-2630)
- [Editor] User name is set to StickyNotes annotations only. (DOC-2632)
- [Editor] The result of viewer.save() is incorrect. (DOC-2629)
- [Search] Search results for "Starts with" option are incorrect in some cases. (DOC-2579)
- [Search] Search results are missing when searching across lines. (DOC-2566)
- [Search] Search results are incorrect if both "Starts with" and "Ends with" options are set. (DOC-2549)
- [Search] Wildcard search using the $ character in the query is incorrect. (DOC-2568)
- [Search] Search results using the "Whole word" option are incorrect. (DOC-2577)
- [Search] Search results using "Proximity" and "Whole word" options together are incorrect. (DOC-2554, DOC-2578)

[2.0.11] - 09-Dec-2020

Fixed

- Error when drawing a circle annotation. (DOC-2564)
- Validation messages are not displayed in some cases. (DOC-2518)
- The shared author indicator disappears after clicking the delete button. (DOC-2547)
- Viewer display issues (DOC-2544, DOC-2590)
- Text selection persists when a new file is loaded into the viewer. (DOC-2601)
- Errors if the viewer is removed from DOM and added back. (DOC-2587)
- The searched results are incorrect. (DOC-2565)
- [Form Filler] Form filler is not disabled after loading a PDF with fields that have been converted to content. (DOC-2558)
- [Annotation Editor] Errors when a page has two instances of GcPdfViewer. (DOC-2496)
- [Annotation Editor] Japanese text cannot be entered correctly in annotation comment. (DOC-2413)
- Localization issues. (DOC-2607)

Added

- Japanese localization.

[2.0.10] - 22-Nov-2020

Fixed

- Highlighted search results are incorrect in some cases. (DOC-2562)
- [Collaboration] The viewer collapses on connection error. (DOC-2571)

Added

- Added the ability to configure automatic reconnect interval. Use the reconnectInterval setting to configure automatic reconnect interval in milliseconds. The default reconnect interval is 5000 milliseconds (5 seconds).

```
// Example - reconnect after a 1 second timeout:
var viewer = new GcPdfViewer('#root', { supportApi: 'api/pdf-viewer', reconnectInterval: 1000 } );
// Example - disable auto reconnect:
var viewer = new GcPdfViewer('#root', { supportApi: 'api/pdf-viewer', reconnectInterval: -1 } );
```

- Added the ability to configure logging level used for persistent connections.

```
// Example - turn on debug logging for persistent connections:
var viewer = new GcPdfViewer('#root', { supportApi: 'api/pdf-viewer' }, logLevel: 'Debug' );
```

[2.0.9] - 17-Nov-2020

Fixed

- [IE11] Fixed regular expression exception. (DOC-2371)
- In some cases the editor is disabled on first launch. (DOC-2516)
- Unable to scroll and edit the form on mobile devices. (DOC-2502)
- The display of search results on the document is out of position in some cases. (DOC-2398)
- Double-click selects the space at the beginning of a word. (DOC-2487)
- Double-click selects punctuation characters along with letters. (DOC-2488)
- Enabling editing features on localhost incorrectly requires a Pro license. (DOC-2440)

[2.0.3] - 05-Nov-2020

Added

- [Annotation Editor] Added enhanced editorDefaults option that allows setting the default values and other settings of annotations. (DOC-2458)

```
// Example:
var viewer = new GcPdfViewer("#root", {
     editorDefaults: {
         squareAnnotation: {
             borderStyle: { width: 5, style: 1 },
             color: [0, 0, 0],
             interiorColor: [255, 0, 0],
         }
     },
     supportApi: "api/pdf-viewer"
});
```

[2.0.2] - 04-Nov-2020

Fixed

- [Collaboration] The share button does not work. (DOC-2399)
- [Collaboration] Previous access information is displayed for new documents. (DOC-2402)
- [Collaboration] Access permission is not changed in real time co-authoring. (DOC-2401)
- [Collaboration] Inserting a blank page is not reflected in real time co-authoring. (DOC-2410)
- [Collaboration] Focus is missing after creating a FreeText annotation. (DOC-2415)
- [Collaboration] The modifier user name disappears. (DOC-2424)
- [Form Filler] Crash when clicking FormFiller after loading a pdf. (DOC-2430)
- [Form Filler] Multiple property does not work for email type. (DOC-2435)
- [Form Filler] A TextField with search type is not displayed if the maxLength property is less than the minLength property. (DOC-2431)

Added

- Support additional wildcards in license keys. (DOC-2347)
- Support for v2 licenses. Only v2 Pro licenses allow the use of SupportApi except on localhost. (DOC-2440)
- [Collaboration] Added new sharing option setting: presenceMode.

```
// Determines the type of presence for collaboration users. Possible values are:
// - 'on' or true: the presence of all users, including the active one, will be shown.
// - 'others': all users except the active user will be shown.
// - 'off' or false: presence will not be shown.
// Default value is 'on'.
// Example - turn off presence indicators:
var viewer = new GcPdfViewer("#root", {
     sharing: {
         presenceMode: 'off'
     },
     supportApi: "api/pdf-viewer"
});
```

- [Collaboration] Added new sharing option setting: presenceColors.

```
// If specified, these colors will be used as color for presence indicators.
// Dictionary, key - user name, value - color string.
// Usage example:
var viewer = new GcPdfViewer("#root", {
     sharing: {
         knownUserNames: ['Jamie Smith', 'Lisa'],
         presenceColors: { 'Anonymous': '#999999', 'Jamie Smith': 'red',  'Lisa': 'blue' }
     },
     supportApi: "api/pdf-viewer"
});
```

- [Form Filler] showFormFiller method, usage example:

```
if (viewer.hasForm) {
  viewer.showFormFiller();
}
```

[2.0.1] - 20-Oct-2020

Fixed

- [Form Filler] The validationmessage property does not work for a date field. (DOC-2356)
- [Form Filler] Wrong tooltip position for OneColumn layout. (DOC-2386)
- [Form Filler] Incorrect order of custom form fields. (DOC-2384)
- [Form Filler] Required ignored if validateoninput is set in code. (DOC-2358)
- [Collaboration] The share button exists even if it is not configured. (DOC-2377)
- [Annotation Editor] Focus is lost when filling an annotation subject field. (DOC-2389)
- [Annotation Editor] Snap to page edge does not work when the page does not already have annotations or fields. (DOC-2394)
- Predefined annotation appearance rendering improved. (DOC-2395)

Added

- [Form Filler] Added applyAfterFailedValidation option. The type of action to execute if form validation fails after clicking Apply button. Possible values are: 'confirm' | 'reject' | 'apply' or Function, default value is 'confirm',

```
// Examples:
// Reject applying changes if validation failed:
options.formFiller = {
  applyAfterFailedValidation: 'reject'
}
// Apply changes even if validation failed:
options.formFiller = {
  applyAfterFailedValidation: 'apply'
}
```

```
// Execute custom function and reject changes:
options.formFiller = {
  applyAfterFailedValidation: function() {
    alert('Validation failed, changes rejected.');
    return false;
  }
}
// Execute custom function and accept changes:
options.formFiller = {
  applyAfterFailedValidation: function() {
    alert('Validation failed, but changes are accepted.');
    return true;
  }
}
```

[2.0.0] - 16-Oct-2020

Fixed

- [Form Filler] Content cannot be cleared if setting default value for "password" type text field. (DOC-2365)
- [Form Filler] Fixed problem with checkbox and radio buttons checked state.
- [Form Filler] The beforeFieldChange event handler was not called for radio / checkbox inputs.
- [Form Filler] SpellCheck property does not work for text type. (DOC-2367)
- [Form Filler] min and max properties do not work for number type. (DOC-2361)
- Ink annotation renders incorrectly on a rotated page. (DOC-2340)
- Redact works incorrectly on a rotated page. (DOC-2337)
- The size and font of the set placeholder is different from the default placeholder. (DOC-2362)
- There is no differences in appearance between disabled and readonly for email text field. (DOC-2363)
- The checked state of the radio button follows the checked state of the check box. (DOC-2352)
- Input type date does not work in IE11. (DOC-2371)
- Setting an incorrect default value is applied successfully. (DOC-2357)

Added

- [Form Filler] Added title option: specifies the Form Filler window title. Example:

```
options.formFiller = {
  title: 'Please fill the form'
}
```

- [Form Filler] One column layout for Form Filler. (DOC-2375)
- Added layout option: specifies the Form Filler dialog layout.
- Possible values are: 'Auto', 'OneColumn', 'TwoColumns'. The default is 'Auto', which uses 'TwoColumns' for large screens and 'OneColumn' for small ones.

```
// Example:
options.formFiller = {
  layout: 'OneColumn'
}
```

- [Form Filler] Added support for multi-line text fields (see multiline property). (DOC-2366)
- [Form Filler] Added new formFiller event handlers:
- onInitialize: called after the list of fields is loaded and initialized but not yet rendered.
- beforeApplyChanges: called when the Apply button has been clicked and fields validation was successful.

```
// Example:
options.formFiller = {
```

```
  onInitialize: function(formFiller) { },
  beforeApplyChanges: function(formFiller) { }
}
```

- [Form Filler] Added the ability to display custom HTML content for in the field list.

```
// Example:
options.formFiller = {
  mappings: {
    'CustomContent_Info1': {
      type: 'custom-content',
      content: '<span>Some HTML content</span>'
    },
  }
}
```

- [Form Filler] Added the ability to display a full-width field input control without a label. (DOC-1594)
- Ability to resize line annotation. (DOC-2364)
- Added page transformations support to Annotation/Form editors. (DOC-2310)
- Support additional wildcards for license keys. (DOC-2347)

[1.3.2] - 10-Sep-2020

Added

- Form filler: added support for min/max/inputmode properties. (DOC-1594)

Changed

- formFiller mapping settings enhanced.
- formFiller.mapping renamed to formFiller.mappings, see docs/index.html#formfieldmapping for the full specification. (DOC-2311)

[1.3.1] - 28-Sep-2020

Added

- Added Form Filler feature. (DOC-1594)
- Added Collaboration Mode feature. (DOC-1595)
- Added the ability to convert annotations and fields to content. (DOC-1883)

[1.2.97] - 04-Nov-2020

Fixed

- Support additional wildcards in license keys. (DOC-2347)
- [Annotation Editor] Editor tools are disabled when opening a page with the viewer that is initially in edit mode. (DOC-2446)

[1.2.96] - 28-Oct-2020

Fixed

- Line annotation disappeared after setting author and clicking subject field. (DOC-2434)
- If the maximum length of a text field is 0, it is ignored (behavior similar to that of Acrobat Reader). (DOC-2436)

Added

- [SupportApi] Support for token-based authentication. (DOC-2432)

- [Search] Ability to use the "Whole word" search option together with the "Wildcard" search option. (DOC-2397)

[1.2.95] - 26-Oct-2020

Fixed

- The position of redact is incorrect in some cases. (DOC-2405)
- The dashed style for line annotation is incorrect. (DOC-2406)
- The line start and end properties of line annotations do not work. (DOC-2407)
- The number of replies is incorrect. (DOC-2422)
- A checked checkbox cannot be unchecked. (DOC-2420)
- The appearance of read-only checkBox is incorrect. (DOC-2418)
- Incorrect behavior of annotations in rotated documents. (DOC-2393)
- Wildcard search results are incorrect in some cases. (DOC-2397)
- Japanese search is incorrect with some options. (DOC-2403)
- Text annotation comments automatically change when user tries to enter text. (DOC-2411)
- Reply comments to text annotations are not deleted correctly. (DOC-2409)
- Text selection with mouse in a scrolled document is broken (regression added in .93). (DOC-2425)

[1.2.94] - 20-Oct-2020

Fixed

- [Annotation Editor] Focus is lost when filling an annotation subject field. (DOC-2389)
- [Annotation Editor] Snap to page edge does not work when the page does not already have annotations or fields. (DOC-2394)
- Predefined annotation appearance rendering improved. (DOC-2395)

[1.2.93] - 17-Oct-2020

Fixed

- Ink annotation renders incorrectly on a rotated page. (DOC-2340)
- Redact works incorrectly on a rotated page. (DOC-2337)

Added

- Ability to resize line annotation. (DOC-2364)
- Added page transformations support to Annotation/Form editors. (DOC-2310)

[1.2.92] - 04-Sep-2020

Fixed

- Memory leak occurs when clicking a document in the document list. (DOC-2293)
- Inaccurate display of the predefined appearance of a radio button if the radio button contains a border. (DOC-2303)

[1.2.91] - 02-Sep-2020

Fixed

- Fixed regression issue with executing JS actions for checkbox widgets with predefined appearance streams. (DOC-2088)
- The reset action does not work for password fields. (DOC-2273)
- GcPdfViewer.LicenseKey is not accessible in Angular projects. (DOC-2276)
- The "Backcolor" label is incorrect in Japanese locale. (DOC-2275)

[1.2.89] - 06-Aug-2020

Fixed

- Dashed border style for annotation does not work. (DOC-2255)
- Checkboxes rendered incorrectly in some cases. (DOC-2198)
- Building node_modules folder throws errors. (DOC-2232)

Changed

- Updated client API documentation.

Added

- Added the ability to add Ink and Redact annotation tools to the default viewer toolbar.

```
// Usage example:
var viewer = new GcPdfViewer("#root", { supportApi: 'api/pdf-viewer' });
viewer.toolbarLayout = { viewer: { default: ["open", 'edit-redact', 'edit-redact-apply', 'edit-ink']} };
viewer.applyToolbarLayout();
viewer.open("Annotations.pdf");
```

- Added support for RadiosInUnison flag. (DOC-2205)
- Added support for radio button appearance streams. (DOC-2208)
- Added replyTool settings option. (DOC-2245)

```
// The replyTool option spec:
replyTool?: {
  readOnly?: boolean,               /* Default is false */
  allowAddNote?: boolean,           /* Default is true */
  allowChangeUserName?: boolean,    /* Default is true */
  allowAddReply?: boolean,          /* Default is true */
  allowDelete?: boolean,            /* Default is true */
  allowStatus?: boolean,            /* Default is true */
  allowChangeOtherUser?: boolean,   /* Default is true */
  allowDeleteOtherUser?: boolean,   /* Default is true */
  allowStatusOtherUser?: boolean,   /* Default is true */
  allowAddReplyOtherUser?: boolean, /* Default is true */
}
// Usage examples:
//
// Prevent changing or deleting another user's comments:
var viewer = new GcPdfViewer('#root',
  { replyTool: {  allowChangeOtherUser: false, allowDeleteOtherUser: false },
    userName: 'John',
    supportApi: 'api/pdf-viewer' });
viewer.addReplyTool('expanded');
//
// Add reply tool in read-only mode:
var viewer = new GcPdfViewer('#root', {
  replyTool: { readOnly: true },
  userName: 'John',
  supportApi: 'api/pdf-viewer' });
viewer.addReplyTool('expanded');
//
// Add reply tool that does not allow changing the author name or other users' comments:
var viewer = new GcPdfViewer('#root', {
  replyTool: { allowChangeUserName: false, allowChangeOtherUser: false },
  userName: 'John',
  supportApi: 'api/pdf-viewer' });
viewer.addReplyTool('expanded');
//
```

```
  // Hide the "Add Sticky Note" item from the context menu:
  var viewer = new GcPdfViewer('#root', {
    replyTool: { allowAddNote: false },
    userName: 'John',
    supportApi: 'api/pdf-viewer' });
  viewer.addReplyTool();
```

[1.2.88] - 17-July-2020

Fixed

- The context menu closed if onBeforeCloseContextMenu tried to prevent it from closing, for example:

```
   viewer.options.onBeforeCloseContextMenu = function(e) {
     console.log("The context menu should not close.");
     return false;
   };
```

- Fixed uncaught TypeError when option "renderer" is set to "svg", for example:

```
  var viewer = new GcPdfViewer("#root", { renderer: "svg" } );
```

Added

- Client API documentation, see "docs" folder.

[1.2.87] - 15-July-2020

Fixed

- [IE11] Regex syntax error exception. (DOC-2211)

[1.2.86] - 14-July-2020

Fixed

- The page jumps when adding a sound annotation. (DOC-2197)
- The position is changed after moving an ink annotation. (DOC-2196)

[1.2.85] - 13-July-2020

Changed

- The snapAlignment option specification changed, the updated specification:

```
  // The Snap Alignment feature customization settings.
  // The *tolerance* setting is the distance between the edges of two objects within which the object that is being
  // moved or resized snaps to the other object.
  // The *margin* setting is the distance from the target object or page edge to which the edge of the object being moved or resized snaps.
  // The *center* setting allows the user to snap objects to centers of other objects (in addition to edges).
  // By default, snap tolerance is 5pt, snap margin is 10pt, snap to center is true.
  snapAlignment: true | false |
  {
    tolerance: number | { horizontal: number | false, vertical: number | false },
    margin: false | true | number | { horizontal: number | boolean, vertical: number | boolean },
    center: false | true | { horizontal: boolean, vertical: boolean },
  }
```

[1.2.84] - 10-July-2020

Fixed

- The FreeText border color is not applied until back color is not selected. (DOC-2179)

Changed

- Free text annotation: property Color renamed to Backcolor; property Border color replaced by Forecolor. (DOC-2179)

[1.2.83] - 07-July-2020

Fixed

- The color for searched result is green sometimes, but sometimes is black. (DOC-2175)
- Reply behavior correction: when we respond to the selected comment in the response tool, the "In reply to" field now contains the selected comment as a reference (instead of the topmost comment). (DOC-2174)
- The last reply comment is put at the top. (DOC-2174)

[1.2.82] - 06-July-2020

Fixed

- Field under applied redact annotation can be focused. (DOC-1999)
- The state of the comb-text field value is not saved. (DOC-2150)
- The position of cursor is incorrect in comb text field. (DOC-2144)
- Non-Unicode chars in license names not supported. (GCL2-259)
- Checkboxes do not execute additional JavaScript actions. (DOC-2086)
- snapAlignment does not accept value of true. (DOC-2098)
- Setting value for fields which have same names doesn't work. (DOC-2085)
- The activity indicator does not go away if an error occurs when opening a document. (DOC-2109)
- The center alignment isn't displayed if setting snapAlignment: { tolerance: 25 }. (DOC-2114)
- Horizontal Center cannot be displayed when setting snapAlignment: { center: { vertical: true } }. (DOC-2115)
- There's no Horizontal margin if not specified it. (DOC-2116)
- Cannot link to the correct position if clicking the searched result. (DOC-2135)
- The first search result is selected after checking "highlight all" item. (DOC-2136)
- Appearance is incorrect when a radio button is checked until a blank space is clicked. (DOC-2083)
- Modifying "In reply to" to a Child comment, the comment will be disappeared from the comment panel. (DOC-2139)
- Cannot and sticky note when the Form Editor is opened. (DOC-2137)
- Cannot search correct result when the target is in text field. (DOC-2140)

Changed

- Status labels in the annotation popup now display icons instead of text.

Added

- Added ability to specify boolean value for snap tolerance and disable vertical or horizontal snap.
- Comb-text field: added alignment property.
- Field widgets: added support for JavaScript actions from additional-actions dictionary. (DOC-2088)

[1.2.81] - 22-June-2020

Fixed

- Wildcard Search does not consider newlines. (DOC-2075)

- Appearance is incorrect when you check a radio button until you click on a blank. (DOC-2083)
- Data input in a PDF form shows incorrect result. (DOC-2082)

Added

- Search panel: highlight all feature. (DOC-2069)
- Added zoomMode property. Accepted values are: 0 - Value, 1 - PageWidth, 2 - WholePage.

```
  // usage example:
  viewer.zoomMode = 2; // Set zoom mode to 'WholePage'
```

[1.2.80] - 15-June-2020

Added

- Added ability to collapse the property panel pages using chevron icon.
- Annotation/Form editor: added snap alignment. (DOC-1882)
  By default, snap alignment is enabled, the default snap tolerance is 5pt both vertically and horizontally. Default snap margin is 10pt. Snap to center is also enabled. Press Alt key if you wish to temporarily disable snap during resize or move action.
  Use snapAlignment option if you wish to customize snap alignment. Some examples:

```
  // Change snap alignment tolerance:
  var viewer = new GcPdfViewer("#root", { snapAlignment: { tolerance: 25 },  supportApi: 'api/pdf-viewer'});
  // Set vertical and horizontal alignment tolerance individually:
  var viewer = new GcPdfViewer("#root", { snapAlignment: { tolerance: { vertical: 10, horizontal: 50 } },
supportApi: 'api/pdf-viewer'});
  // Disable snap to center of the element:
  var viewer = new GcPdfViewer("#root", { snapAlignment: { center: false},  supportApi: 'api/pdf-viewer'});
  // Enable snap to center of the element for vertical alignment only.
  var viewer = new GcPdfViewer("#root", { snapAlignment: { center: {vertical: true, horizontal: false, } },
supportApi: 'api/pdf-viewer'});
  // Disable snap margin:
  var viewer = new GcPdfViewer("#root", { snapAlignment: { margin: false },  supportApi: 'api/pdf-viewer'});
  // Enable vertical snap margin:
  var viewer = new GcPdfViewer("#root", { snapAlignment: { margin: { vertical: 10, horizontal: false} },
supportApi: 'api/pdf-viewer'});
  // Disable snap alignment feature:
  var viewer = new GcPdfViewer("#root", { snapAlignment: false,  supportApi: 'api/pdf-viewer'});
```

- Annotation/Form editor: arrow keys move the current element, Shift-arrow resize the element. (DOC-1955)
- Viewer search improvements. (DOC-1885)
  - Proximity search
  - Starts with/ends with
  - Wildcards
- Annotation/Form editor: Added the ability to copy and paste annotations or fields. (DOC-1881)
  Related methods and properties:

```
  // Execute Copy action  (Ctrl+C shortcut).
  // @param buffer Optional. Data to copy.
  execCopyAction(buffer?: CopyBufferData): Promise<boolean>

  // Execute Cut action (Ctrl+X shortcut).
  // @param buffer Optional. Data to cut.
  execCutAction(buffer?: CopyBufferData): Promise<boolean>

  // Execute Paste action (Ctrl+V shortcut).
  // @param point Optional. Insert point, note, pageIndex should be specified.
```

```
  execPasteAction(point?: GcSelectionPoint): Promise<boolean>

  // Execute Delete action (DEL shortcut).
  // @param buffer Optional. Data to delete.
  execDeleteAction(buffer?: CopyBufferData): Promise<boolean>

  // Returns selected text.
  getSelectedText(): string

  // Indicates whether the buffer contains any data.
  get hasCopyData: boolean
```

- Annotation/Form editor: Added the ability to clone the current annotation/field.

```
  // Clone annotation or field given by parameter annotation.
  cloneAnnotation(annotation: AnnotationBase)
```

- Added text annotation reply tool (opens on the right side of the viewer). To enable, use the addReplyTool() method. (DOC-1884)

```
  // Enable the Text Annotation Reply Tool.
  // Note that to enable adding, removing or editing replies, the SupportApi must be configured.
  // Otherwise, the reply tool opens in read-only mode.
  // @param sidebarState Optional. Pass 'expanded' value if you wish the Reply tool to be expanded initially.
Default value is collapsed.
  addReplyTool(sidebarState: GcRightSidebarState = 'collapsed')

  // Indicates whether the Reply Tool has been added.
  get hasReplyTool(): boolean

  // Gets right sidebar object. Use this object if you want to manipulate the right sidebar.
  get rightSidebar(): GcRightSidebar

  // Usage example:
  var viewer = new GcPdfViewer("#root");
  viewer.addReplyTool('collapsed');
  viewer.rightSidebar.show('expanded', 'reply-tool');
  viewer.rightSidebar.hide();
  viewer.rightSidebar.expand();
  viewer.rightSidebar.collapse();
  viewer.rightSidebar.toggle();
```

- Added viewer context menu and the ability to customize it. Related properties and methods:

```
  // By default, the viewer uses its own context menu.
  // Set this option to true if you want to use the browser context menu.
  // Please, note that if this option is set to true, some functions of the
  // context menu will not be available (for example, actions of the Editor and Reply tool).
  // The default value is false.
  useNativeContextMenu: boolean;

  // This handler function will be called when the context menu is about to be shown.
  // Function arguments:
  //   items: DropdownItem[],                /* Modify this array if you wish to change the current set of
menu items */
  //   mousePosition: {x: number, y: number}, /* The mouse position */
  //   viewer: GcPdfViewer                   /* The owner GcPdfViewer */
  // You can use this function to customize the context menu.
  // Return false if you want to prevent the context menu from opening.
  onBeforeOpenContextMenu: Function;

  // This handler function will be called when the context menu is about to be hidden.
```

```
  // Function arguments:
  //   viewer: GcPdfViewer                    /* The owner GcPdfViewer */
  // Return false if you want to prevent the context menu from closing.
  onBeforeCloseContextMenu: Function;

  // This code shows how to modify the context menu and add search using Google and Bing search engines:
  viewer.options.onBeforeOpenContextMenu = function (items, mousePosition, viewer) {
      var selectedText = viewer.getSelectedText();
      if (selectedText) {
          // Remove existent items:
          items.splice(0, items.length);
          // Add own menu items:
          items.push({
              type: 'button',
              text: 'Search using Google',
              onClick: function () {
                  window.open('http://www.google.com/search?q=' + encodeURI(selectedText), '_blank');
              }
          });
          items.push({
              type: 'button',
              text: 'Search using Bing',
              onClick: function () {
                  window.open('https://www.bing.com/search?q=' + encodeURI(selectedText), '_blank');
              }
          });
      }
      return true;
  };
```

- Added editorDefaults option, use this option if you wish to change some default editor values.
  For example you can change the default sticky note color:

```
  var viewer = new GcPdfViewer("#root", { editorDefaults: { stickyNote: { color: '#FF0000'} },  supportApi:
'api/pdf-viewer'});
  // This is the default value for the editorDefaults option:
  {
      textAnnotation: {
          color: '#ffdc38',
          contents: ''
      },
      stickyNote: {
          color: '#38e5ff',
          contents: ''
      }
  }
```

- Show the total number of results in the search panel. (DOC-2068)

Changed

- Text annotation: group all replies to an annotation together as threaded comments. (DOC-2023)
- Form editor: field captions in the list now show field names.
- Added the ability to collapse the property panel pages using the chevron icon.

[1.1.70] - 11-June-2020

Fixed

- Form editor close icon is not shown when the editor is opened using the toolbar button. (DOC-2048)
- Default border width value for new checkbox and radio button fields changed from 1 to 0. (DOC-1978)

[1.1.69] - 31-May-2020

Fixed

- Incorrect sticky note icon appearance. (DOC-2010)
- Incorrect file extension when downloading a PDF using the "save" button in IE11 if the HTTP response contains content-disposition header in modern format. (DOC-2009)
- (Safari) Checkbox field cannot be checked. (DOC-2013)
- friendlyFileName option does not affect the file name in the properties dialog. (DOC-2019)
- Cannot rename radio button field to same name to create radio button group. (DOC-2025)
- Pressing the Tab key on the last form field on a page goes back to first form field in the page instead of the first control on the next page. (DOC-2024)

Changed

- Auto-generated IDs in the Annotation Editor simplified.

Added

- Annotations/From editor: added UI notification when opening a PDF that does not allow editing. (DOC-1977)
- Added new properties: fileUrl, fileName. (DOC-2019)

```
    // Gets the URI that was used to open the document.
    // Returns an empty string when the document was opened from binary data.
    get fileUrl(): string

    // Gets the file name that was used to open the document.
    // The file name is determined as follows:
    // - if a local file was opened using the "Open File" dialog, returns the local file name;
    // - else, if a new file was created using the "newDocument" method, returns the argument passed to that
method;
    // - else, if options.friendlyFileName is not empty, returns its value;
    // - else, returns the name generated from the URL passed to the "open" method.
    get fileName(): string

    // Usage example:
    viewer.onAfterOpen.register(function() {
        alert("Opened document, fileUrl: " + viewer.fileUrl + ", fileName: " + viewer.fileName);
    });
    viewer.open('/Annotations.pdf');
```

[1.1.66] - 08-May-2020

Fixed

- No response when the "download" button is clicked in IE11. (DOC-2007)
- FreeText annotation's text box size is incorrect when annotation is updated using the annotation editor in some cases. (DOC-1990)

[1.1.65] - 07-May-2020

Fixed

- Filename is missing the .pdf extension when the "download" tool is used to download the PDF. (DOC-2006)

[1.1.64] - 05-May-2020

Fixed

- Cannot rename a field if its name ends with an underscore and a digit. (DOC-2001)
- Incorrect default field names. (DOC-2002)
- Method 'scrollAnnotationIntoView' scrolls the view even if the annotation is already. (DOC-2003)

Changed

- Default value for new text fields' border color changed to black. (DOC-1978)

[1.1.63] - 04-May-2020

Fixed

- Tab doesn't work if you want to focus the field from last to first field. (DOC-1987)
- The size property of FileAttachment annotation is incorrect after undo. (DOC-1988)
- Some PDFs are displayed incorrectly. (DOC-1990)
- File attachments still show after deleting. (DOC-1973)
- "Unknown Widget" is shown after adding a text field with a duplicate name. (DOC-1991)
- Deleting a form field could delete just the widget, leaving the field. (DOC-1991)
- The Combs of Comb text field are not displayed. (DOC-1989)
- Form field values are not saved when the field's owner is another field. (DOC-1997)
- Forecolor applies to border as well. (DOC-1963)
- Signature is not visible in the thumbnail. (DOC-2000)

Changed

- SupportApi: show an error message when an unlicensed copy of GcPdf is used. (DOC-1818)
- 'Foreground color' property name changed to 'Forecolor' for Text Fields. (DOC-1964)

[1.1.62]

Fixed

- Problem with updating form fields tab order. (DOC-1982)
- Fields reordering issues. (DOC-1984)
- Default value for Text Field border width changed to 1. (DOC-1978)
- The default zoom mode becomes "Fit to page" when option restoreViewStateOnLoad is set to false. (DOC-1981)

Added

- userData option: Arbitrary data associated with the viewer. This data is sent to the server when the document is saved.

[1.1.61]

Fixed

- Incorrect rendering of redact annotations in some PDFs. (DOC-1976)
- Text field border width controls in panel not working correctly. (DOC-1979)

Changed

- Pdf viewer migrated from Preact to React. (DOC-1886)

[1.1.60]

Fixed

- Incorrect signature rendering. (DOC-1974)
- The thumbnail is not focused. (DOC-1972)
- addDocumentListPanel method: fixed problem when documents list opened inside WebForms application.
- event onBeforeOpenFile is not raised when document is opened using ReportDocumentMoniker instance.

Added

- new method: loadDocumentList() // Load updated document list into document list panel.
- new option: author // Optional. Default author name. The option is used by Annotation Editor as default value for 'author' field.

[1.1.59]

Fixed

- Focused property element is not retained after property change. (DOC-1912)
- (Support API) Error message is shown when opening a password-protected file on server. (DOC-1946)

[1.1.57]

Fixed

- Text field background color value is not loaded into property list. (DOC-1872)
- FontFamily is not applied to text fields. (DOC-1861)
- Text justification is not applied to a text field widget when a modified document is saved and loaded into the viewer again. (DOC-1873)
- Focused property element is not retained after property change. (DOC-1912)
- Some GcPdfViewer buttons fire form submit when running inside a WebForms application. (DOC-1922)
- Fixed problem with resolving download action for Web Api 2 (WebForms) routing.
- Pan tool is not activated in some cases.

Added

- Text fields: added support for font color and font size.
- Added new method saveChanges: uploads local changes to server without file download:

```
saveChanges(): Promise<boolean>
```

- Added new method submitForm: submits form to the server.

```
submitForm(submitUrl: string): void
```

- Added new property: canEditDocument // boolean, indicates whether opened document can be edited using SupportApi.
- Localization for progress dialog, property list, thumbnails.

Changed

- Returned value for method save() changed from void to Promise

[1.1.56]

Fixed

- Incorrect thumbnail scale when browser zoom level changed. (DOC-1869)

Changed

- Properties dialog: close button style updated, added missed translation keys

[1.1.55]

Fixed

- Fixed: properties panel does not collapse when button 'back to view tools' clicked in some cases.
- ColorUtils, hexToRgb: fixed problem with incorrect color conversion when color argument is specified using rgb model.

Added

- Properties dialog localization.

Changed

- Properties dialog style updated.
- Annotation/form editor sidebar behavior updated.

[1.1.54]

Fixed

- Link annotation can be navigated in editor mode. (DOC-1848)
- Sidebar pinned state is not retained when switching from editor mode to viewer mode. (DOC-1849)

Added

- Localization support. (DOC-993)
- Display error message when document size exceeds SupportApi server size limit.
- Added ability to edit basic Link annotation properties. (DOC-1847)
- Added new property: viewer.zoomValue // number, gets/sets the current zoom percentage level.

[1.1.53]

Fixed

- Searching for text with whole word on does not work in some cases. (DOC-1824)

Added

- Added ability to open annotation or form editor when no document is loaded. (DOC-1804)
- supportApi option: added ability to suppress error and information messages about Support API availability. (DOC-1806) Example:

```
var viewer = new GcPdfViewer('#root', {supportApi: {apiUrl: 'api/pdf-viewer', suppressInfoMessages: true,
suppressErrorMessages: true }});
```

- Added new property:

```
get hasDocument(): boolean // Returns true if a document is loaded
```

- Added the ability to change annotation printable flag. (DOC-1841, DOC-1840)
- Text fields: added 'Required' property.
- Delete existing, or add a new empty page. (DOC-1699)
- Added confirmation for large attachments (> 4MB). (DOC-1745)
- Added ability to hide signature widgets when 'hide-annotations' button is checked.

- Added new option coordinatesOrigin: the coordinate system into which objects are transformed before they are displayed. Possible values are 'TopLeft' and 'BottomLeft'. Default value is 'TopLeft'. This option is used by Annotation/Form editors. Example:

```
var viewer = new GcPdfViewer("#root", { coordinatesOrigin: 'BottomLeft', supportApi:"api/gcpdfviewer" });
```

- Added new option watermarkBanner: specifies watermark text which will be shown at the bottom/right corner of the viewer. Example:

```
var viewer = new GcPdfViewer('#root', { watermarkBanner: "Document Solutions PDF Viewer v{{version}}." });
```

- Added new event onAfterOpen: occurs when a document has been opened. Example:

```
var viewer = new GcPdfViewer('#root');
viewer.onAfterOpen.register(function() {
  console.log("Document opened.");
});
viewer.open('Test.pdf');
```

- Added Signature annotation verification (available only when supportApi option is set). (DOC-999)
- Added Form editor. (DOC-1597)
- Added Annotation Editor. (DOC-1581, DOC-1582, DOC-1093, DOC-1589, DOC-1588, DOC-1579, DOC-1591, DOC-1599, DOC-1590, DOC-1592, DOC-1580, DOC-1585)
- Added client API to edit annotations programmatically. (DOC-1587)
  - New properties:

```
    // Gets all document annotations.
    get annotations(): Promise<{ pageIndex: number, annotations: AnnotationBase[] }[]>

    // Returns true if document has been changed by user.
    get hasChanges(): boolean

    // Gets a value indicating whether the pdf viewer can undo document changes.
    get hasUndoChanges: boolean

    // Gets a value indicating whether the pdf viewer can redo document changes.
    get hasRedoChanges: boolean

    // Gets current undo changes index.
    get undoIndex(): number

    // Gets total undo changes count.
    get undoCount(): number

    // Gets or sets more precise Edit mode for Annotations or Form editor.
    get/set editMode(): EditMode

    // Gets or sets the layout mode (Viewer, AnnotationEditor or FormEditor).
    // Default layout mode is Viewer.
    get/set layoutMode(): LayoutMode

    // Defines toolbar layout for different viewer modes: viewer, annotationEditor,formEditor.
    // Usage example:
    // viewer.toolbarLayout.viewer.default = ["open", "save"];
    // viewer.toolbarLayout.annotationEditor.default = ["open", "save", "$split", "new-document", "edit-ink",
"edit-text"];
    // viewer.applyToolbarLayout();
    get toolbarLayout(): GcPdfViewerToolbarLayout
```

- New methods:

```
    // Call this method in order to apply changed toolbarLayout.
    applyToolbarLayout()

    // Add annotation to document.
    addAnnotation(pageIndex: number, annotation: AnnotationBase): Promise<boolean>

    // Update annotation.
    updateAnnotation(pageIndex: number, annotation: AnnotationBase): Promise<boolean>

    // Remove annotation from document.
    removeAnnotation(pageIndex: number, annotationId: string): Promise<boolean>

    // Select annotation for editing.
    // @param pageIndex Zero-based page index.
    // @param annotation Id or Annotation object itself.
    selectAnnotation(pageIndex: number, annotation: AnnotationBase | string): Promise<boolean>

    // Reset annotation selection.
    unselectAnnotation()

    // Scroll annotation into view.
    scrollAnnotationIntoView(pageIndex: number, annotation: AnnotationBase)

    // Undo last document change.
    undoChanges()

    // Redo next document change.
    redoChanges()
```

```
    // Changes coordinate system origin for a rectangle given by parameter
    // bounds and returns rectangle value converted to a new coordinate system origin;
    // @param pageIndex Page index (Zero based).
    // @param bounds Bounds array: [x1, y1, x2, y2].
    // @param srcOrigin Source coordinate system origin. Possible values are: 'TopLeft' or 'BottomLeft'.
    // @param destOrigin Destination coordinate system origin. Possible values are: 'TopLeft' or
'BottomLeft'.
    changeBoundsOrigin(pageIndex: number, bounds: number[],
                       srcOrigin: 'TopLeft' | 'BottomLeft', destOrigin: 'TopLeft' | 'BottomLeft'): number[]

    // Returns PDF page's view port information.
    // @param pageIndex Zero-based page index.
    // @returns object containing following fields:
    //   {
    //     viewBox: number[],                // Original page bounds: [x1, y1, x2, y2].
    //                                       // If you want to know original page's width/height, you can get
it using viewBox values:
    //                                       // var pageWidth  = viewBox[2] - viewBox[0];
    //                                       // var pageHeight = viewBox[3] - viewBox[1];
    //     width: number,                    // Current width of the page in user space (scale and rotation
values are applied),
    //     height: number,                   // Current height of the page in user space (scale and rotation
values are applied)
    //     scale: number,                    // Active scale value
    //     rotation: number,                 // Active rotation value
    //  }
    getViewPort(pageIndex: number)

    // Update set of annotations which is located on one page.
    // @param pageIndex
```

```
    // @param annotations
    // @returns Promise, resolved by updated annotation objects.
    public updateAnnotations(pageIndex: number, annotations: AnnotationBase | AnnotationBase[]):
Promise<{ pageIndex: number, annotations: AnnotationBase[] }>

    // Update radio buttons group given by parameter fieldName with new field value.
    // @param fieldName Grouped radio buttons field name
    // @param newValue New fieldValue
    // @param skipPageRefresh boolean. Set to true if you don't need to update page display. Default is
false.
    // @returns Promise, resolved by boolean value, true - radio buttons updated, false - some error
occurred.
    updateGroupFieldValue(fieldName: string, newValue: string, skipPageRefresh?: boolean): Promise<boolean>

    // Find annotation(s) within opened document.
    // Returns promise which will be resolved with search results.
    findAnnotation(findString: string, findParams?: { findField?: 'id' | 'title' | 'contents' | 'fieldName' |
string,
                                                      pageNumberConstraint?: number,
                                                      findAll?: boolean
    }): Promise<{ pageNumber: number, annotation: AnnotationBase }[]>

     // Example: find annotation with id '2R':
        viewer.findAnnotation("2R").then(function(annotation) {  if(annotation)    alert('Annotation
found.');  else    alert('Annotation not found.');  });

     // Example: find annotation with title 'Some Title':
        viewer.findAnnotation("Some Title", {findField: 'title'}).then(function(annotation)  {    } );

     // Example: find all field widgets with name field1:
      viewer.findAnnotation("field1", {findField: 'fieldName', findAll: true}).then(function(dataArray) {
        if(dataArray.length > 0) {
          alert('Found ${dataArray.length} fields, value of the first field is $
{dataArray[0].annotation.fieldValue}');
        } else {
          alert('field1 not found.');
        }
      });
```

- Added additional user-friendly page navigation methods. (DOC-1612)

```
  // Navigate page with specific page number.
  goToPageNumber(pageNumber: number)

  // Navigate first page.
  goToFirstPage()

  // Navigate previous page.
  goToPrevPage()

  // Navigate next page.
  goToNextPage()

  // Navigate last page.
  goToLastPage()
```

```
  // Scroll page into view.
  scrollPageIntoView(params: { pageNumber: number; destArray?: any[]; allowNegativeOffset?: boolean; })
  // @param params object, with parameters:
  //   pageNumber - number. Page number.
  //   destArray - optional. Array with destination information:
  //     destArray[0] - not used, can be null, pdf page reference (for internal use only).
```

```
  //     destArray[1] - contains destination view fit type name:
  //       { name: 'XYZ' }   - Destination specified as upper-left corner point and a zoom factor.
  //       { name: 'Fit' }   - Fits the page into the window
  //       { name: 'FitH' }  - Fits the width of the page into the window
  //       { name: 'FitV' }  - Fits the height of the page into a window.
  //       { name: 'FitR' }  - Fits the rectangle specified by its upper-left and lower-right corner points
into the window.
  //       { name: 'FitB' }  - Fits the bounding box into the window (fits the rectangle containing all
visible elements on the page into the window).
  //       { name: 'FitBH' } - Fits the width of the bounding box into the window.
  //       { name: 'FitBV' } - Fits the height of the bounding box into the window.
  //     destArray[2] - x position offset
  //     destArray[3] - y position offset (note, the lower-left corner of the page is the origin of the
coordinate system (0, 0))
  //     destArray[4] - optional, can be null, contains bounding box width when view name is FitR, contains
scale when view name is XYZ,
  //     destArray[5] - optional, can be null, contains bounding box height when view name is FitR
  // @param allowNegativeOffset optional, boolean, true when negative page offset should be  allowed.
  //
  // Examples:
  // Scroll page into view:
  viewer.scrollPageIntoView( { pageNumber: 10 } )
  // Scroll annotation into view, example:
  var rectangle = annotation.rect;
  var pagePosX = rectangle[0];
  var pagePosY = rectangle[1] + Math.abs(rectangle[3] - rectangle[1]);
  var pageScale = viewer.zoom.mode === ZoomMode.Value ? viewer.zoom.factor / 100.0 || 1.0;
  viewer.scrollPageIntoView({ pageNumber: pageIndex + 1, destArray: [null, { name: "XYZ" }, pagePosX,
pagePosY, pageScale] });
```

- Added editor shortcuts: Del: Delete selected annotation. Esc: Unselect annotation. Ctrl Z: Undo changes. Ctrl Shift Z: Redo changes. Ctrl Y: Redo changes.

[1.0.44]

Fixed

- Incomplete form gets submitted when the PDF form has multiple pages. (DOC-1688)

Added

- Document security: added support for user access restrictions which is defined in PDF document (print/text copying/acro-form filling).
- Document Properties dialog: added 'Security' tab. (DOC-1372)

[1.0.42]

Fixed

- Initial view mode does not set to 'Continuous View' in some cases. (DOC-1488)
- Current page thumbnail does not scroll into view. (DOC-1435)

Changed

- Outline panel: button title changed to "Bookmarks".
- Outline panel: added ability to collapse outline items.
- Attachments panel: added ability to open PDF attachments in the viewer instead of downloading the file.

Added:

- New event onBeforeOpen: occurs immediately before document open.

```
// Usage example:
var viewer = new GcPdfViewer('#root');
viewer.onBeforeOpen.register(function(args) {
  alert("A new document will be opened,\n payload type(binary or url): " + args.type +",\n payload(bytes or
string): " + args.payload);
});
viewer.open('Test.pdf');
```

[0.7.29]

Fixed

- Text outside of the document cannot be selected. (DOC-1413)
- Next page button does not scroll to the page if part of it is already visible. (DOC-1414)
- The icon button in navigation panel don't fully display in edge browser. (DOC-1398)
- (iOS) Document content disappears when changing zoom. (DOC-1408)
- When a form is posted, incorrect values are sent for some radio buttons. (DOC-1388)

[0.7.28]

Fixed

- When a form is posted, incorrect names are sent for combo and list boxes. (DOC-1388)

[0.7.27]

Added

- Document properties and fonts dialog (toolbar button name 'doc-properties'). (DOC-1339)

[0.7.24]

Fixed

- Thumbnails show incorrectly for non-standard page sizes. (DOC-1346)
- Incorrect page orientation in print preview in some cases. (DOC-1347)

[0.7.23]

Fixed

- The zoom out button disappears in Internet explorer. (DOC-1333)
- License icon does not show fully in Edge browser. (DOC-1302)
- Radio button and checkbox cannot be selected in Edge browser. (DOC-1336)
- FreeText callout lines are not shown when the annotation's appearance is specified by an appearance stream. (DOC-1317)

Changed

- If a FreeText annotation has an appearance stream, but it includes text that requires an external CMap that cannot be located, the text specified in the annotation is shown instead of the appearance stream. (DOC-1317)

Added

- Added the ability to hide FreeText annotations using the 'Hide Annotations' button.
- Added support for CID-keyed fonts with external CMap tables. (DOC-1328) New options:
  - cMapUrl - in some cases, predefined character map files are needed to show CJK text output. The cMapUrl option specifies the URL of the folder where the CMap files are located. The default value is 'resources/

bcmaps/'. A CMap specifies the mapping from character codes to character selectors, it is used to extract Unicode text from a PDF document. In most cases the CMaps are fully embedded in the PDF document, but sometimes a CMap in a PDF is specified by a PDF name object, where this name identifies a predefined CMap that should be known to the PDF processor. The compressed predefined CMaps that are included with GcPdfViewer are located in the 'resources/bcmaps/' folder, and have the .bin extension.
  - cMapPacked - if true, the viewer will look for the compressed version of the CMap files (with the extension '.bin'), otherwise the viewer will look for unpacked CMap files without extension. The default value is true.

[0.7.21]

Fixed

- Fixed the Zoom button in IE. (COREUI-14)
- Fixed the Clear button in SearchPanel. (ARJ-650)
- PDF content is not displayed correctly. (DOC-1317)

Changed

- Zoom In/Out buttons behavior correction: The zoom step is now 10% for zoom factors below 100%, and 25% for zoom factors above 100%. Also added 150% zoom value. (ARJ-606)

[0.7.20]

Added

- Added npmjs package. (DOC-1301)

Fixed

- Navigation toolbar buttons are not working properly when document is opened using Ctrl+O.

[0.7.19]

Fixed

- License error message: fixed problem with icon, added line breaks. (DOC-1296)
- License error message shown twice in some cases. (DOC-1297)
- Cannot edit List box form control in GcPdfViewer. (DOC-1235)
- 'Go To Last' button hidden by default in "Toggle FullScreen" view. (DOC-1232)
- Search with "whole word" option result is not correct when it is a form data. (DOC-1233)
- Highlight area is not correct in some scene. (DOC-1257)
- Prompt "invalid license key" when I set evaluation 365 days key for GcPdfViewer. (DOC-1297)
- The appearance of the watermark is not correct. (DOC-1295)
- Script gcpdfviewer.vendor.js is no longer required

[0.7.18]

Added

- Licensing behavior updated. (DOC-1282)

[0.7.17]

Fixed

- 'Hide-annotations' does not hide Normal appearance stream of the redact annotation. (DOC-1283)

[0.7.16]

Added

- Support Normal/Rollover/Down appearance streams for REDACT annotations. (DOC-1283)

[0.7.15]

Added

- GcPdfViewer does not work when scripts loaded from the head tag. (DOC-1286)

[0.7.14]

Added

- Added licensing. (DOC-1282) Example of how to create GcPdfViewer with product license:

```
<script>
  // Add your license
  GcPdfViewer.LicenseKey = 'xxx';
  // Add your code
  window.onload = function(){
    const viewer = new GcPdfViewer("#viewer");
    viewer.addDefaultPanels();
  }
</script>
```

- Added ability to track previously opened PDF file. Note that only files opened from URI can be tracked (not from binary data). Use restoreViewStateOnLoad option in order to restore a previously opened file on next viewer load:

```
  { restoreViewStateOnLoad: true }
  // or
  { restoreViewStateOnLoad: { trackFile: true } }
```

[0.7.13]

Fixed

- Sidebar animation during restore view state disabled
- Document title toolbar item - hover style removed

[0.7.12]

Fixed

- Restore previously opened sidebar correction: do not expand disabled sidebar panel on viewer load.
- Search character * in PDF viewer the search result is not correct. (DOC-1026)
- Correction: Search highlight is not fully visible when it is located at a page bottom edge
- Fixed regression problem with double click selection (added in build 0.7.6)

Changed

Double-click time increased from 300ms to 500ms.

[0.7.11]

Fixed

- Clear search highlight when search panel closed

- Focus search query input when search panel opened
- Restore previously opened sidebar panel on document load (see restoreViewStateOnLoad option).
- There are no response when click "More Results" button after search text. (DOC-1178)

[0.7.10]

Fixed

- Popup title appearance correction: underline now hidden when popup content is empty.
- Fixed: The modification date (if it is available) is not displayed in the popup correctly.
- Z order of redact annotation changed after loaded by JS PDF Viewer. (DOC-1177)

[0.7.9]

Added

- New toolbar button: 'About'.
- Redact annotations. (DOC-1206)
- New option: hideAnnotationTypes - specifies annotation types which will be hidden when the 'hide-annotations' button is checked.

```
// Default value:
['Text', 'FreeText', 'Line', 'Square', 'Circle', 'Polygon', 'Polyline', 'Ink', 'Popup',
'Sound', 'Polygon', 'RadioButton', 'Checkbox', 'PushButton', 'Choice', 'TextWidget', 'Redact']
// Possible values are:
['Text',  'Link',  'FreeText',  'Line',  'Square',  'Circle',  'Polygon',  'Polyline',  'Ink',  'Popup',
'FileAttachment',
'Sound',  'ThreadBead',  'RadioButton',  'Checkbox',  'PushButton',  'Choice', 'TextWidget',  'Redact']
// or 'All' or 'None'
// Examples:
let options = { hideAnnotationTypes: 'All' }; // Hide all possible annotations.
let options = { hideAnnotationTypes: ['PushButton', 'Redact'] }; // Hide Push button and Redact annotations
only.
```

- New property: version - returns current viewer version.
- Added loading icon indicator for doc-title toolbar item.

[0.7.8]

Added

- Added new toolbar button: Hide annotations, button key is 'hide-annotations', use this button if you wish to preview original document without annotations, this can be useful when you view a redacted document. Annotations that are affected by the 'hide-annotations' button: Redact, Sound, Text, FreeText, Line, Square, Circle, Polyline, Ink, Polygon, RadioButton, Checkbox, PushButton, Choice. Annotations that are not affected by the 'hide-annotations' button: FileAttachment, Article thread bead, Link, Signature, Caret, Highlight, Underline, Stamp, Squiggly, StrikeOut, Popup.

[0.7.7]

Added

- Support redact annotations. (DOC-1206)

[0.7.6]

Added

- Display shortcuts in toolbar button tooltips.

Fixed

- Text still selected after mouse click empty area. (DOC-1182)
- Highlight area is not correct when search text contains "-". (DOC-1199)

[0.7.5]

Fixed

- Fixed regression issue with selected text copy feature using Ctrl+C (added in build 0.7.3)

[0.7.4]

Fixed

- Thumbnails panel performance with large pdf. (DOC-1231)

[0.7.3]

Fixed

- restoreViewStateOnLoad option fix: Now the zoom is restored immediately without waiting for the document load. Note that trackScale option should be set to true: `restoreViewStateOnLoad: {trackScale: true}`
- R/Shift+R shortcuts do not work after clicking on toolbar/sidebar.

[0.7.2]

Added

- New method applyOptions: Use this method in order to propagate changed options. Example:

```
var viewer = new GcPdfViewer(selector);
viewer.options.buttons = ['print', 'rotate'];
viewer.applyOptions();
```

Fixed

- Unable to set viewer buttons using the 'buttons' option.
- The Open method does not load PDF file in IE11.
- Appearance for interactive checkbox and radio buttons updated.

[0.7.1]

Changed

- Now pdf viewer throws exception when trying to load a large file (>1000 pages) without specifying background worker (see workerSrc option). (DOC-1230)

[0.7.0]

Fixed

- Viewer hangs when a large linearized PDF is loaded. (DOC-1230)

[0.6.9]

Fixed

- Fixed: thumbnails not shown in IE11.
- Fix regression bug with thumbnails panel activation (added in build 0.6.7).

[0.6.8]

Fixed

- Button boundary and background color are not visible. (DOC-1207)
- Performance issues when using mouse wheel when a large (>1000 pages) PDF is loaded. (DOC-1194)

[0.6.7]

Changed

- Font icons replaced by svg icons for toolbar and sidebar.
- Show file name in the viewer. (DOC-1191)

[0.6.6]

Fixed

- Form data can not be searched. (DOC-1173)
- Default toolbar buttons order changed:

```
['open', '$navigation', '$split', 'textselection', 'pan', '$zoom', '$fullscreen', 'rotate', 'view-mode',
'theme-change', 'print', 'save', 'doc-title']
```

[0.6.5]

Fixed

- Form fields not editable on loading PDF acroform in IE/IE Edge. (DOC-1184)
- ReportListPanel renamed to DocumentListPanel
- Cannot see the choice of list box in print preview clearly. (DOC-1170)
- Extra outline when text widget annotation is focused removed. (DOC-1171)
- Preparing document for printing dialog invoked by "Ctrl + P" hotkey is different from which invoked by push "Print Document" button in the toolbar. (DOC-1181)
- Cannot quit "Loading" state after cancel input password of the pdf file. (DOC-1198)
- "Ctrl + P" hot-key can not take effect if push "Print Document" and cancel print before. (DOC-1185)
- Text still selected after mouse click empty area in the page. (DOC-1182)
- Password value show after load pdf which has text field that set password property to true. (DOC-1186)
- Click outline node has no response which action is navigate to a web site. (DOC-1187)
- Always show "Continuous View" tool tip no matter it is in single page view mode or continuous view mode. (DOC-1179)
- Some form data lost after loaded by JS PDF viewer. (DOC-1169)
- The search dialog is replaced by Search Panel in sidebar. (DOC-1180)
- Can not select text in forms on IE browser. (DOC-1202)
- Text style is not correct in search result. (DOC-1201)
- Can not see attachments after load attach pdf file which has attachments files. (DOC-1188)
- Can not go to destination of the outline node correctly. (DOC-1193)
- Highlight area is not correct when search text contains "-". (DOC-1199)

- Highlight area is not correct in certain scene. (DOC-1200)
- Page Size is not correct when print. (DOC-1192)

Changed

- restoreViewStateOnLoad option: added ability to track theme.

Added

- New option documentListUrl: Url to document list service used by DocumentListPanel The service should return json string with available documents array, e.g.: ["pdf1.pdf", "pdf2.pdf"].

```
// ASP .NET service back-end action sample :
[HttpGet("documents")]
public ActionResult Documents()
{
  var doucmentsFolder = new System.IO.DirectoryInfo("pdfdocs");
  var files = from file in doucmentsFolder.EnumerateFiles("*.pdf") select file.Name;
  return new ObjectResult(files.ToArray());
}
```

[0.6.4]

Added

- Page title will now be shown under thumbnail.
- New toolbar buttons: text selection/pan
- Current article thread indication
- pdf.js updated to latest version:
  - Added CaretAnnotation support
  - SVG renderer enhanced (see renderer option)
  - Show modification date for annotation pop-ups if it is available and can be parsed correctly (date string should be in ISO 8601 standard)
- Added option theme: Use this option to change the default viewer theme.

```
// Example:
var viewer = new GcPdfViewer(selector, { renderInteractiveForms: true, theme: 'themes/light-blue' });
```

- Added option restoreViewStateOnLoad: Track viewer state changes, save state to local storage and restore on first load. Set this option to false if you wish to disable viewer state changes tracking.
- Added method setTheme(): use to change the active theme.
- Added property rotation: number, use to get/set the current document rotation in degrees.
- Added even onError: indicates error event.

```
// Event arguments:
  { message: string, source: GcPdfViewer, type: 'open' | string, exception?: any }
// Usage example:
  var viewer = new GcPdfViewer("#root", { keepFileData: true });
  viewer.onError.register(handleError);
  function handleError(eventArgs) {
    if (eventArgs.message.indexOf("Invalid PDF structure") !== -1) {
      var message = eventArgs.message;
      var fileData = eventArgs.source.fileData;
      if (fileData) {
        message = new TextDecoder('utf-8').decode(fileData);
      }
      eventArgs.source.reportError({ message: message, severity: "warn" });
    }
  }
```

Fixed

- Thread beads should be hidden during printing. (DOC-1163)
- Page number is not correct in the toolbar when read as article thread mode. (DOC-1164)
- (IE11/Edge) Fixed problem with article threads. (DOC-1167)
- Do not draw border when color is not specified even if style and width specified. (DOC-1172)
- Can not input data in form control under IE and edge browser. (DOC-1176)
- Can not input data in form control when select pan tool in the toolbar. (DOC-1166)
- Do not jump to other page when rotate page. (DOC-1165)
- Fixed: S, H keyboard shortcuts does not update toolbar.
- Fixed: Open SearchPanel using Ctrl+F.
- Fixed problem with thumbnails title when document contains page labels
- Fixed problem with externalLinkTarget option.
- Document title loading text correction - show file name instead of full URL.
- Article threads: by default article title should be empty
- Fixed problem with article threads navigation when one page contains more than one overlapped beads from different articles.
- Added error handling for Reports List fetch.

[0.6.3]

Added

- New method addDefaultPanels: Add default set of sidebar panels with default order.

```
// Usage example:
const viewer = new GcPdfViewer("#viewer1", { file: 'file1.pdf' });
viewer.addDefaultPanels();
```

Changed

- Pages rendering priority updated: navigated page takes priority over other pages, even when this page is less visible compared to others.
- Zoom using two finger gesture for mobile devices.

Fixed

- Fixed several IE 11 issues.
- Fixed problem with context menu for selected text.
- Behavior for sidebar buttons state updated.
- Behavior of the canvas selection for paragraph corner corrected.
- Fixed exception when another document loaded

[0.6.2]

Fixed

- Caret behavior for Ctrl+Home, Ctrl+End keys updated.

[0.6.1]

Added

- Initial version.