

USER GUIDE

# Essential Studio for File Formats Java

---

Version - v19.3.0.53 | Release Date - November 12, 2021



|   |    |
|---|----|
| Welcome to Syncfusion Essential File Formats Java Platform.....       | 8  |
| How to best read this user guide .....                                | 8  |
| Additional help resources .....                                       | 8  |
| Create a support incident .....                                       | 8  |
| List of File Format libraries: .....                                  | 8  |
| Quick Start links: .....  | 8  |
| Installation and Upgrade.....   | 8  |
| System Requirements For Java FileFormats .....                        | 8  |
| Download FileFormats Installer .....                                  | 9  |
| Download the Trial Version .....                                      | 9  |
| Download the License Version.....                                     | 12 |
| Installation using Web Installer .....                                | 12 |
| Installation .....  | 12 |
| Uninstallation.....   | 24 |
| Installation using Offline Installer .....                            | 34 |
| Installing with UI .....  | 34 |
| Installing in silent mode .....                                       | 45 |
| Common Installation Errors .....                                      | 45 |
| Unlocking the license installer using the trial key.....              | 46 |
| License has expired .....   | 46 |
| Unable to find a valid license or trial .....                         | 47 |
| Unable to install because of another installation.....                | 48 |
| Unable to install due to controlled folder access .....               | 50 |
| Upgrading Syncfusion FileFormats .....                                | 51 |
| Upgrading to the latest version .....                                 | 51 |
| Upgrade from trial version to license version.....                    | 51 |
| Available Syncfusion Java packages.....                               | 52 |
| Configure to download Syncfusion Java packages from Apache Maven..... | 52 |
| Configure to download Syncfusion Java packages from Gradle .....      | 52 |
| Licensing.....  | 53 |
| Syncfusion Licensing in Java FileFormats.....                         | 53 |
| License Key Generation.....   | 54 |
| License Key Registration .....  | 54 |
| Common Licensing Errors .....   | 54 |
| License key not registered .....                                      | 55 |



|   |    |
|---|----|
| Invalid key .....   | 55 |
| Trial Expired .....   | 55 |
| Platform Mismatch .....   | 55 |
| Version Mismatch .....  | 55 |
| Could not load Syncfusion.Licensing.dll assembly version...? .....  | 56 |
| Licensing FAQ.....  | 56 |
| How to upgrade from Trial version after purchasing a license? ..... | 56 |
| Where can I get a license key? .....                                | 56 |
| Word Library .....  | 56 |
| Overview of Word library (DocIO) .....                              | 56 |
| External Jars Required .....  | 57 |
| Supported Java versions .....                                       | 57 |
| Download Jars .....   | 57 |
| Getting Started of Word library (DocIO).....                        | 58 |
| Creating a new Word document with few lines of code .....           | 58 |
| Creating a new Word document from scratch with basic elements ..... | 59 |
| Performing Mail merge .....   | 63 |
| Document Object Model representation .....                          | 70 |
| Loading & saving document.....                                      | 72 |
| Opening an existing document .....                                  | 72 |
| Opening an existing document from Stream.....                       | 72 |
| Opening the read only Word document .....                           | 72 |
| Saving a Word document to file system .....                         | 73 |
| Saving a Word document to Stream .....                              | 73 |
| Sending to a client browser .....                                   | 73 |
| Closing a document.....   | 74 |
| Working with Word document .....                                    | 74 |
| Iterating through document elements .....                           | 74 |
| Cloning a Word document .....                                       | 78 |
| Merging Word documents .....  | 79 |
| Working with Styles .....   | 80 |
| Working with Word document properties .....                         | 82 |
| Working with Content Type Properties .....                          | 83 |
| Setting the Background for a Word document.....                     | 85 |
| Working with Alternate chunks .....                                 | 86 |



|  |     |
|--|-----|
| Working with Sections .....                              | 87  |
| Specifying Page Properties.....                          | 88  |
| Creating Multi-column document.....                      | 88  |
| Creating document with different page settings.....      | 89  |
| Working with Headers and Footers .....                   | 89  |
| Adding Page Numbers.....                                 | 93  |
| Removing a Section.....                                  | 95  |
| Working with Paragraph .....                             | 95  |
| Applying paragraph formatting.....                       | 96  |
| Working with text .....                                  | 99  |
| Working with Images .....                                | 101 |
| Working with lists .....                                 | 105 |
| Working with hyperlinks .....                            | 111 |
| Working with symbols.....                                | 114 |
| Appending breaks .....                                   | 115 |
| Working with Text Box.....                               | 115 |
| Working with Tables in Word document.....                | 117 |
| Nested Table .....                                       | 120 |
| Align text within a table .....                          | 120 |
| Apply formatting to Table, Row and Cell .....            | 122 |
| Merging cells vertically and horizontally .....          | 125 |
| Specifying table header row to repeat on each page ..... | 127 |
| Keeping rows from breaking across pages .....            | 128 |
| Iterating through table elements.....                    | 128 |
| Removing the table .....                                 | 129 |
| Working with Bookmarks.....                              | 129 |
| Adding a bookmark .....                                  | 129 |
| Obtaining a bookmark instance .....                      | 130 |
| Removing a Bookmark from Word document.....              | 130 |
| Retrieving contents within a bookmark.....               | 130 |
| Retrieving bookmark contents within a table.....         | 132 |
| Inserting content into a bookmark .....                  | 133 |
| Deleting content from a bookmark .....                   | 136 |
| Replacing content in a bookmark .....                    | 136 |
| Working with document Fields.....                        | 138 |



|   |     |
|---|-----|
| Adding fields.....  | 139 |
| Formatting fields .....   | 139 |
| Updating fields .....   | 140 |
| IF field.....   | 140 |
| Document variables .....  | 141 |
| Cross reference .....   | 142 |
| Unlink fields.....  | 142 |
| Sequence Field .....  | 143 |
| Working with Find and Replace .....                               | 154 |
| Finding contents in a Word document .....                         | 155 |
| Find and replace text with other text .....                       | 156 |
| Find and replace text with an image.....                          | 157 |
| Find and replace a pattern of text with a merge field .....       | 157 |
| Find and replace text with a table .....                          | 158 |
| Find and replace text in Word document with another document..... | 160 |
| Find and replace text extending to several paragraphs.....        | 161 |
| Find text in a Word document and format .....                     | 161 |
| Working with Shapes in Word document.....                         | 162 |
| Adding shapes .....   | 162 |
| Grouping shapes .....   | 164 |
| Ungrouping shapes .....   | 168 |
| Working with Mail merge .....                                     | 168 |
| Mail merge process.....   | 168 |
| Simple Mail merge .....   | 170 |
| Performing Mail merge for a group .....                           | 170 |
| Performing Nested Mail merge for group .....                      | 170 |
| Performing Mail merge with business objects .....                 | 171 |
| Performing Nested Mail merge with relational data objects .....   | 171 |
| Event support for mail merge .....                                | 171 |
| Mail merge options .....  | 171 |
| Working with Watermark .....                                      | 172 |
| Text watermark.....   | 172 |
| Picture Watermark.....  | 172 |
| Working with Comments .....                                       | 173 |
| Adding a Comment .....  | 173 |



|   |     |
|---|-----|
| Modifying a Comment .....                         | 173 |
| Removing Comments .....                           | 174 |
| Accessing parent comment .....                    | 174 |
| Working with Form Fields .....                    | 175 |
| Check Box .....                                   | 175 |
| Drop-Down .....                                   | 176 |
| Text Form field .....                             | 177 |
| Working with Security .....                       | 179 |
| Encrypting with password .....                    | 179 |
| Opening the encrypted Word document .....         | 179 |
| Remove encryption .....                           | 179 |
| Protecting Word document from editing .....       | 179 |
| Working with Content Controls .....               | 180 |
| What is Content Control? .....                    | 180 |
| Common properties of Content Control .....        | 181 |
| Why Content Control? .....                        | 182 |
| Types of Content Controls .....                   | 191 |
| Working with Mathematical Equation .....          | 194 |
| Types of equation .....                           | 195 |
| Modify existing equation .....                    | 208 |
| Accepting or Rejecting Track Changes .....        | 210 |
| Accept all changes .....                          | 211 |
| Reject all changes .....                          | 211 |
| Accept all changes by a particular reviewer ..... | 212 |
| Reject all changes by particular reviewer .....   | 212 |
| Revision information .....                        | 213 |
| Word Document Conversion .....                    | 213 |
| Working with Document Conversions .....           | 213 |
| HTML conversion .....                             | 214 |
| Text file .....                                   | 214 |
| Supported and Unsupported Features .....          | 214 |
| Word document features .....                      | 214 |
| Frequently Asked Questions .....                  | 220 |
| How to modify an existing style? .....            | 220 |
| How to set OpenType Font Features? .....          | 221 |



|  |     |
|--|-----|
| How to insert a table from HTML string in Word document? .....   | 223 |
| How to set table cell width? .....   | 224 |
| How to position a table in a Word document?.....   | 224 |
| How to set the text direction to a table in Word document? .....   | 225 |
| How to remove headers and footers from the document? .....   | 225 |
| Which units does Java Word library uses for measurement properties such as size, margins, etc, in a Word document? ..... | 226 |
| Migration from Microsoft Office Automation to Essential DocIO .....  | 226 |
| How to check whether a Word document contains tracked changes or not? .....  | 229 |
| How to accept or reject track changes of specific type in the Word document? .....                                       | 229 |
| How to enable track changes for Word document? .....   | 230 |



## Welcome to Syncfusion Essential File Formats Java Platform

Essential File Formats for Java includes a Java class library to create, edit, write, and convert Word file formats in Java applications without Microsoft Office or interop dependencies.

### How to best read this user guide

- The best way to get started would be to read the “Getting Started” section for the component you would like to start first. The “Getting Started” section gives enough information, so it is recommended to read this section end-to-end before starting to write a code. All other information can be referred as needed.
- After learning the basics about the component, integrate the component into your application. A good starting point is to refer to the code examples in the sample browser and in this user guide. It is very likely that you can find a code example that resembles your intended usage scenario.

### Additional help resources

The [Knowledge Base](#) section contains responses for common questions asked by the customers. This would be a good place to search for the topics that are not covered in the User Guide.

Similar to the Knowledge Base, the [Forum](#) section also contains responses to the questions that are asked by other customers.

### Create a support incident

If you are unable to find the information you are looking for in the self-help resources mentioned above, please contact us by creating a [support ticket](#).

### List of File Format libraries:

| File Format library            | Description   |
|--------------------------------|---|
| Word Library (Essential DocIO) | Java class library used to create, read, edit, and convert Microsoft Word files in Java applications. |

### Quick Start links:

[Create Word file in Java without Microsoft Office](#)

## Installation and Upgrade

### System Requirements For Java FileFormats

The system requirements for using our Syncfusion Java FileFormats platform are as follows

#### Operating Systems

##### Windows

- Windows 10 (x86, x64).
- Windows 8, 8.1(x86, x64).
- Windows 7 SP1(x86, x64).
- Windows Vista SP2(x86, x64).
- Windows Server 2019 (x64).



- Windows Server 2016(x64).
- Windows Server 2012 R2 (x64).
- Windows Server 2012 (x64).
- Windows Server 2008 R2 SP1 (x64).

#### macOS

- macOS version 10.9 (Mavericks) and later (x64).

#### Linux

- Ubuntu Linux 12.04 LTS and later (x64, x82).
- Suse Linux Enterprise Server 10 SP2 and later (x64, x82).
- Red Hat Enterprise Linux 5.5 and later (x64, x82).
- Oracle Linux 5.5 and later (x64, x82).

#### Hardware Environment

- Processor: x86 or x64
- RAM: 512 MB (minimum), 1 GB (recommended)
- Hard disk: up to 8.5 GB of available space may be required. However, 750 MB free space is required in boot drive even if you are installing in other drive.

#### Supported Java versions

Syncfusion Java libraries supports Java SE 8.0(1.8) or above versions.

#### Download FileFormats Installer

[DocIO](#), [PDF](#), [Presentation](#), and [XlsIO](#) controls will be included in the Syncfusion FileFormats installer. You can either download the licensed installer or try our trial installer depending on your license.

- Trial Installer - Licensed Installer

#### Download the Trial Version

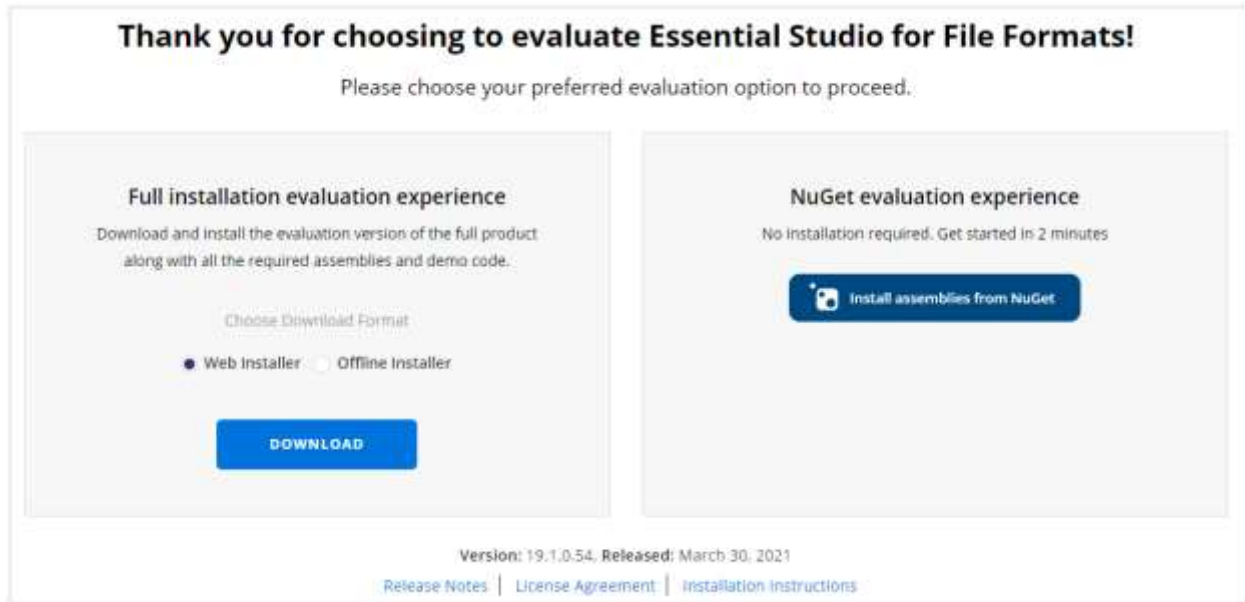
Our 30-day trial can be downloaded in two ways.

- Download Free Trial Setup
- Start Trials if using components through [maven repository](#)

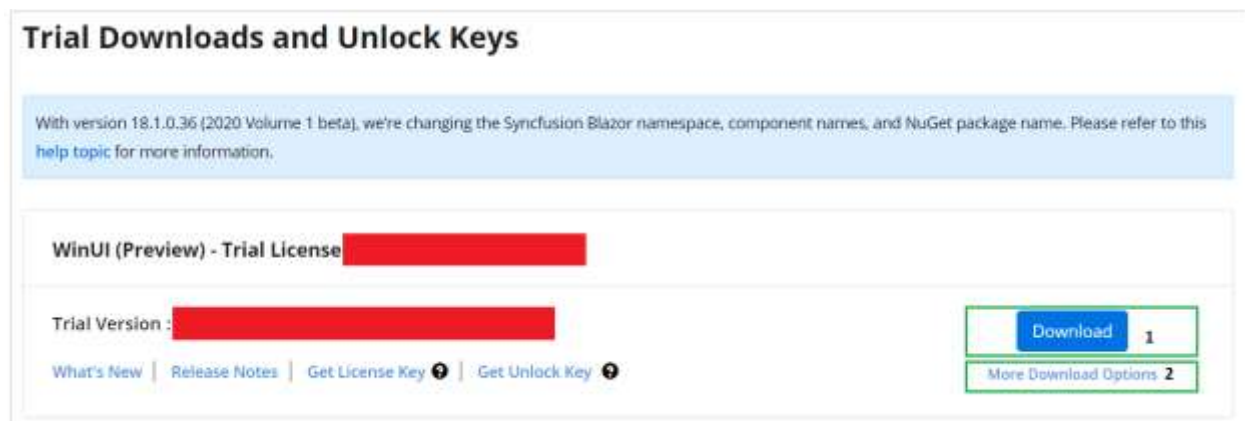
#### Download Free Trial Setup

1. You can evaluate our 30-day free trial by visiting the [Download Free Trial](#) page and select the FileFormats platform.
2. After completing the required form or logging in with your registered Syncfusion account, you can download the FileFormats trial installer from the confirmation page. (See the screenshot below.)



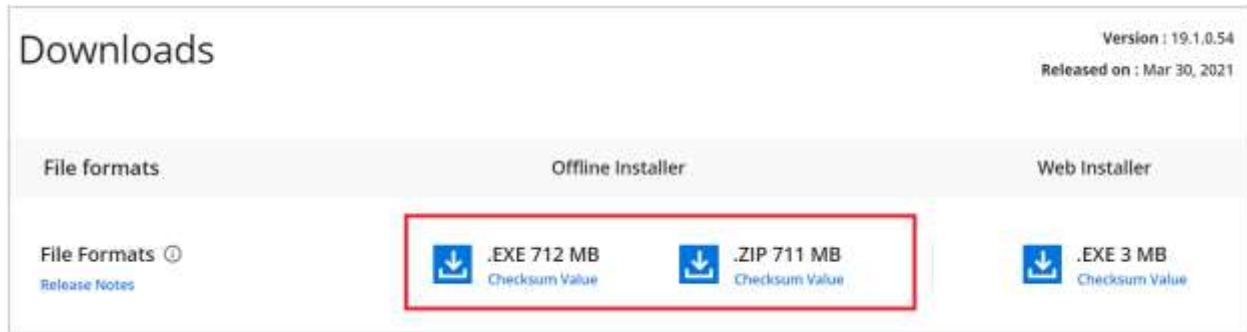


3. With a trial license, only the latest version's trial installer can be downloaded.
4. After downloading, the Syncfusion FileFormats trial installer can be unlocked using either the trial unlock key or the Syncfusion registered login credential. More information on generating an unlock key can be found in [this](#) article.
5. Before the trial expires, you can download the trial installer at any time from your registered account's [Trials & Downloads](#) page (See the screenshot below.)
6. Click the Download (element 1 in the screenshot below) button to get the Syncfusion Essential Studio FileFormats web installer.



7. Click the More Download Options (element 2 in the above screenshot) button to get the Essential Studio FileFormats Offline trial installer which is available in EXE and ZIP format.





Start Trials if using components through [maven repository](https://jars.syncfusion.com)

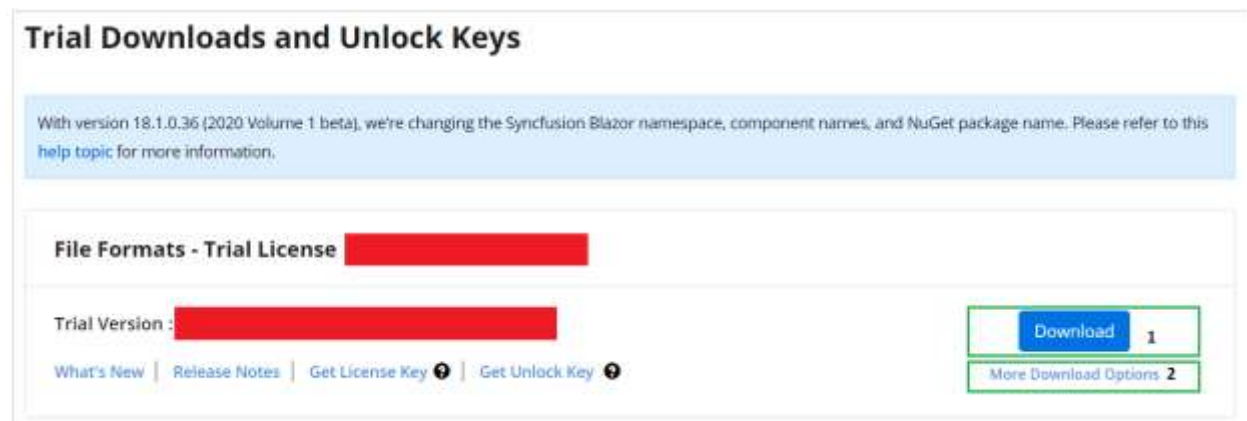
1. You can start your 30-day free trial for FileFormats from the [Start Trial](#) page from your account.



2. To access this page, you must sign up\log in with your Syncfusion account.
3. Begin your trial by selecting the FileFormats product.

**Note:** If you've already used the trial products and they haven't expired, you won't be able to start the trial for the same product again.

4. After you've started the trial, go to the [Trials & Downloads](#) page to get the latest version trial installer. You can generate the [unlock](#) key here at any time before the trial period expires. (See the screenshot below.)



5. You can find your current active trial products on the [Trials & Downloads](#) page.



### Download the License Version

1. Syncfusion licensed products will be available in the [License & Downloads](#) page under your registered Syncfusion account.
2. You can view all the licenses (both active and expired) associated with your account.
3. Click the Download (element 1 in the screenshot below) button to download the respective product's installer.
4. The most recent version of the installer will be downloaded from this page.
5. To download older version installers, go to [Downloads Older Versions](#) (element 2 in the screenshot below).
6. You can download other platform\add-on installers by going to More Downloads Options (element 3 in the screenshot below).
7. For Windows OS, EXE and Zip formats are available for download. They are both Offline Installers.



You can also refer to the [Online installer](#) and [Offline installer](#) links for step-by-step installation guidelines.

### Installation using Web Installer

You can refer to the [Download](#) section to learn how to get the FileFormats trial or licensed installer.

#### Overview

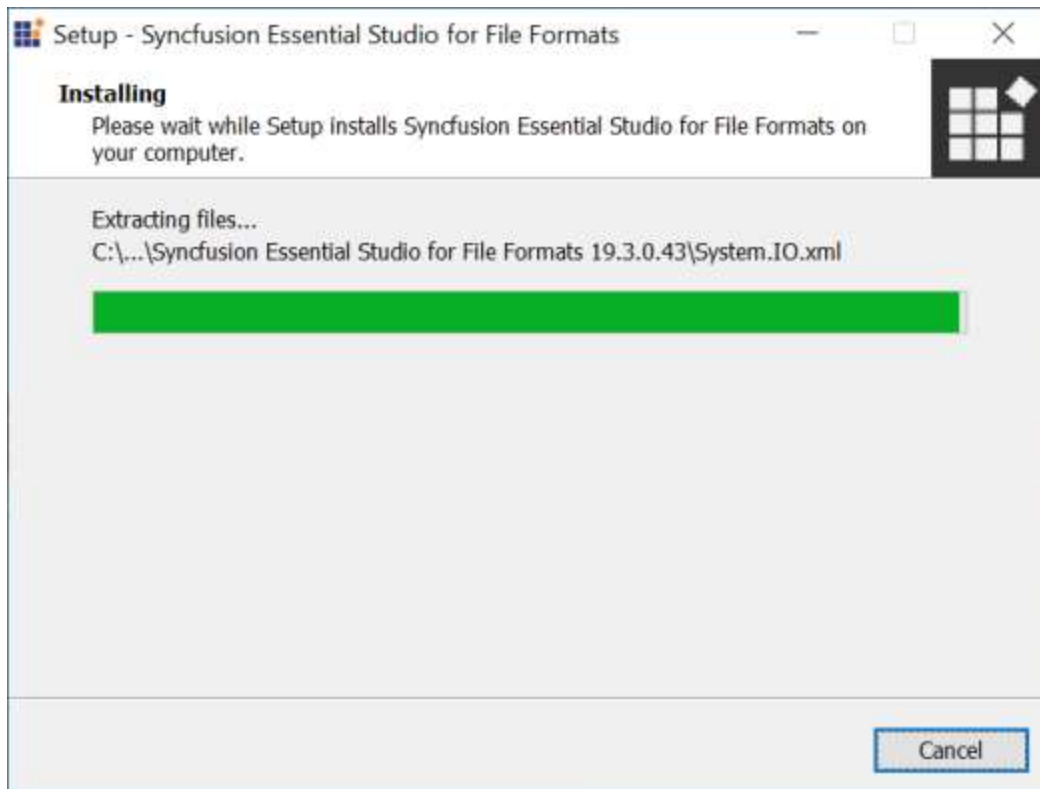
For the Essential Studio FileFormats product, Syncfusion offers a Web Installer. This installer alleviates the burden of downloading a larger installer. You can simply download and run the online installer, which will be smaller in size and will download and install the Essential Studio products you have chosen. You can get the most recent version of Essential Studio Web Installer [here](#).

#### Installation

The steps below show how to install Essential Studio FileFormats Web Installer.

1. Open the Syncfusion Essential Studio FileFormats Web Installer file from downloaded location by double-clicking it. The Installer Wizard automatically opens and extracts the package.

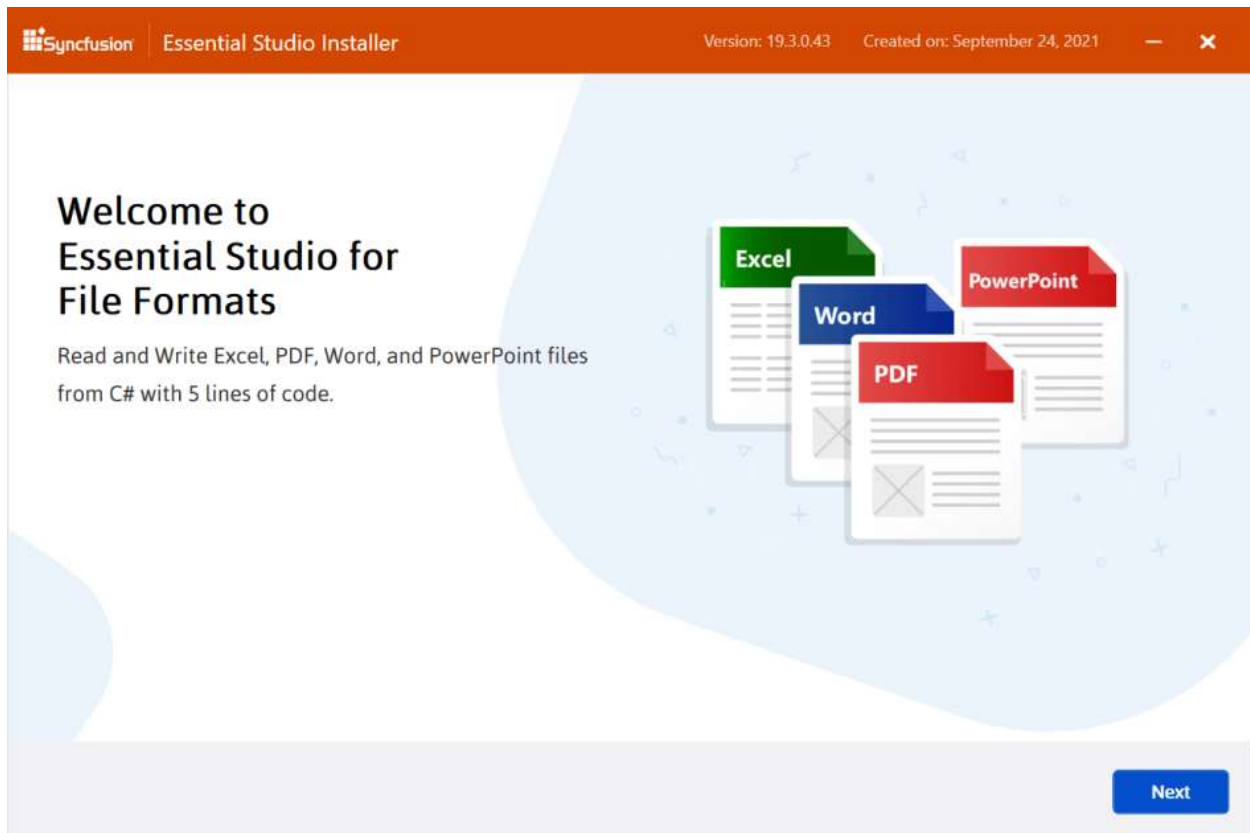




**Note:** The installer wizard extracts the syncfusionessentialfileformatswebinstaller\_{version}.exe dialog, which displays the package's unzip operation.

2. The Syncfusion FileFormats Web Installer's welcome wizard will be displayed. Click the Next button.

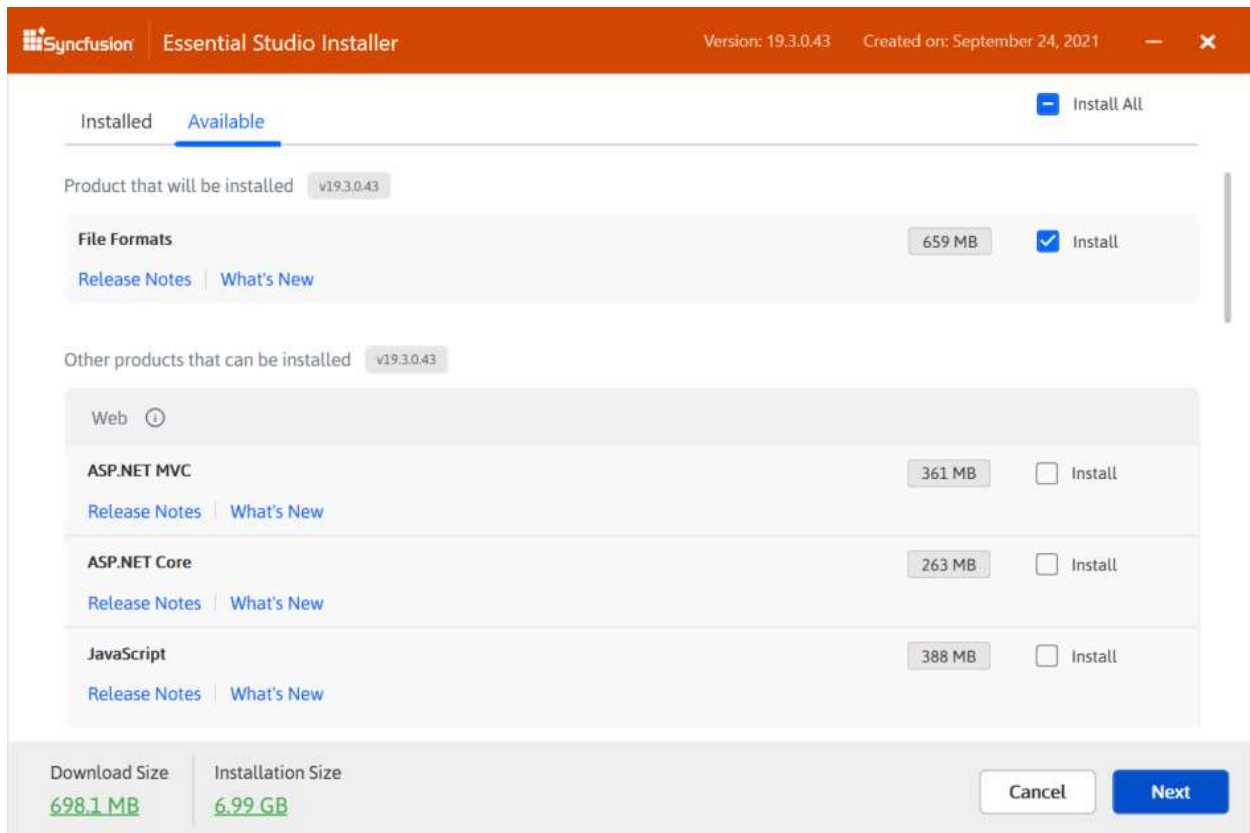




3. The Platform Selection Wizard will appear. From the **Available** tab, select the products to be installed. Select the **Install All** checkbox to install all products.

<em>Available</em>

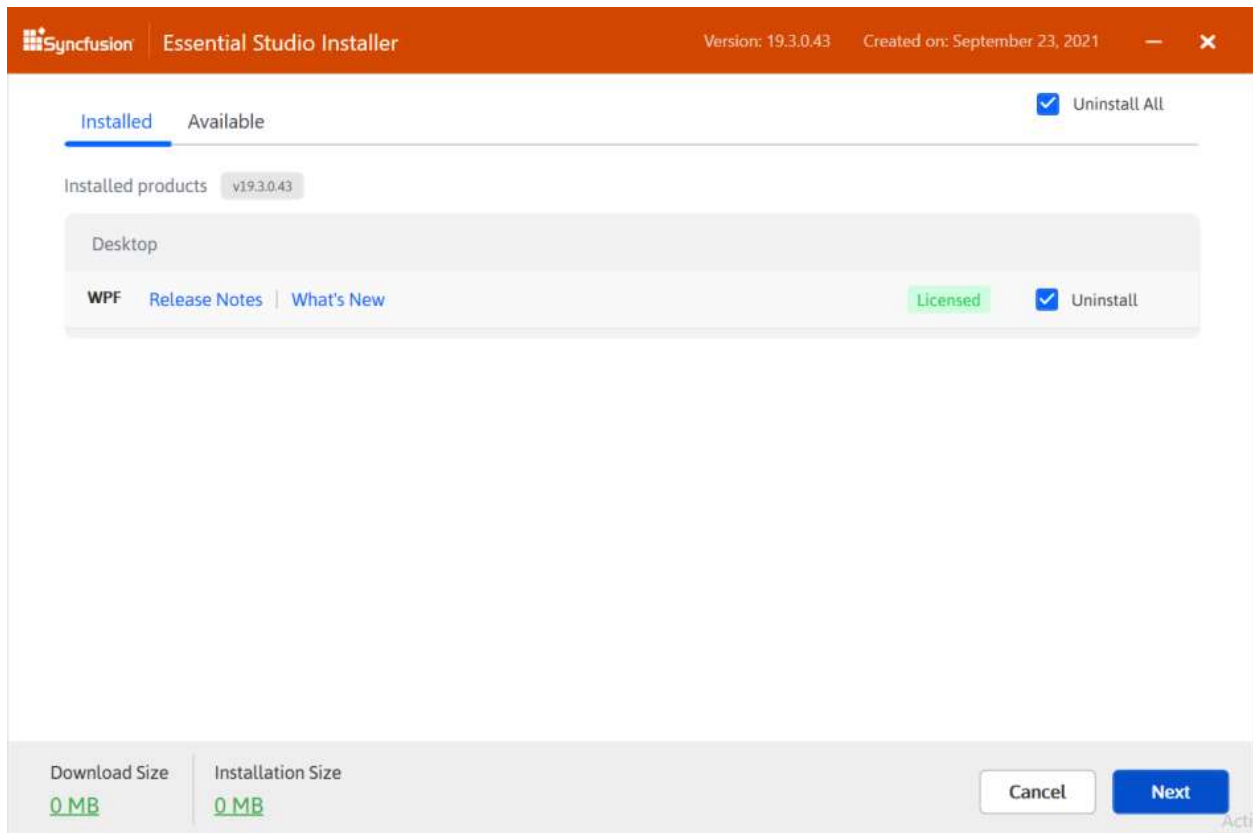




If you have multiple products installed in the same version, they will be listed under the **Installed** tab. You can also select which products to uninstall from the same version. Click the Next button.

<em>Installed</em>

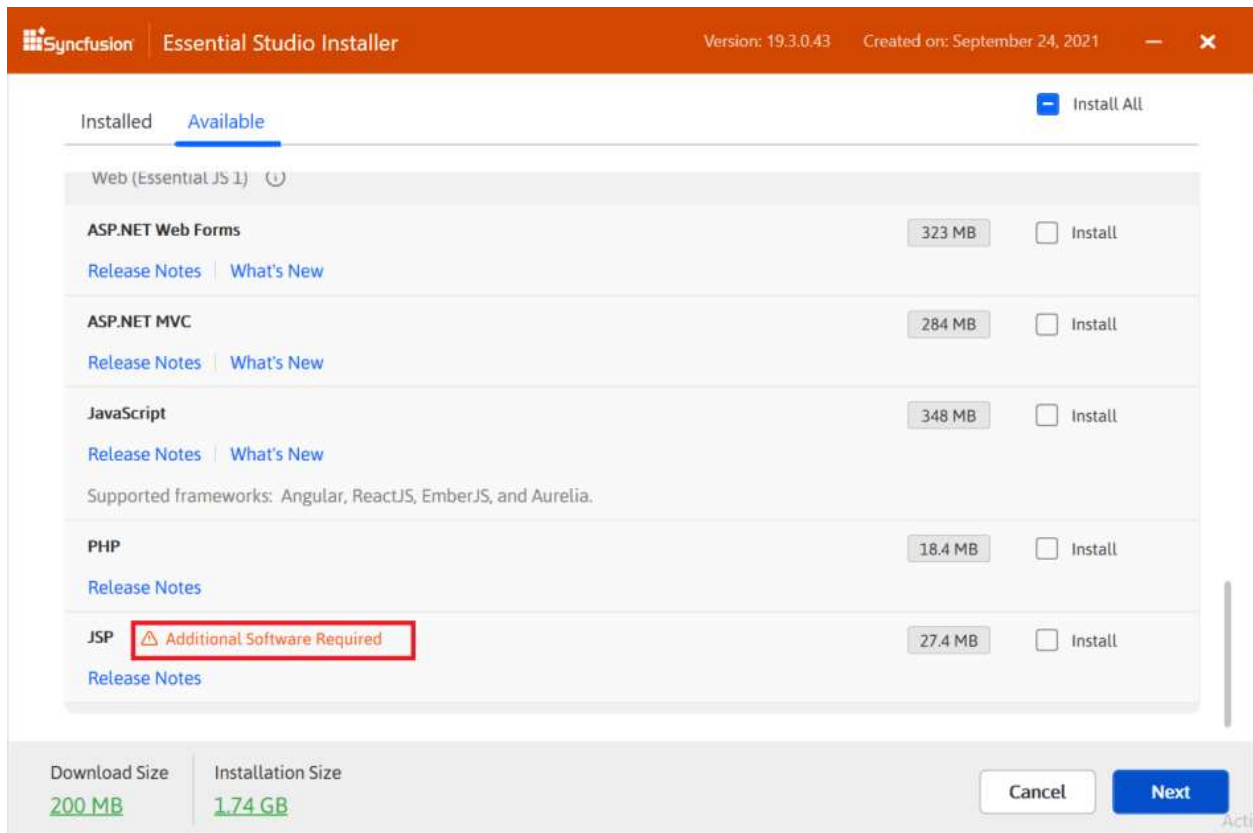




**Information:** If the required software for the selected product isn't already installed, the **Additional Software Required** alert will appear. You can, however, continue the installation and install the necessary software later.

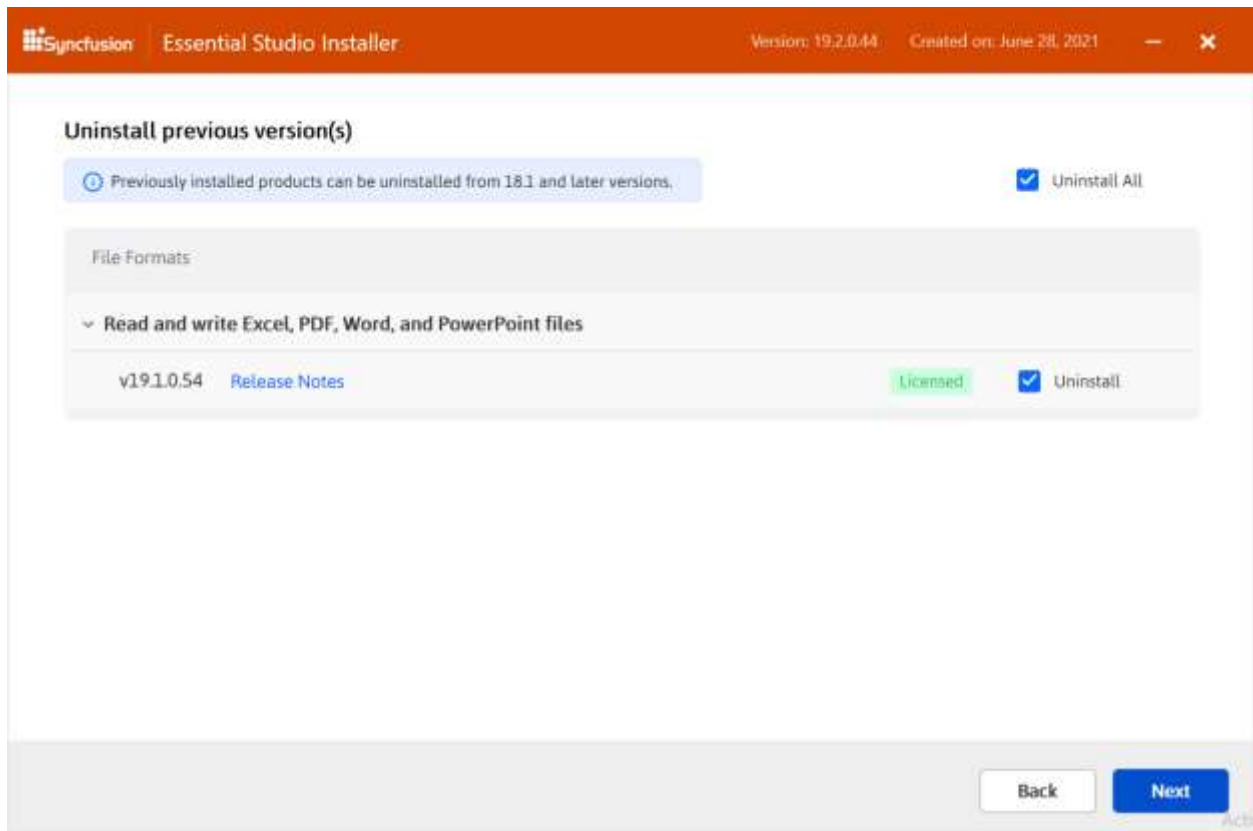
### Required Software





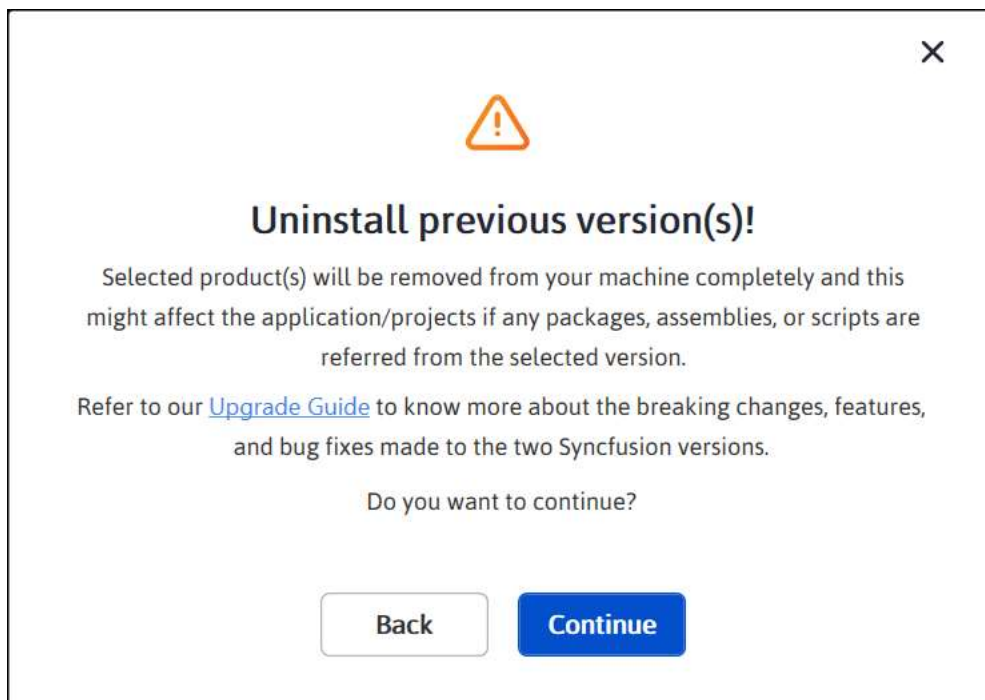
4. If previous version(s) for the selected products are installed, the Uninstall previous version wizard will be displayed. You can see the list of previously installed versions for the products you've chosen here. To remove all versions, check the **Uninstall All** checkbox. Click the Next button.





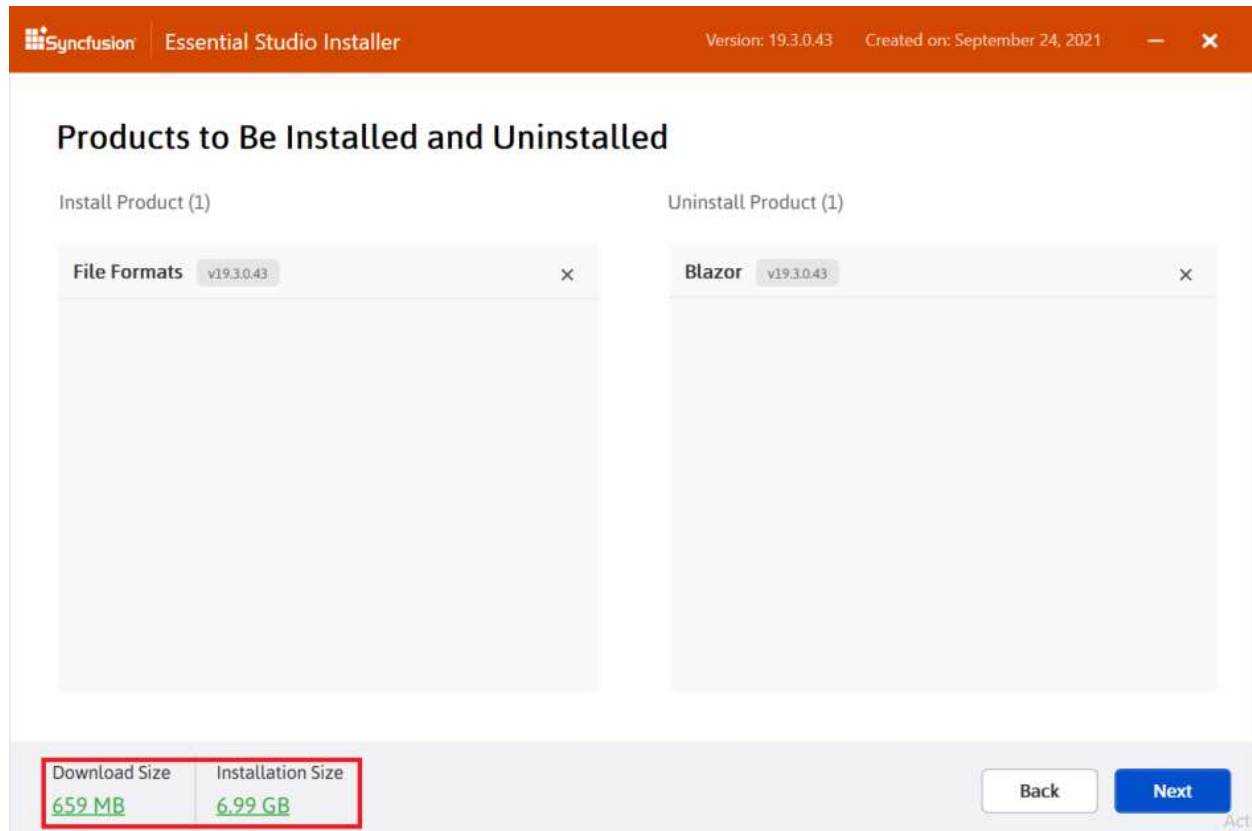
**Note:** From the 2021 Volume 1 release, Syncfusion has provided option to uninstall the previous versions from 18.1 while installing the new version.

5. Pop up screen will be displayed to get the confirmation to uninstall selected previous versions.





6. The Confirmation Wizard will appear with the list of products to be installed/uninstalled. You can view and modify the list of products that will be installed and uninstalled from this page.



**Note:** By clicking the **Download Size** and **Installation Size** links, you can determine the approximate size of the download and installation

7. The Configuration Wizard will appear. You can change the Download, Install, and Demos locations from here. You can also change the Additional settings on a product-by-product basis. Click Next to install with the default settings.



**Configuration**

**Download Location**  
 C:\ProgramData\Syncfusion\19.3.0.43\Downloads\ [Browse](#)

**Installation Location**  
 C:\Program Files (x86)\Syncfusion\Essential Studio\ [Browse](#)

**Demos Location**  
 C:\Users\Public\Documents\Syncfusion\ [Browse](#)

☒ I Agree to the [License Terms](#) and [Privacy Policy](#)

**Additional Settings**

- ☒ Install Demos [File Formats](#)
- ☒ Register Syncfusion assemblies in GAC [File Formats](#)
- ☒ Configure Syncfusion Controls in Visual Studio [File Formats](#)
- ☒ Create Desktop Shortcut(s) [File Formats](#)
- ☒ Create Start Menu Shortcut(s) [File Formats](#)

| Download Size | Installation Size |
|---------------|-------------------|
| 698.1 MB      | 6.99 GB           |

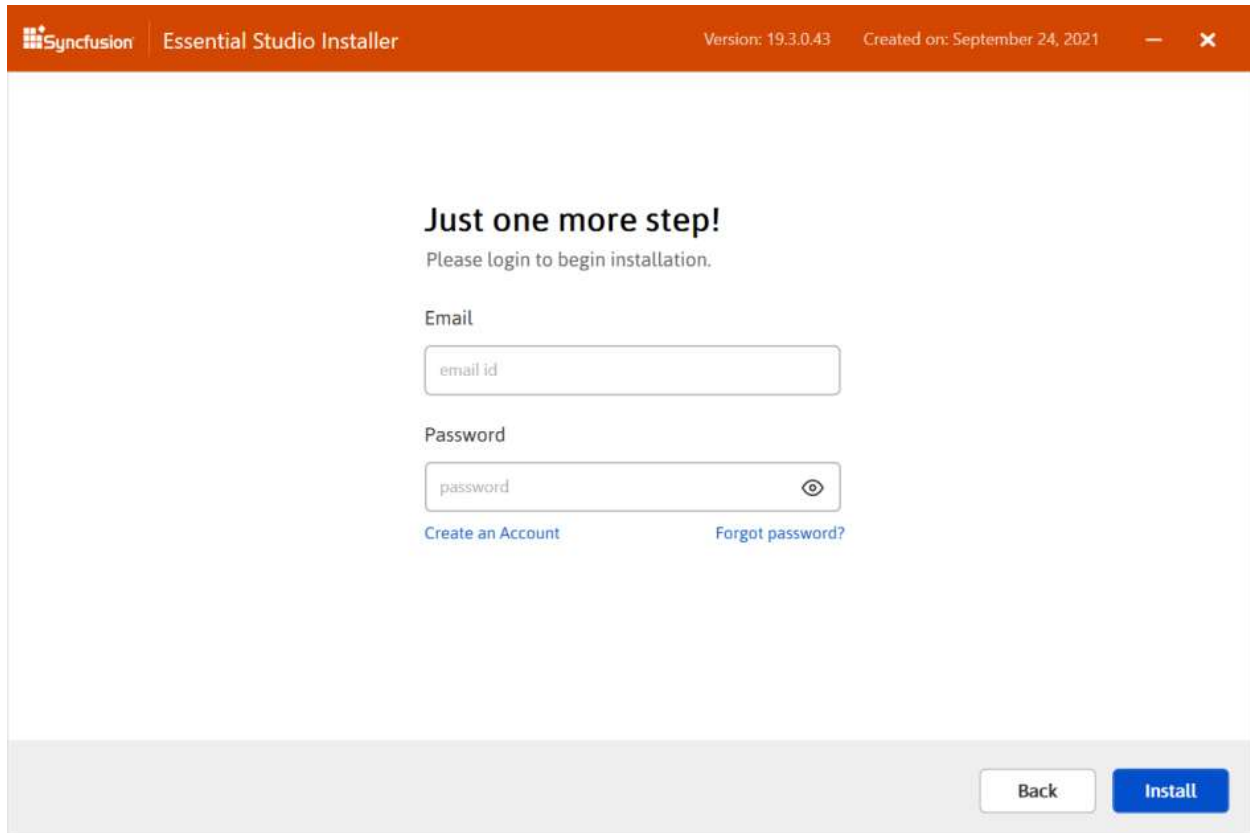
[Back](#) [Next](#)

### Additional settings

- Select the **Install Demos** check box to install Syncfusion samples, or leave the check box unchecked, if you do not want to install Syncfusion samples
  - Select the **Register Syncfusion Assemblies in GAC** check box to install the latest Syncfusion assemblies in GAC, or clear this check box when you do not want to install the latest assemblies in GAC.
  - Select the **Configure Syncfusion controls in Visual Studio** check box to configure the Syncfusion controls in the Visual Studio toolbox, or clear this check box when you do not want to configure the Syncfusion controls in the Visual Studio toolbox during installation. Note that you must also select the Register Syncfusion assemblies in GAC check box when you select this check box.
  - Select the **Configure Syncfusion Extensions controls in Visual Studio** checkbox to configure the Syncfusion Extensions in Visual Studio or clear this check box when you do not want to configure the Syncfusion Extensions in Visual Studio.
  - Check the **Create Desktop Shortcut** checkbox to add a desktop shortcut for Syncfusion Control Panel
  - Check the **Create Start Menu Shortcut** checkbox to add a shortcut to the start menu for Syncfusion Control Panel
8. After reading the License Terms and Conditions, check the **I agree to the License Terms and Privacy Policy** check box. Click the Next button.
  9. The login wizard will appear. You must enter your Syncfusion email address and password. If you do not already have a Syncfusion account, you can create one by clicking on **Create an Account**.



If you have forgotten your password, click **Forgot Password** to create a new one. Click the Install button.

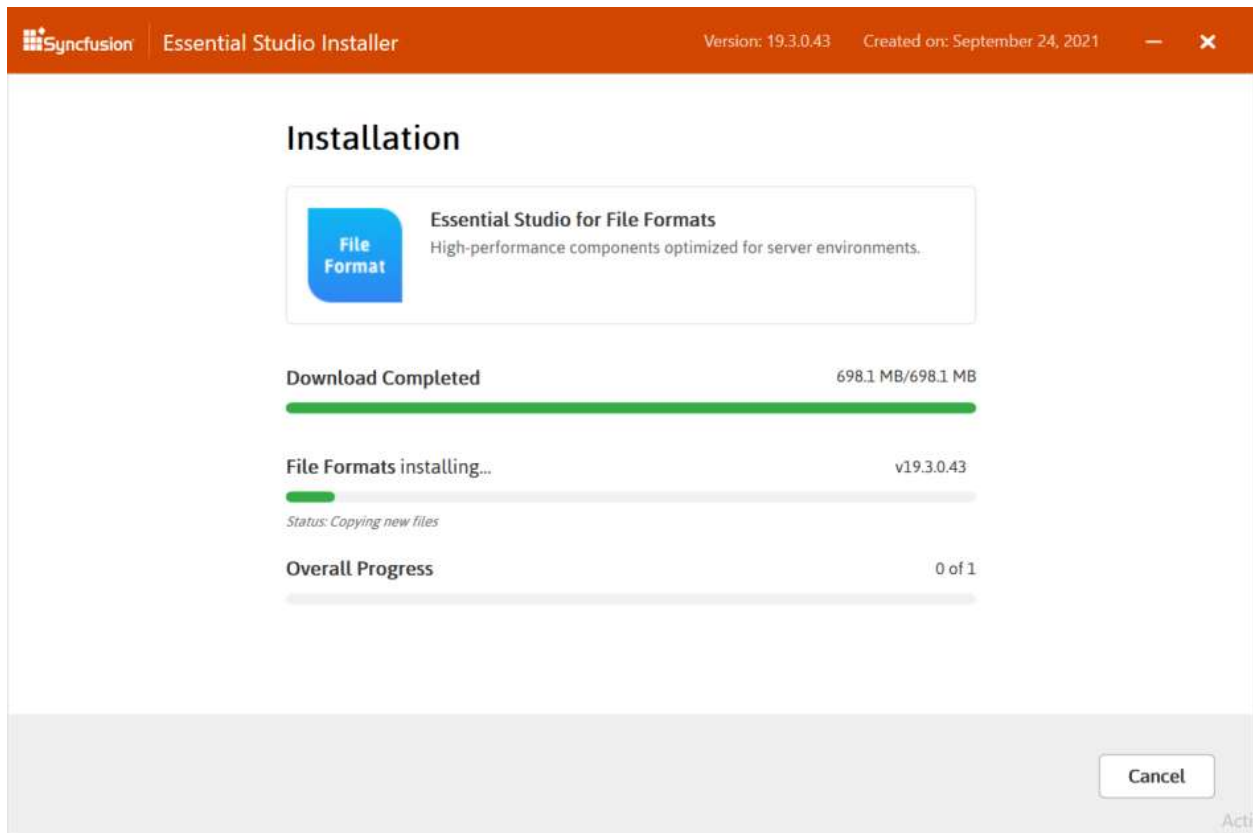


The screenshot shows the 'Syncfusion Essential Studio Installer' window. The title bar is orange and contains the Syncfusion logo, the text 'Essential Studio Installer', the version 'Version: 19.3.0.43', and the creation date 'Created on: September 24, 2021'. The main content area has a white background with the heading 'Just one more step!' and the instruction 'Please login to begin installation.' Below this are two input fields: 'Email' with a placeholder 'email id' and 'Password' with a placeholder 'password' and a toggle icon. At the bottom of the input fields are two links: 'Create an Account' and 'Forgot password?'. At the bottom right of the window are two buttons: 'Back' and 'Install'.

**Information:** The products you have chosen will be installed based on your Syncfusion License (Trial or Licensed).

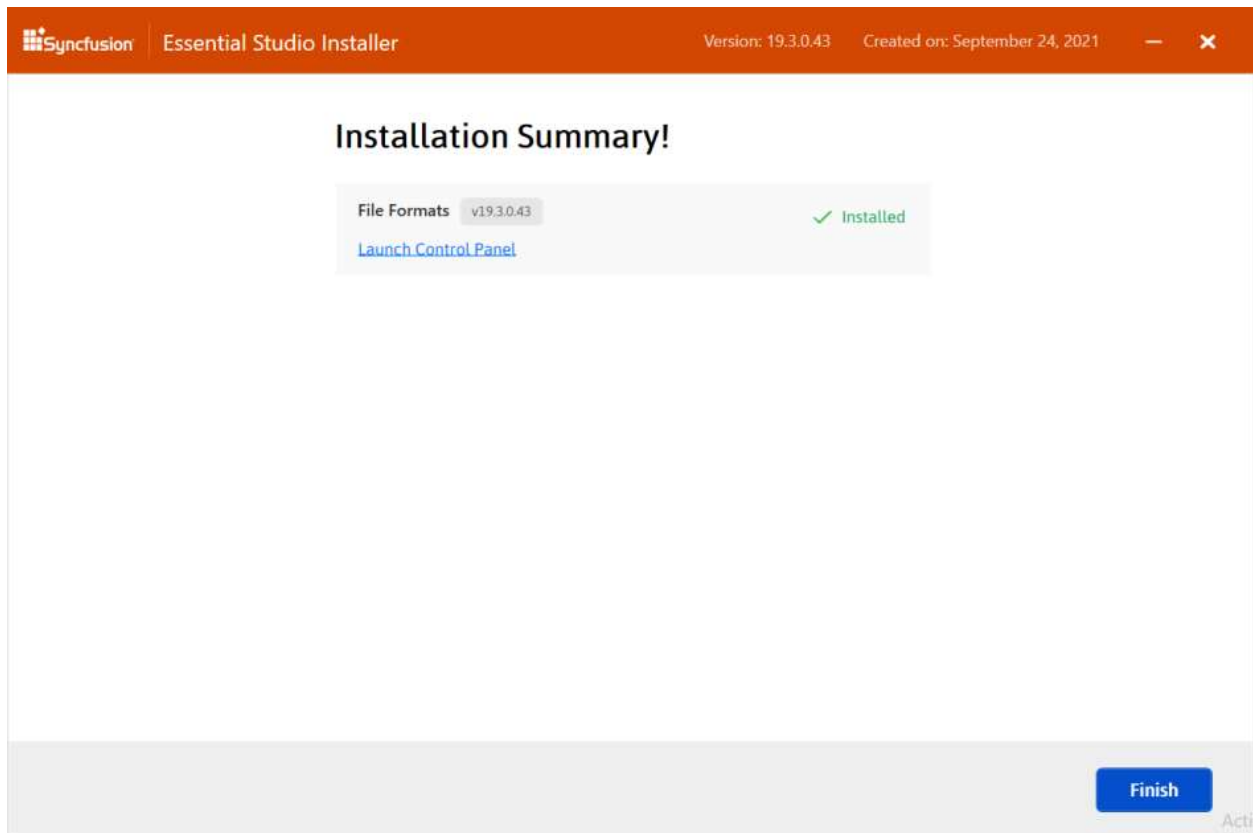
10. The download and installation\uninstallation progress will be displayed as shown below.





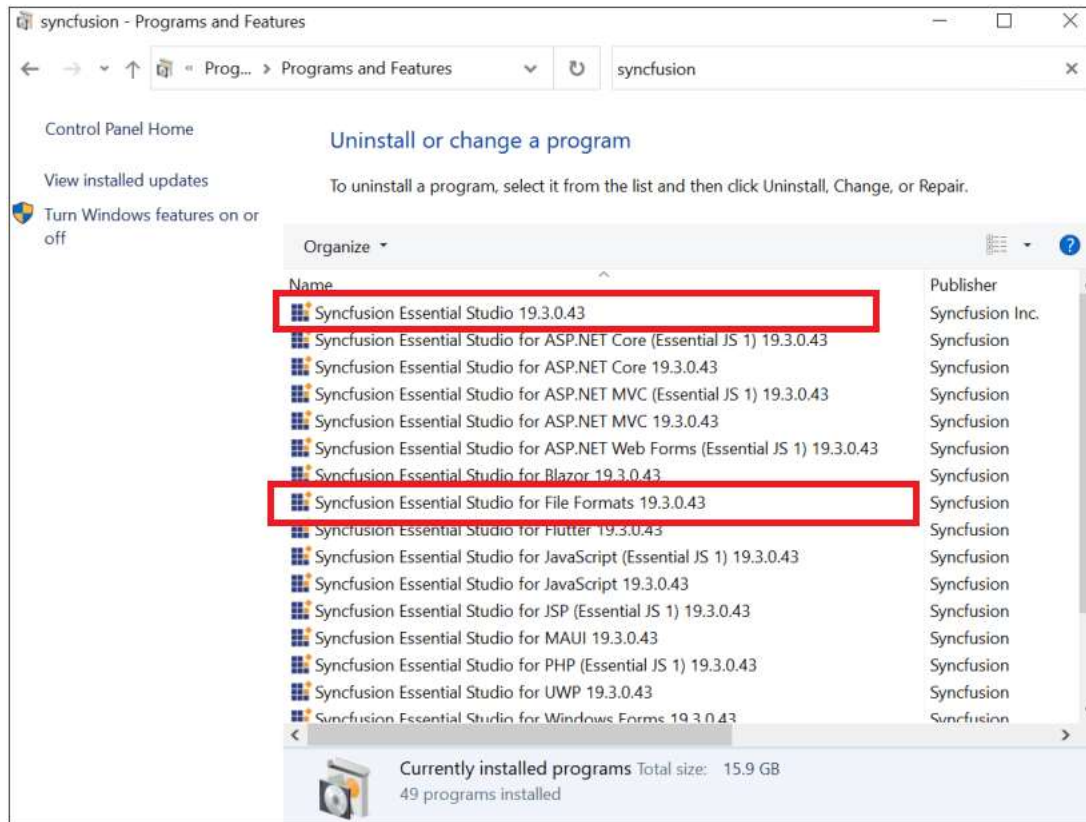
11. When the installation is finished, the **Summary** wizard will appear. Here you can see the list of products that have been installed successfully and those that have failed. To close the Summary wizard, click Finish.





- To open the Syncfusion Control Panel, click **Launch Control Panel**.
  12. After installation, there will be two Syncfusion control panel entries, as shown below. The Essential Studio entry will manage all Syncfusion products installed in the same version, while the Product entry will only uninstall the specific product setup.





**Note:** \* You can find the required jars in the FileFormats installed location

- **Location:** {ProgramFilesFolder}\Syncfusion\Essential Studio\ {Platform}\ {version}\JarFiles
- **Example:** C:\Program Files (x86)\Syncfusion\Essential Studio\FileFormats\19.1.0.54\JarFiles

## Uninstallation

Syncfusion FileFormats installer can be uninstalled in two ways.

- Uninstall the FileFormats using the Syncfusion FileFormats web installer
- Uninstall the FileFormats from Windows Control Panel

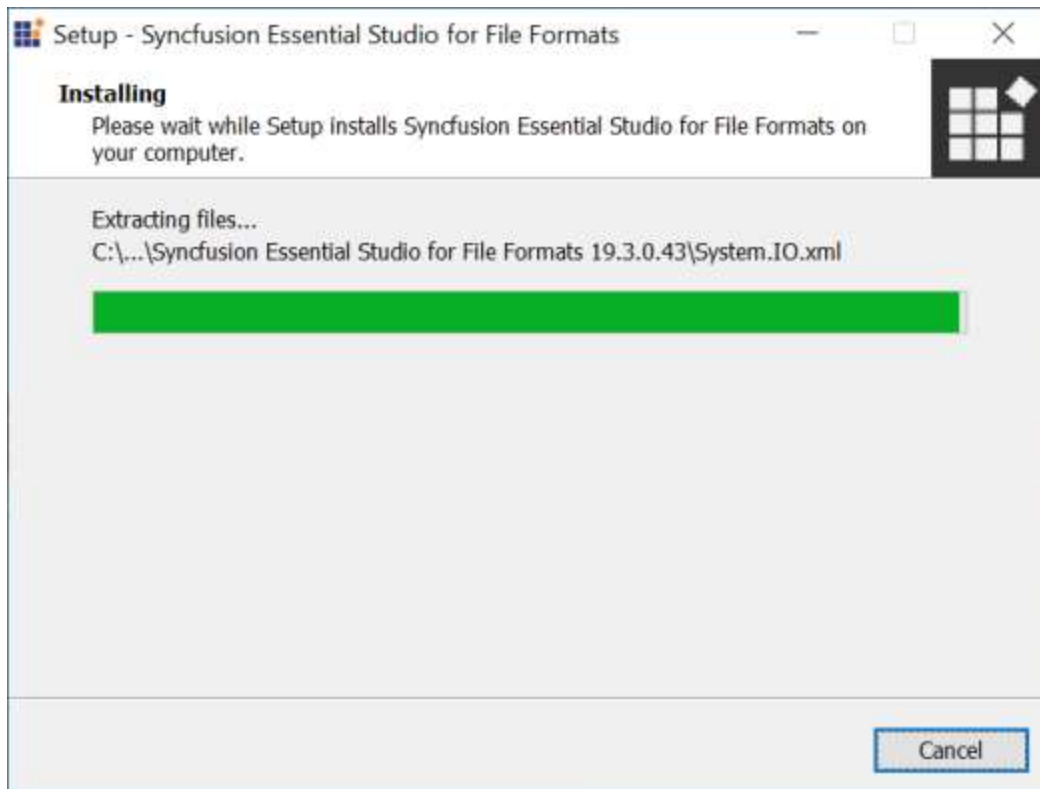
Follow either one of the option below to uninstall Syncfusion Essential Studio FileFormats installer.

### Option 1: Uninstall the FileFormats using the Syncfusion FileFormats web installer

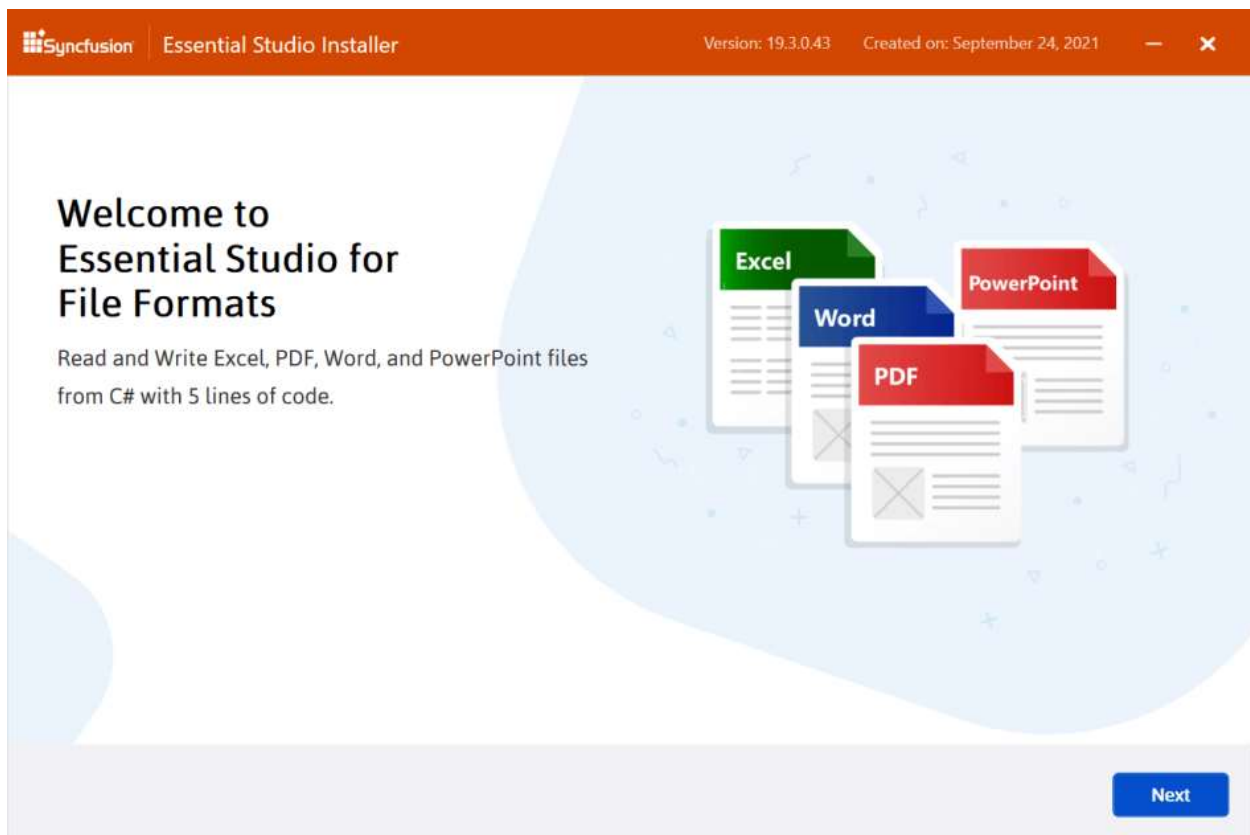
Syncfusion provides the option to uninstall products of the same version directly from the Web Installer application. Select the products to be uninstalled from the list, and Web Installer will uninstall them one by one.

Open the Syncfusion Essential Studio FileFormats Online Installer file from downloaded location by double-clicking it. The Installer Wizard automatically opens and extracts the package





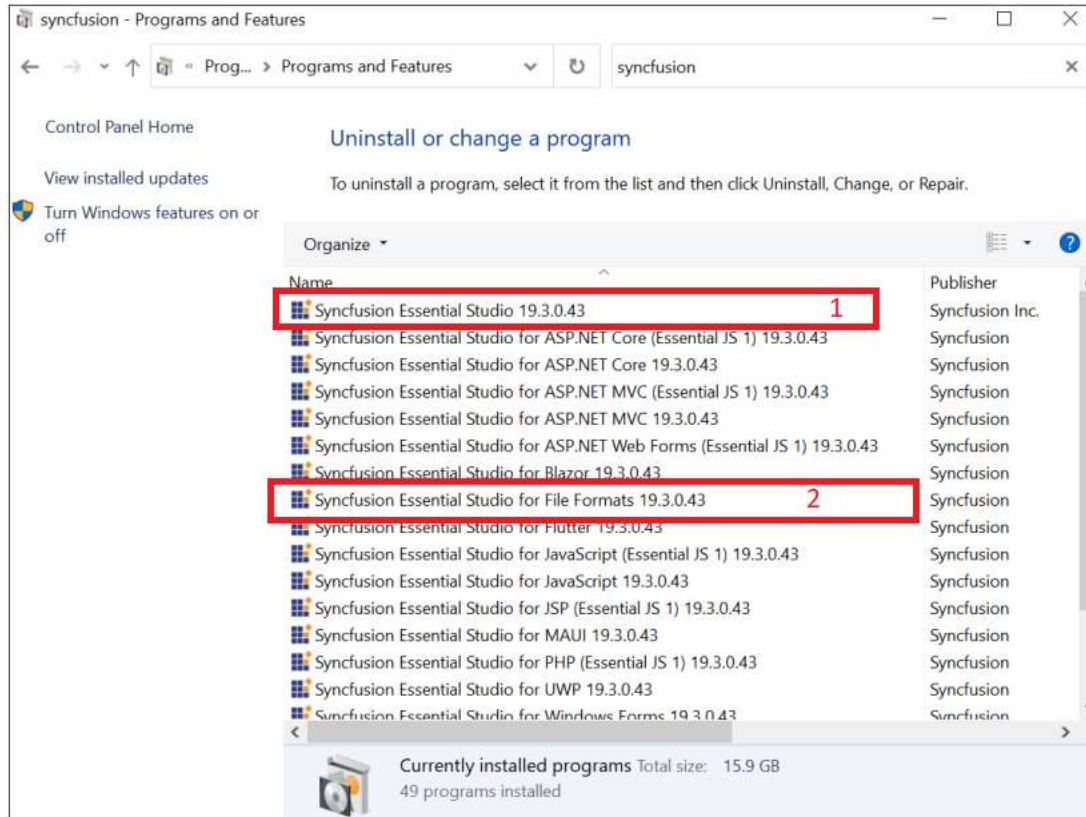
The Syncfusion FileFormats Web Installer's welcome wizard will be displayed. Click the Next button





## Option 2: Uninstall the FileFormats from Windows Control Panel

You can uninstall all the installed products by selecting the **Syncfusion Essential Studio {version}** entry (element 1 in the below screenshot) from the Windows control panel, or you can uninstall FileFormats alone by selecting the **Syncfusion Essential Studio for FileFormats {version}** entry (element 2 in the below screenshot) from the Windows control panel.

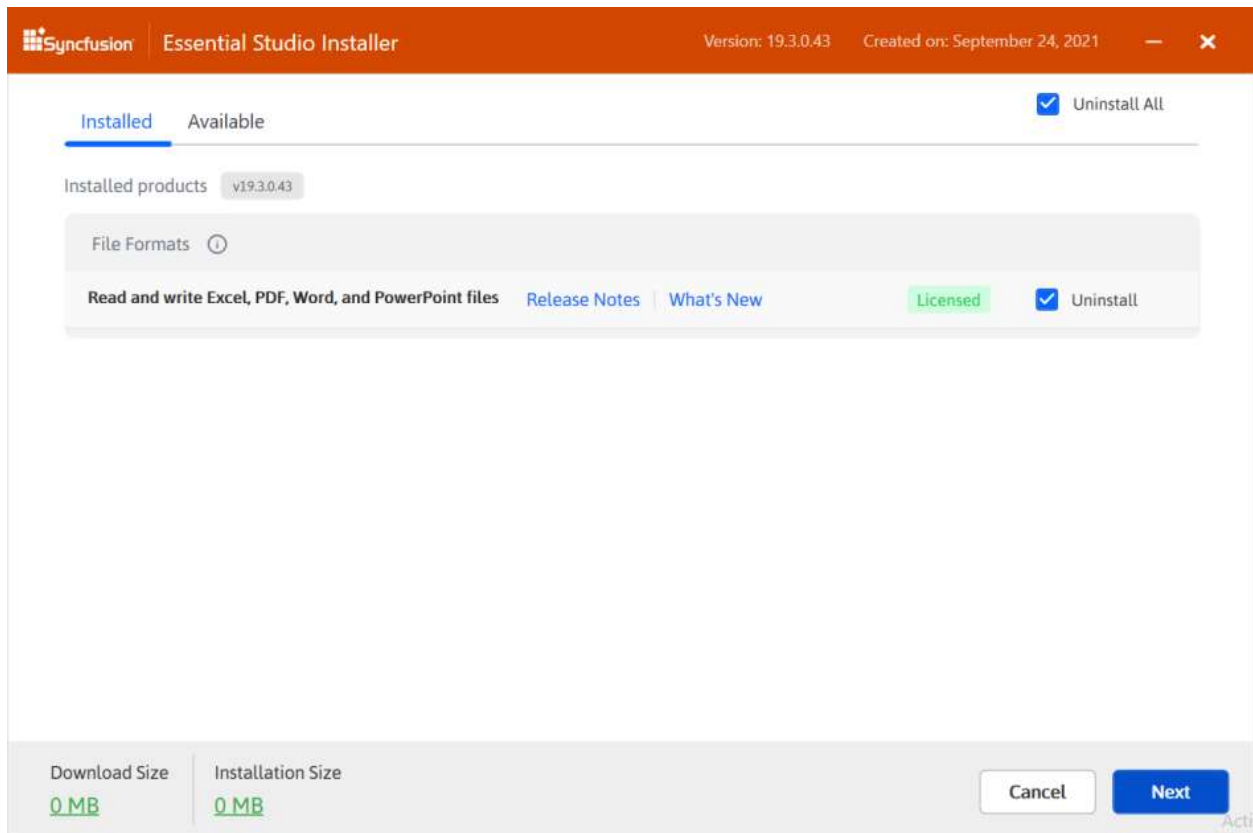


**Note:** If the **SynCFusion Essential Studio for FileFormats {version}** entry is selected from the Windows control panel, the SynCFusion Essential Studio FileFormats alone will be removed and the below default MSI uninstallation window will be displayed.

1. The Platform Selection Wizard will appear. From the **Installed** tab, select the products to be uninstalled. To select all products, check the **Uninstall All** checkbox. Click the Next button.

<em>Installed</em>

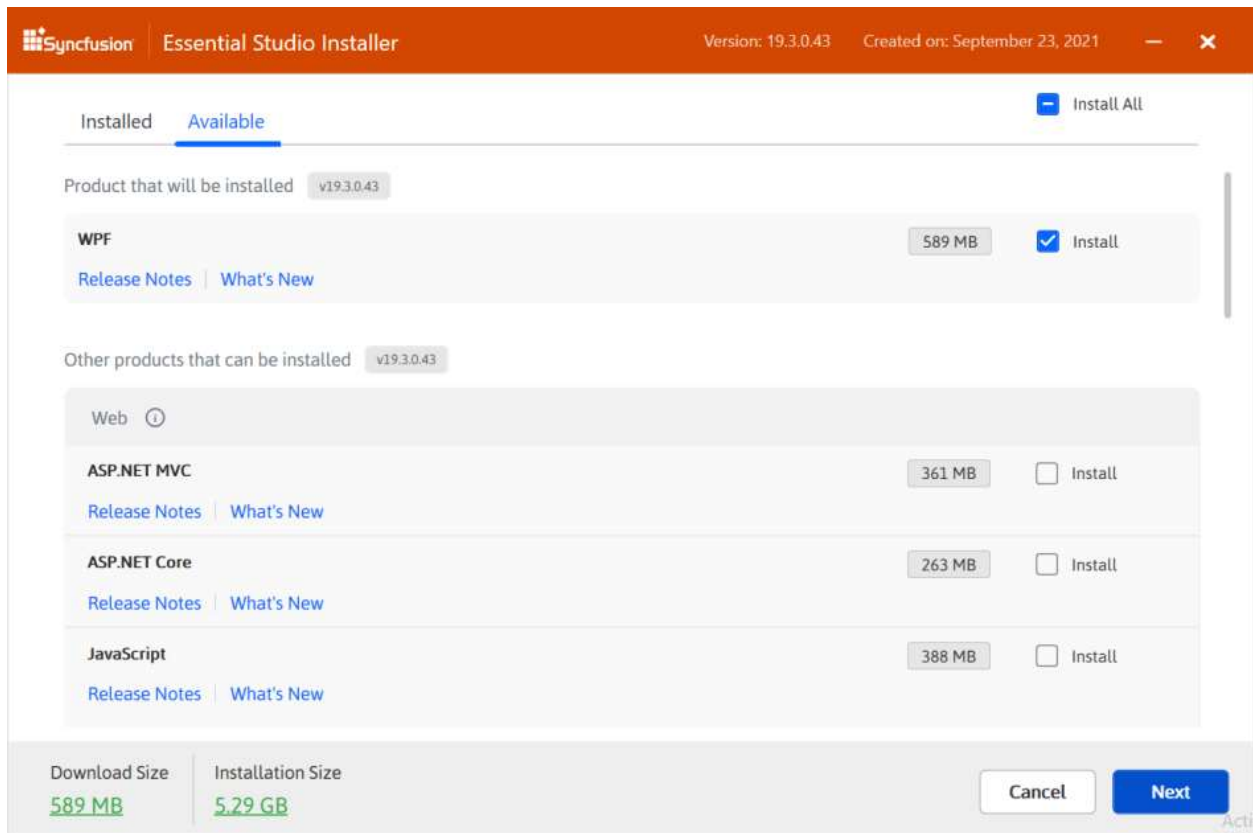




You can also select the products to be installed from the **Available** tab. Click the Next button.

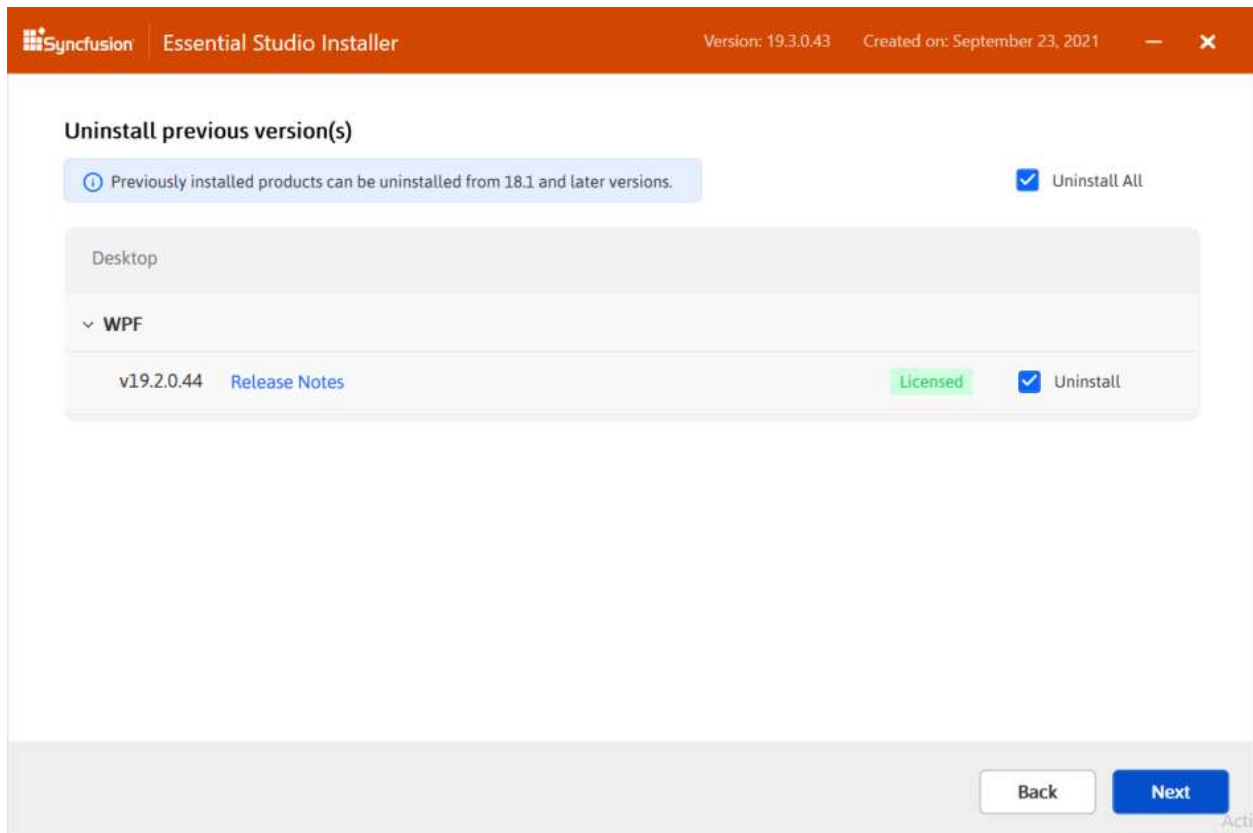
<em>Available</em>



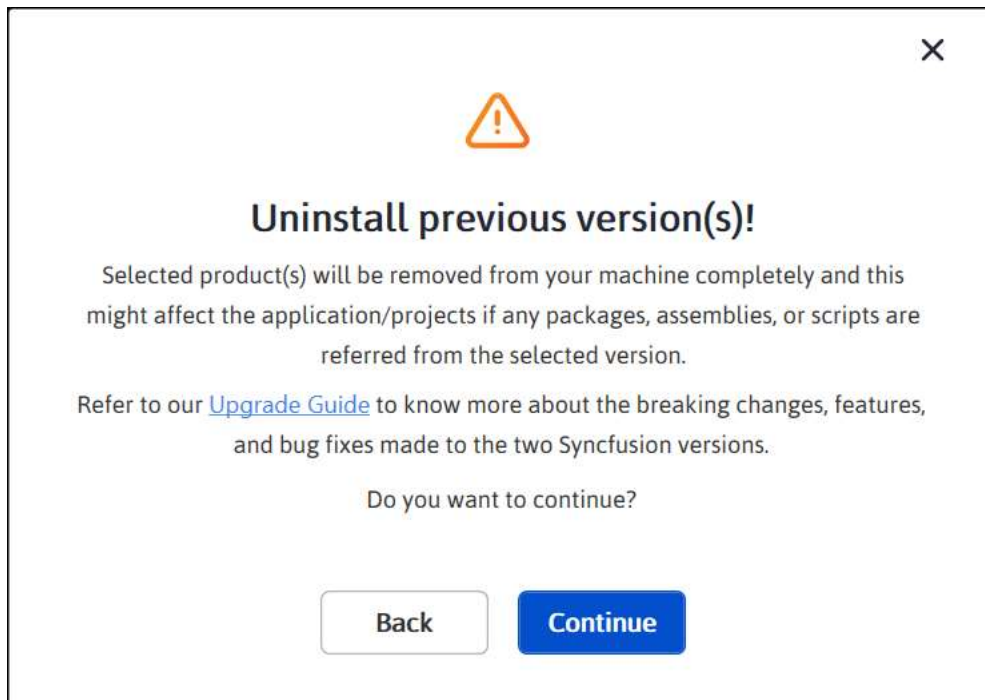


2. If any other products selected for installation, Uninstall previous version wizard will be displayed with previous version(s) installed for the selected products. Here you can view the list of installed previous versions for the selected products. Select **Uninstall All** checkbox to select all the versions. Click Next.



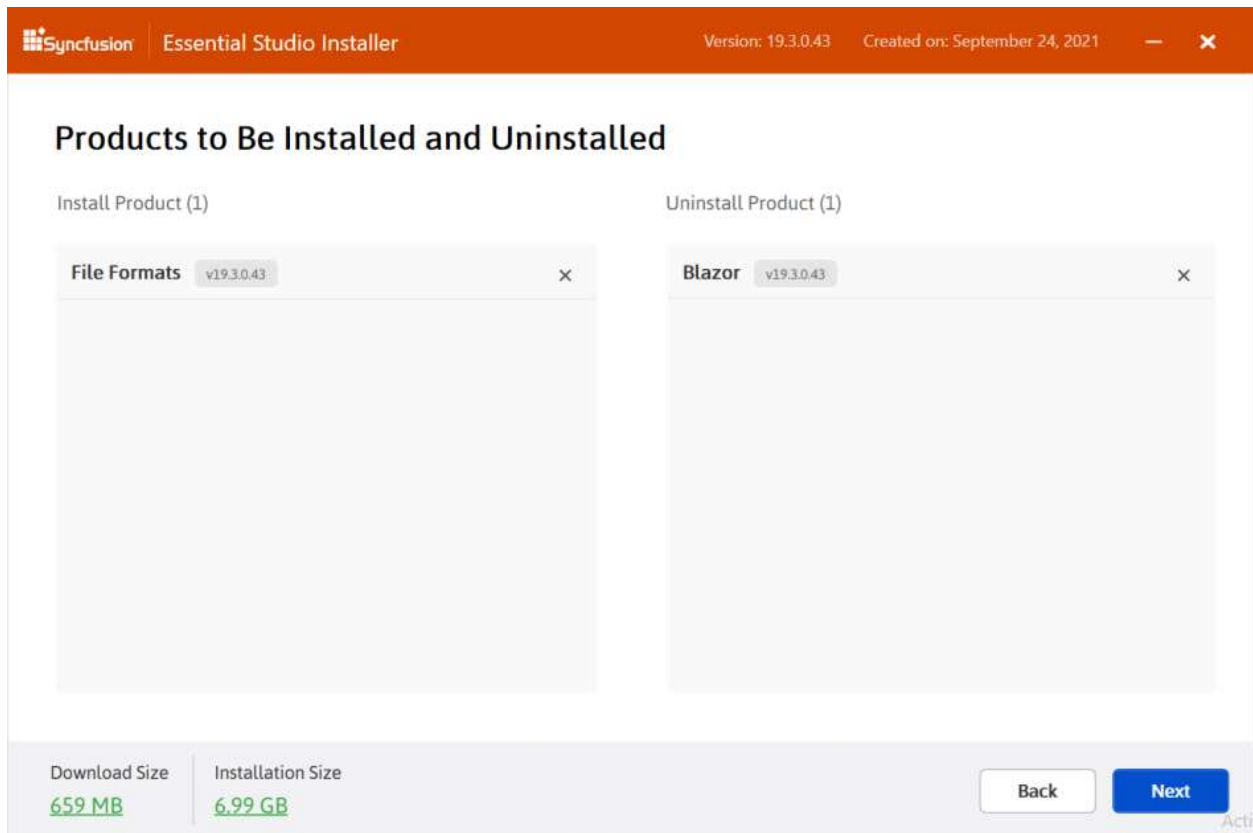


3. Pop up screen will be displayed to get the confirmation to uninstall selected previous versions.



4. The Confirmation Wizard will appear with the list of products to be installed/uninstalled. Here you can view and modify the list of products that will be installed/uninstalled.





**Note:** By clicking the **Download Size** and **Installation Size** links, you can determine the approximate size of the download and installation

5. The Configuration Wizard will appear. You can change the Download, Install, and Demos locations from here. You can also change the Additional settings on a product-by-product basis. Click Next to install with the default settings.

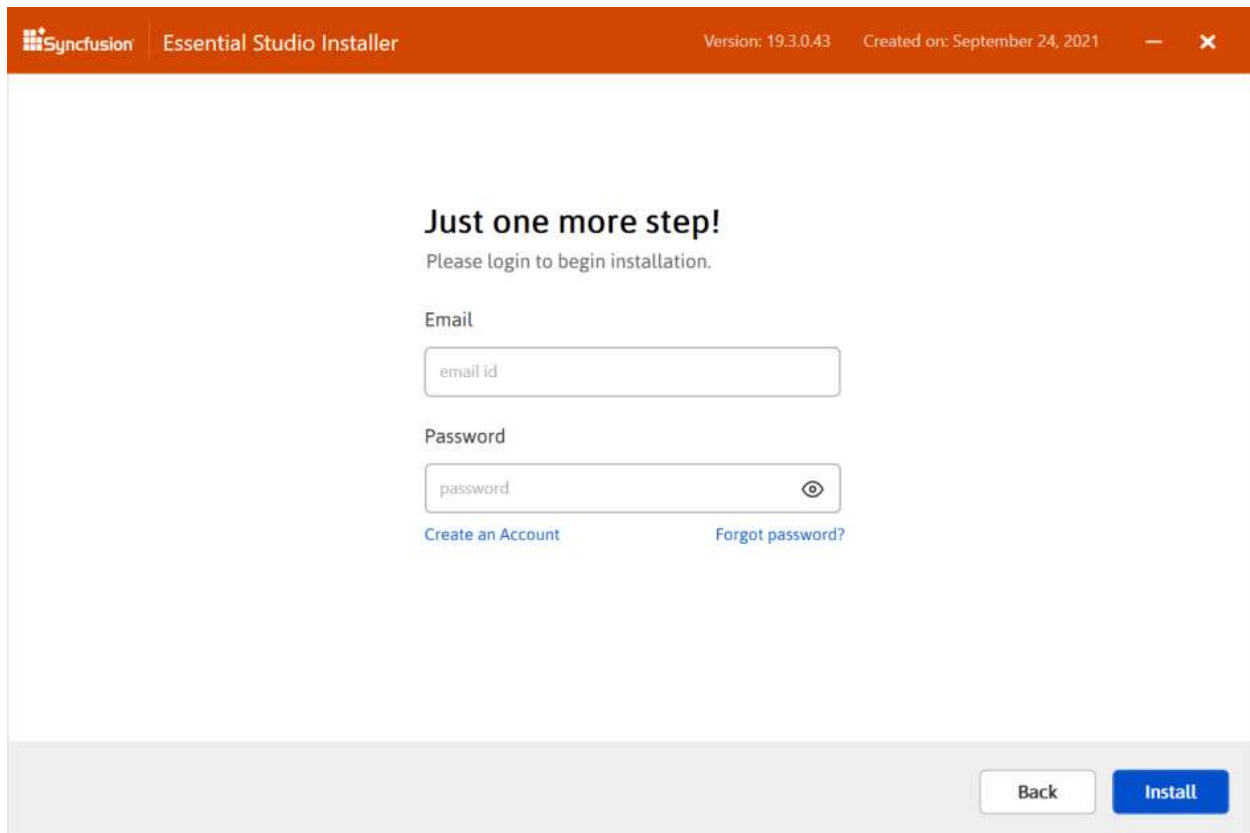


The screenshot shows the 'Configuration' window of the Syncfusion Essential Studio Installer. The window has an orange header bar with the Syncfusion logo, the title 'Essential Studio Installer', the version '19.3.0.43', and the creation date 'September 23, 2021'. The main content area is divided into two columns. The left column contains three sections: 'Download Location' with a text box showing 'C:\ProgramData\Syncfusion\19.3.0.43\Downloads\' and a 'Browse' button; 'Installation Location' with a text box showing 'C:\Program Files (x86)\Syncfusion\Essential Studio\' and a 'Browse' button; and 'Demos Location' with a text box showing 'C:\Users\Public\Documents\Syncfusion\' and a 'Browse' button. Below these is a checkbox labeled 'I Agree to the License Terms and Privacy Policy' which is checked. The right column is titled 'Additional Settings' and contains a list of options, each with a checked checkbox and a 'WPF' sub-option: 'Install Demos', 'Register Syncfusion assemblies in GAC', 'Configure Syncfusion Controls in Visual Studio', 'Configure Syncfusion Extensions in Visual Studio', 'Create Desktop Shortcut(s)', and 'Create Start Menu Shortcut(s)'. At the bottom, there is a table showing 'Download Size' as '597.2 MB' and 'Installation Size' as '5.29 GB'. To the right of the table are 'Back' and 'Next' buttons. The 'Next' button is highlighted in blue.

| Download Size | Installation Size |
|---------------|-------------------|
| 597.2 MB      | 5.29 GB           |

6. After reading the License Terms and Conditions, check the **I agree to the License Terms and Privacy Policy** check box. Click the Next button. 7. The login wizard will appear. You must enter your Syncfusion email address and password. If you do not already have a Syncfusion account, you can create one by clicking on **Create an Account**. If you have forgotten your password, click **Forgot Password** to create a new one. Click the Install button.





The screenshot shows the 'Essential Studio Installer' window. The title bar is orange and contains the Syncfusion logo, the text 'Essential Studio Installer', and version/creation information: 'Version: 19.3.0.43 Created on: September 24, 2021'. The main content area has a white background with the heading 'Just one more step!' and the instruction 'Please login to begin installation.' Below this are two input fields: 'Email' with a placeholder 'email id' and 'Password' with a placeholder 'password' and a toggle icon. At the bottom of the form are two links: 'Create an Account' and 'Forgot password?'. At the bottom right of the window are two buttons: 'Back' and 'Install'.

Syncfusion Essential Studio Installer

Version: 19.3.0.43 Created on: September 24, 2021

### Just one more step!

Please login to begin installation.

Email

Password

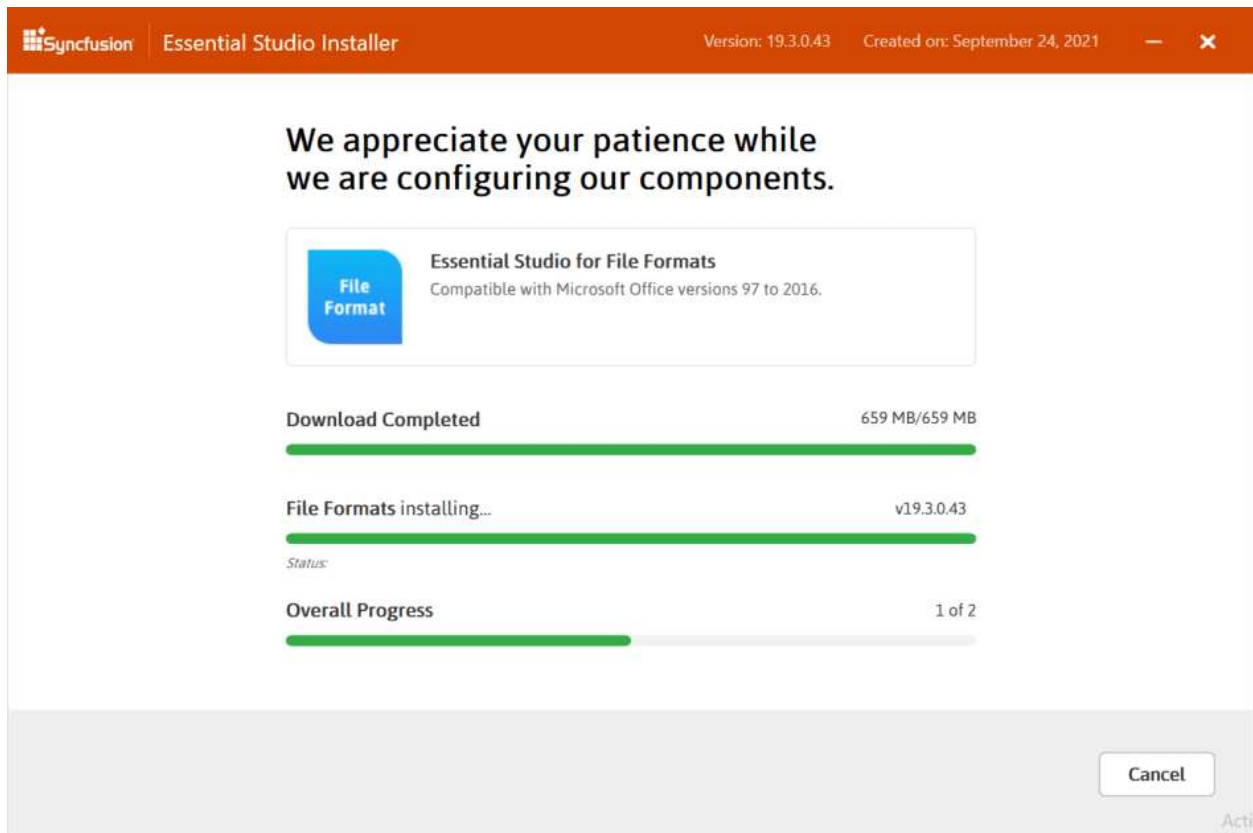
[Create an Account](#) [Forgot password?](#)

[Back](#) [Install](#)

**Information:** The products you have chosen will be installed based on your Syncfusion License (Trial or Licensed).

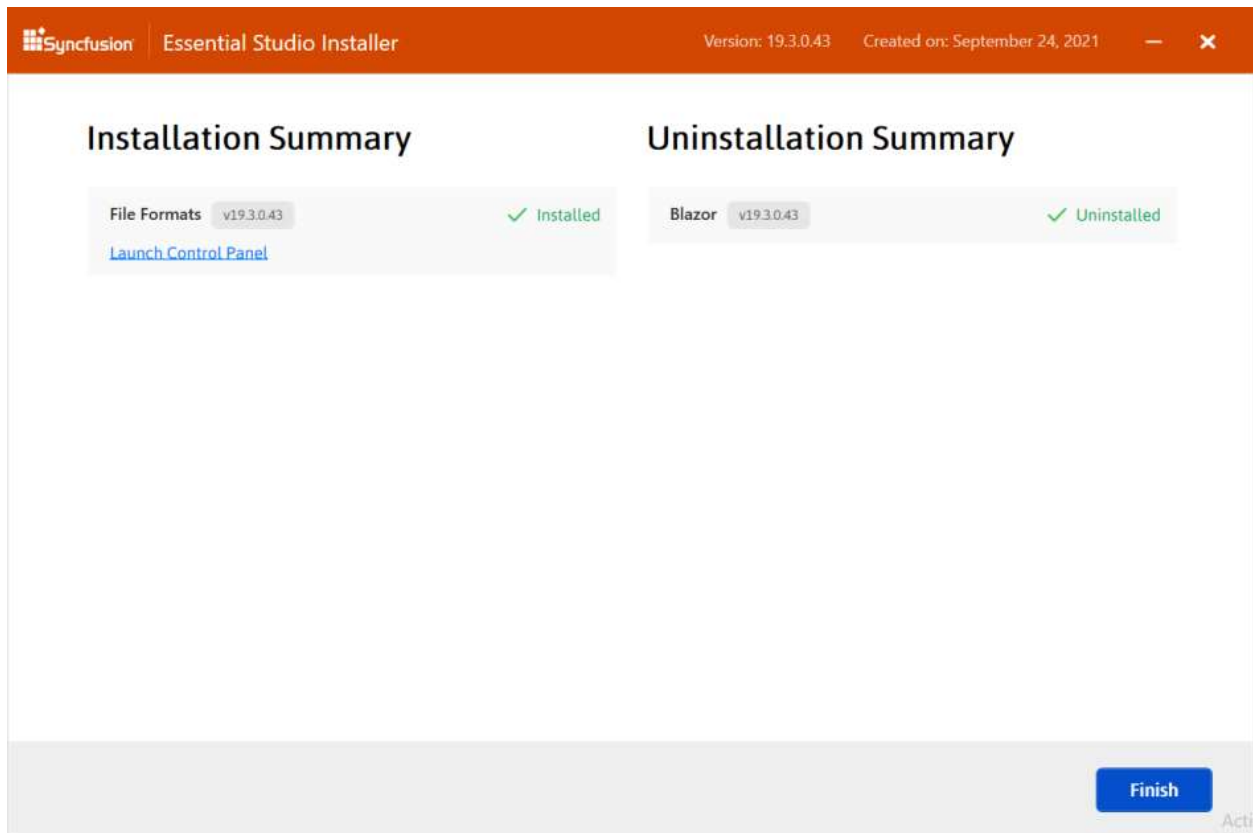
8. The download, installation, and uninstallation progresses will be shown.





9. When the installation is finished, the **Summary** wizard will appear. Here you can see the list of products that have been successfully and unsuccessfully installed/uninstalled. To close the Summary wizard, click Finish.





- To open the Syncfusion Control Panel, click **Launch Control Panel**.

### Installation using Offline Installer

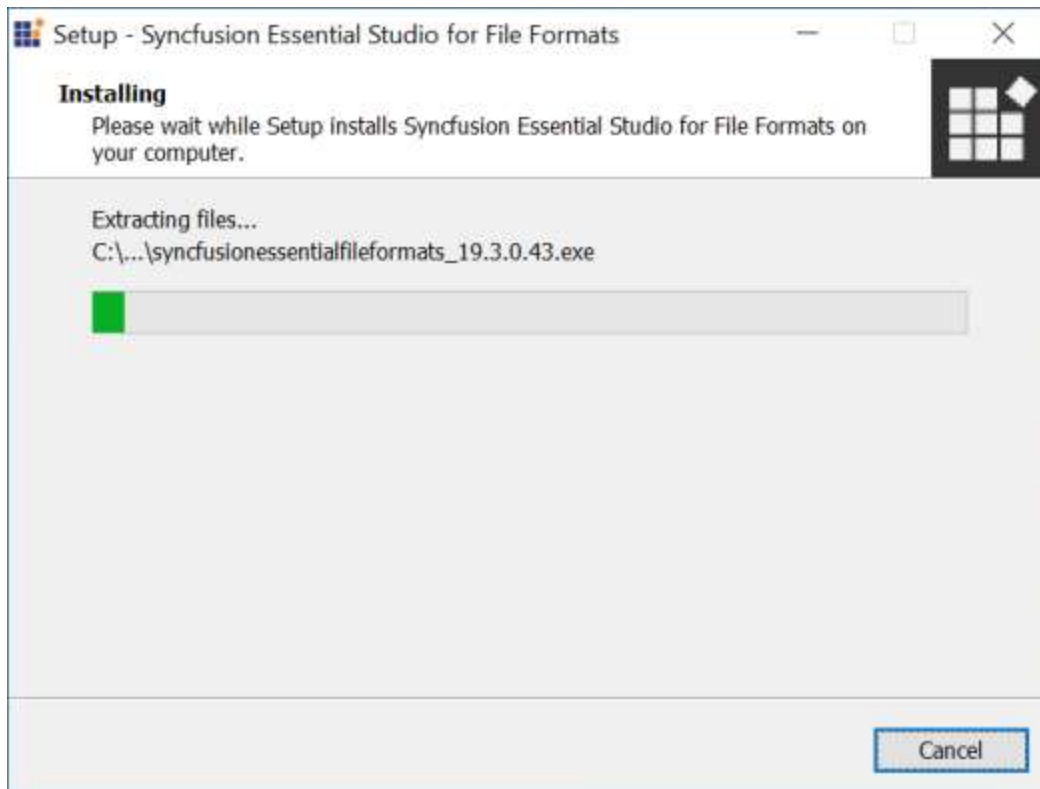
You can refer to the [Download](#) section to learn how to get the FileFormats trial or licensed installer.

### Installing with UI

The steps below show how to install the Essential Studio FileFormats installer.

1. Open the Syncfusion FileFormats offline installer file from downloaded location by double-clicking it. The Installer Wizard automatically opens and extracts the package





**Note:** The Installer wizard extracts the syncfusionessentialfileformats\_(version).exe dialog, which displays the package's unzip operation.

2. To unlock the Syncfusion offline installer, you have two options:

- *Login To Install*
- *Use Unlock Key*

### Login To Install

You must enter your Syncfusion email address and password. If you don't already have a Syncfusion account, you can sign up for one by clicking **"Create an account"**. If you have forgotten your password, click on **"Forgot Password"** to create a new one. Once you've entered your Syncfusion email and password, click Next.



**Syncfusion**

## Essential Studio for File Formats

Version: 19.3.0.43, Date: September 24, 2021

Thank you for choosing Syncfusion Essential Studio. Please enter the required information to proceed further.

**Login To Install** **Use Unlock Key**

Email \*

Password \*

[Forgot Password?](#)

Don't have an account? [Create an account](#)

☐ I agree to the [License Terms](#) and [Privacy Policy](#)

Next

### Use Unlock Key

Unlock keys are used to unlock the Syncfusion offline installer, and they are platform and version specific. You should use either Syncfusion licensed or trial Unlock key to unlock Syncfusion FileFormats installer.

The trial unlock key is only valid for 30 days, and the installer will not accept an expired trial key.

To learn how to generate an unlock key for both trial and licensed products, see [this](#) Knowledge Base article.



Syncfusion

## Essential Studio for File Formats

Version: 19.3.0.43, Date: September 24, 2021

Thank you for choosing Syncfusion Essential Studio. Please enter the required information to proceed further.

[Login To Install](#) **Use Unlock Key**

If you do not have an unlock key, [request one from Syncfusion](#).

Unlock Key \*

[Free Evaluation Key](#)

☐ I agree to the [License Terms](#) and [Privacy Policy](#)

Next

3. After reading the License Terms and Privacy Policy, check the **“I agree to the License Terms and Privacy Policy”** check box. Click the Next button.

4. Change the install and sample locations here. You can also change the Additional settings. Click Next\Install to install with the default settings.





### Additional Settings

- Select the **Install Demos** check box to install Syncfusion samples, or leave the check box unchecked, if you do not want to install Syncfusion samples
- Select the **Register Syncfusion Assemblies in GAC** check box to install the latest Syncfusion assemblies in GAC, or clear this check box when you do not want to install the latest assemblies in GAC.
- Select the **Configure Syncfusion controls in Visual Studio** check box to configure the Syncfusion controls in the Visual Studio toolbox, or clear this check box when you do not want to configure

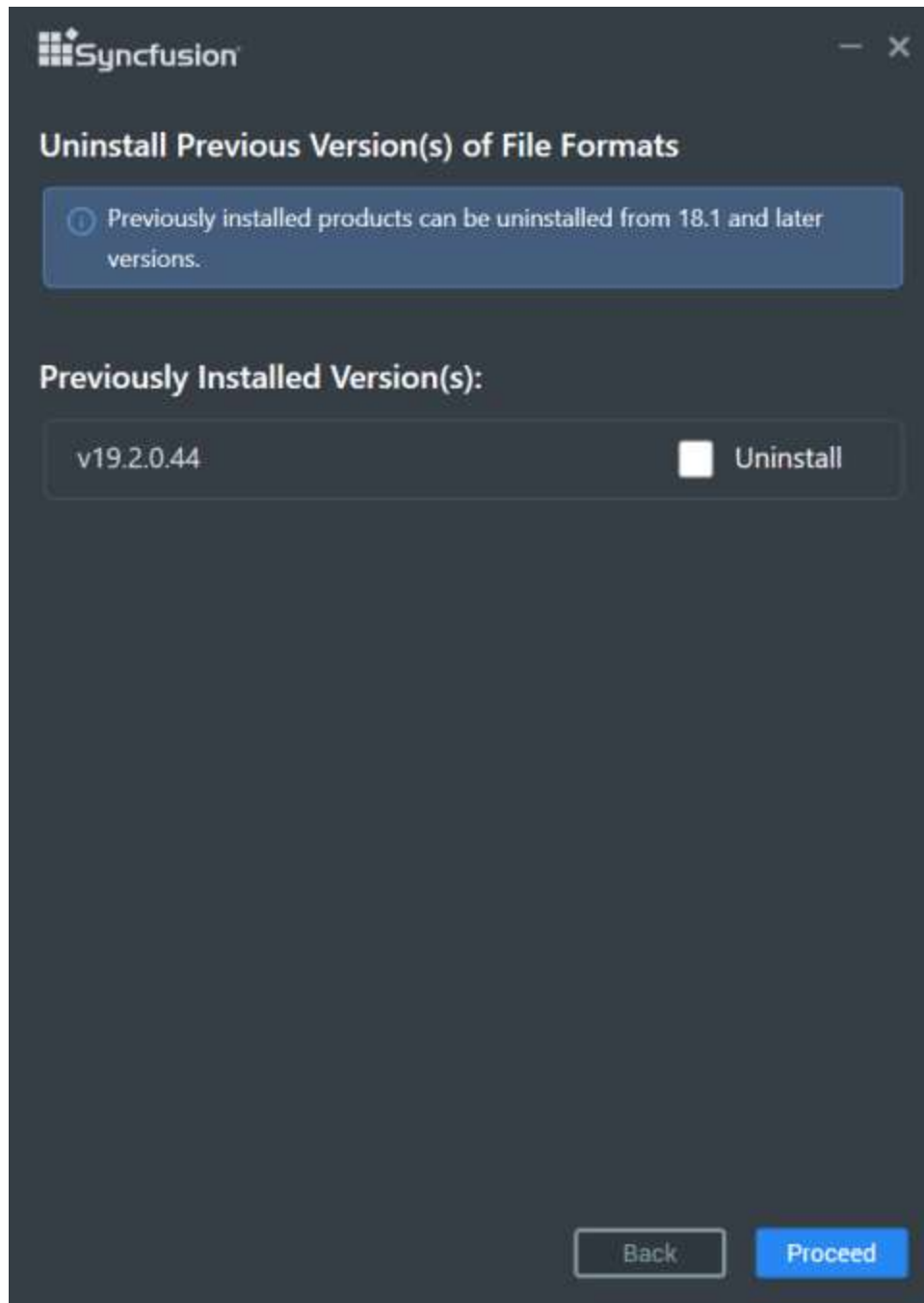


the Syncfusion controls in the Visual Studio toolbox during installation. Note that you must also select the Register Syncfusion assemblies in GAC check box when you select this check box.

- Select the **Configure Syncfusion Extensions controls in Visual Studio** checkbox to configure the Syncfusion Extensions in Visual Studio or clear this check box when you do not want to configure the Syncfusion Extensions in Visual Studio.
- Check the **Create Desktop Shortcut** checkbox to add a desktop shortcut for Syncfusion Control Panel
- Check the **Create Start Menu Shortcut** checkbox to add a shortcut to the start menu for Syncfusion Control Panel

5. If any previous versions of the current product is installed, the Uninstall Previous Version(s) wizard will be opened. Select **Uninstall** checkbox to uninstall the previous versions and then click the Proceed button.



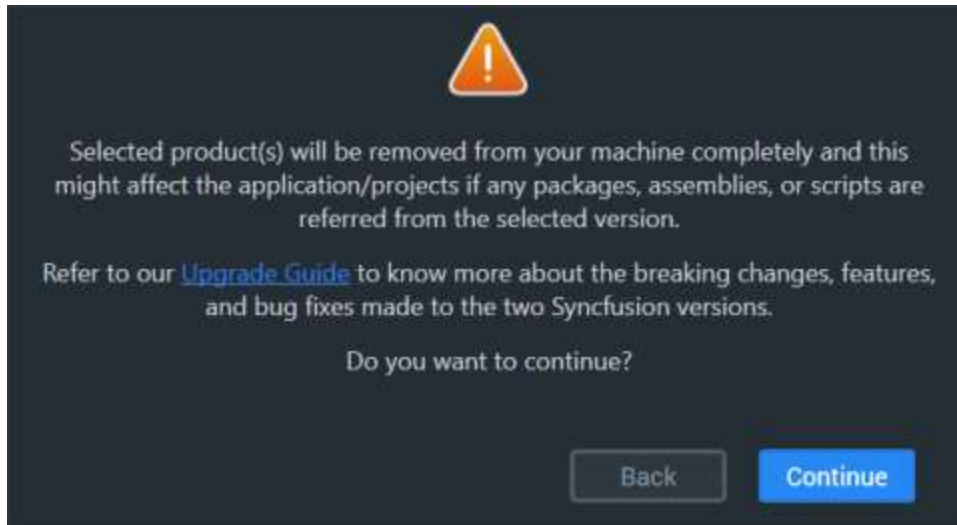


**Note:** From the 2021 Volume 1 release, Syncfusion has added the option to uninstall previous versions from 18.1 while installing the new version.

**Note:** If any version is selected to uninstall, a confirmation screen will appear; if continue is selected, the Progress screen will display the uninstall and install progress, respectively. If none of the versions are chosen to be uninstalled, only the installation progress will be displayed.

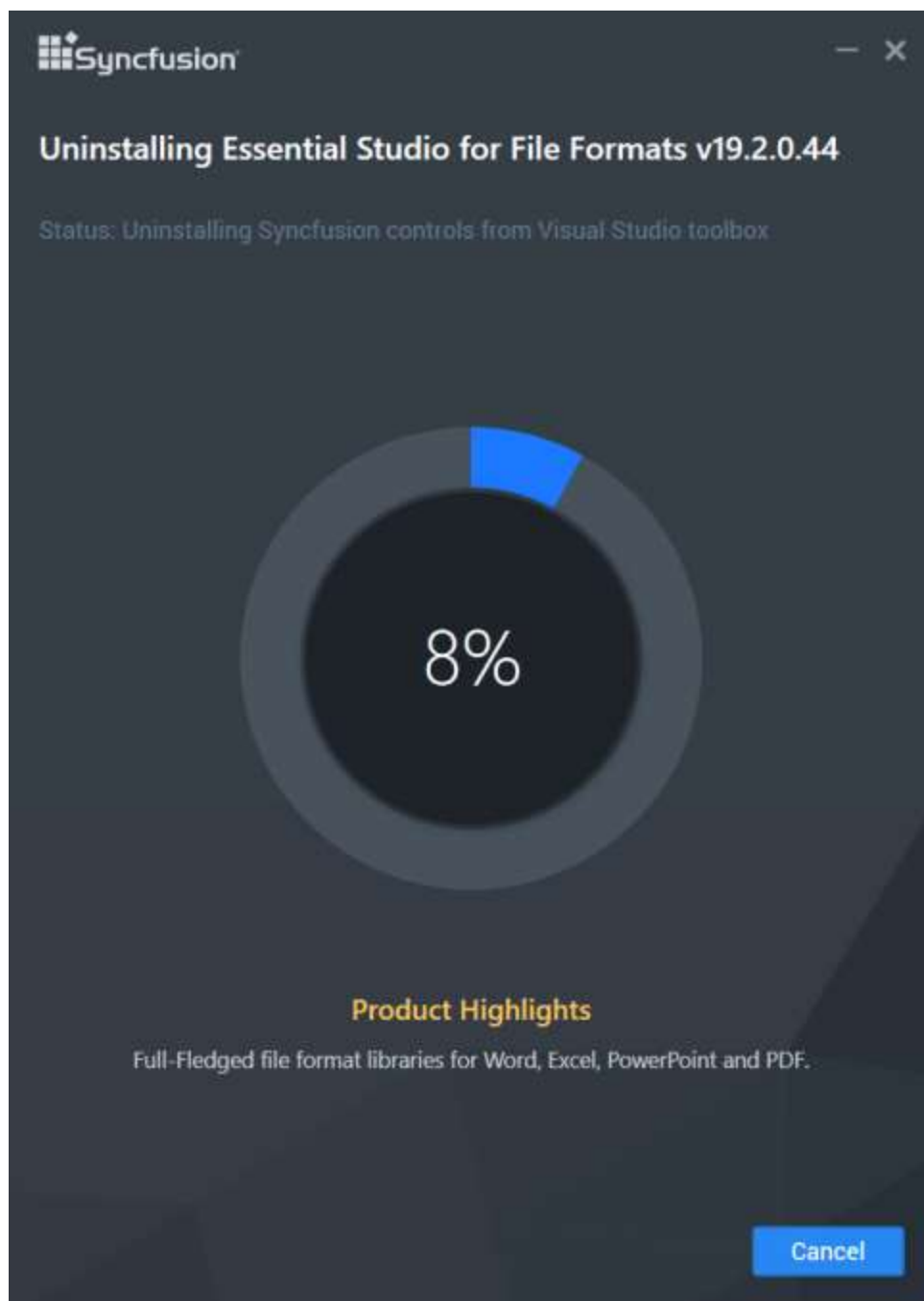
#### Confirmation Alert





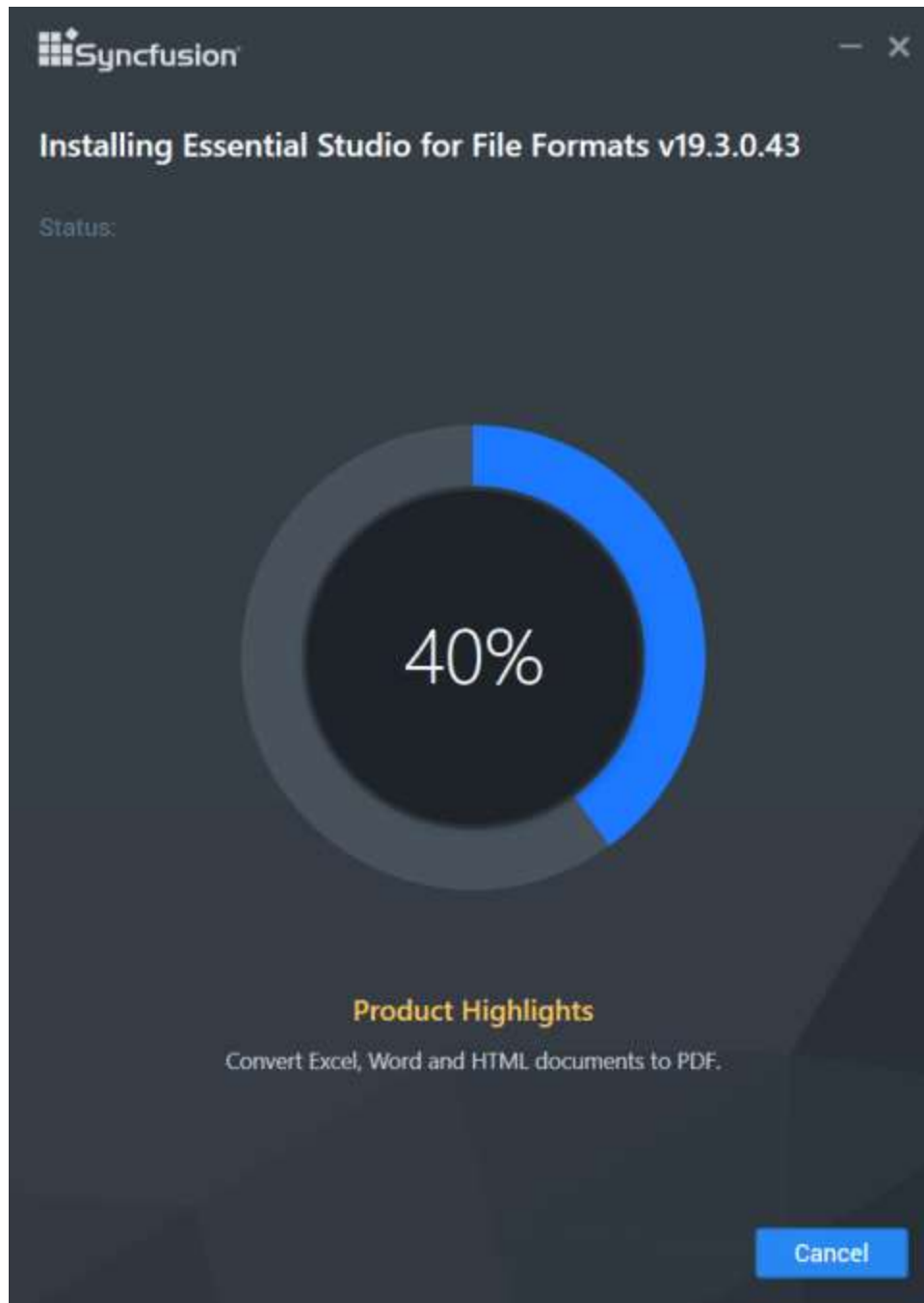
**Uninstall Progress:**





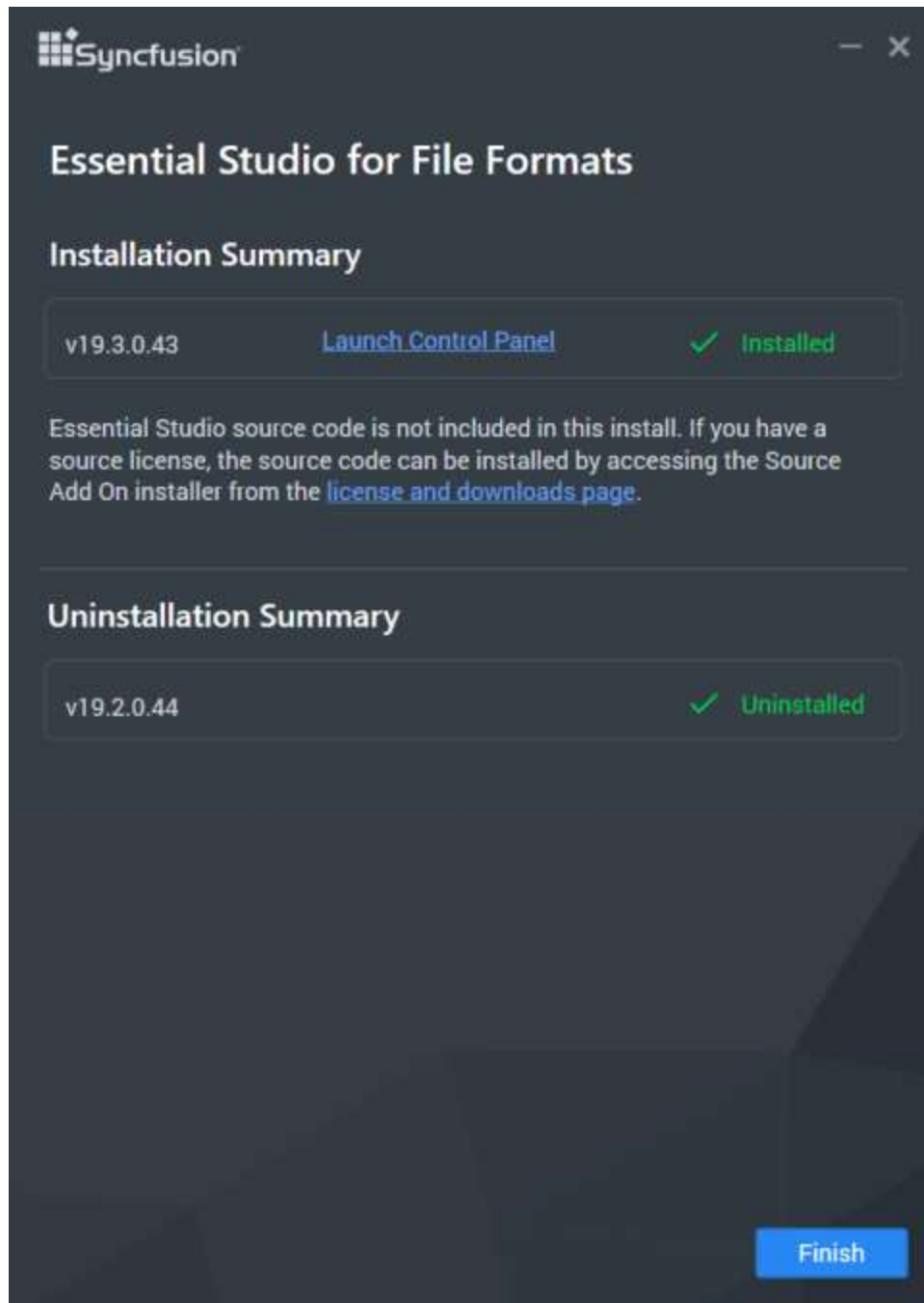
**Install Progress**





**Note:** The Completed screen is displayed once the FileFormats product is installed. If any version is selected to uninstall, The completed screen will display both install and uninstall status.





7. After installing, click the **Launch Control Panel** link to open the Syncfusion Control Panel.
8. Click the Finish button. Your system has been installed with the Syncfusion Essential Studio FileFormats product.

**Note:** \* You can find the required jars in the FileFormats installed location

- **Location:** {ProgramFilesFolder}\Syncfusion\Essential Studio\ {Platform}\ {version}\JarFiles
- **Example:** C:\Program Files (x86)\Syncfusion\Essential Studio\FileFormats\19.1.0.54\JarFiles



### Installing in silent mode

The Syncfusion Essential Studio FileFormats Installer supports installation and uninstallation via the command line.

#### Command Line Installation

To install through the Command Line in Silent mode, follow the steps below.

1. Run the Syncfusion FileFormats installer by double-clicking it. The Installer Wizard automatically opens and extracts the package. 2. The file *syncfusionessentialfileformats(version).exe* file will be extracted into the Temp directory. 3. Run %temp%. The Temp folder will be opened. The *syncfusionessentialfileformats(version).exe* file will be located in one of the folders. 4. Copy the extracted *syncfusionessentialfileformats\_(version).exe* file in local drive. 5. Exit the Wizard. 6. Run Command Prompt in administrator mode and enter the following arguments.

**Arguments:** "installer file path\SyncfusionEssentialStudio(platform)\_(version).exe" /Install silent /UNLOCKKEY:"(product unlock key)" [/log "{Log file path}"] [/InstallPath:{Location to install}] [/InstallSamples:{true/false}] [/InstallAssemblies:{true/false}] [/UninstallExistAssemblies:{true/false}] [/InstallToolbox:{true/false}]

**Note:** [...] – Arguments inside the square brackets are optional.

**Example:** "D:\Temp\syncfusionessentialfileformatsx.x.x.x.exe" /Install silent /UNLOCKKEY:"product unlock key" /log "C:\Temp\EssentialStudioPlatform.log" /InstallPath:C:\Syncfusion\x.x.x.x /InstallSamples:true /InstallAssemblies:true /UninstallExistAssemblies:true /InstallToolbox:true

7. Essential Studio for FileFormats is installed.

**Note:** x.x.x.x should be replaced with the Essential Studio version and the Product Unlock Key needs to be replaced with the Unlock Key for that version.

#### Command Line Uninstallation

Syncfusion Essential FileFormats can be uninstalled silently using the Command Line.

1. Run the Syncfusion FileFormats installer by double-clicking it. The Installer Wizard automatically opens and extracts the package. 2. The file *syncfusionessentialfileformats(version).exe* file will be extracted into the Temp directory. 3. Run %temp%. The Temp folder will be opened. The *syncfusionessentialfileformats(version).exe* file will be located in one of the folders. 4. Copy the extracted *syncfusionessentialfileformats\_(version).exe* file in local drive. 5. Exit the Wizard. 6. Run Command Prompt in administrator mode and enter the following arguments.

**Arguments:** "Copied installer file path\syncfusionessentialfileformats\_(version).exe" /uninstall silent

**Example:** "D:\Temp\syncfusionessentialfileformats\_x.x.x.x.exe" /uninstall silent

7. Essential Studio for WPF is uninstalled.

### Common Installation Errors

This article describes the most common installation errors, as well as the causes and solutions to those errors.

- [Unlocking the license installer using the trial key](#)
- [License has expired](#)

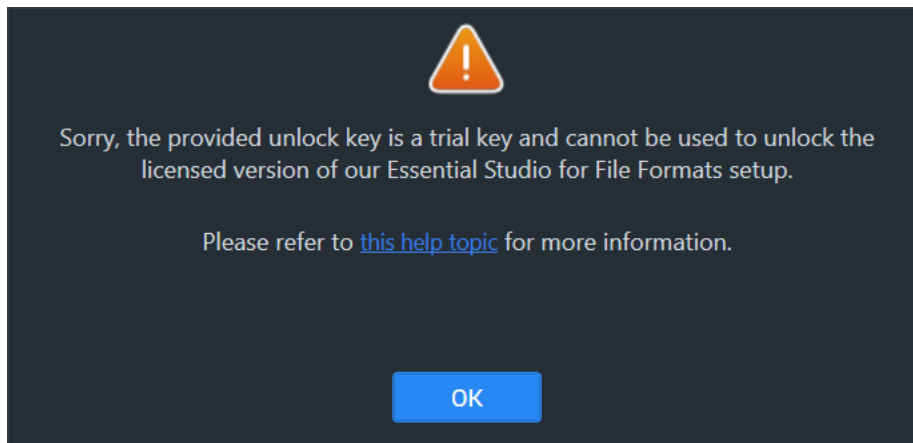


- [Unable to find a valid license or trial](#)
- [Unable to install because of another installation](#)
- [Unable to install due to controlled folder access](#)

Unlocking the license installer using the trial key

*Problem*

**Error Message:** Sorry, the provided unlock key is a trial unlock key and cannot be used to unlock the licensed version of our Essential Studio for FileFormats installer.



*Reason*

You are attempting to use a Trial unlock key to unlock the licensed installer.

*Suggested solution*

Only a licensed unlock key can unlock a licensed installer. So, to unlock the Licensed installer, use the Licensed unlock key. To generate the licensed unlock key, refer to [this](#) article.

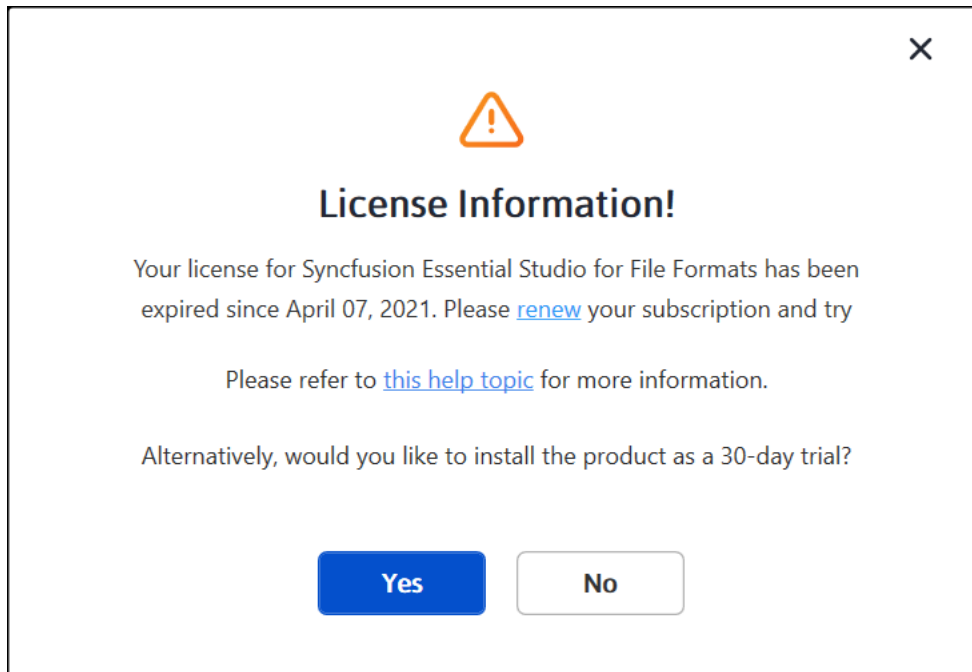
License has expired

*Problem*

**Error Message:** Your license for Syncfusion Essential Studio for FileFormats has been expired since {date}. Please renew your subscription and try again.

**Online Installer**





#### *Reason*

This error message will appear if your license has expired.

#### *Suggested solution*

You can choose from the options listed below.

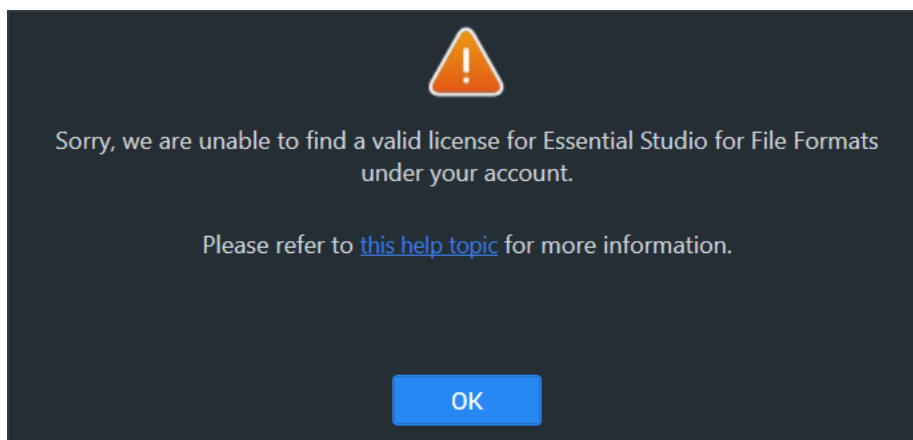
1. You can renew your subscription [here](#).
2. You can get a new license [here](#).
3. You can reach out to our sales team by emailing .
4. You can also extend the 30-day trial period after your license has expired.

#### Unable to find a valid license or trial

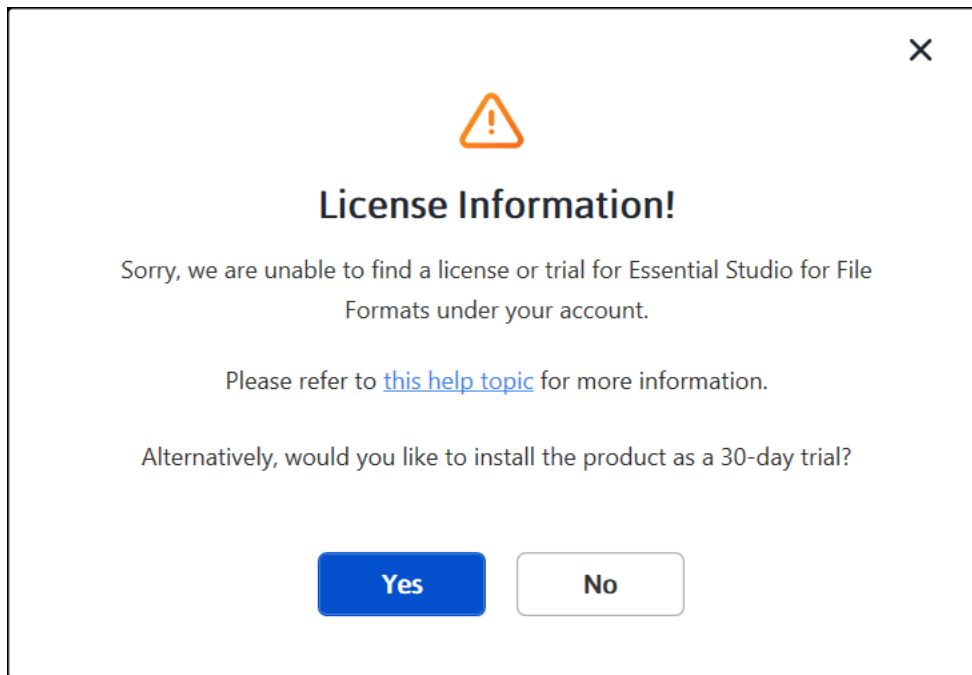
##### *Problem*

**Error Message:** Sorry, we are unable to find a valid license or trial for Essential Studio for FileFormats under your account.

<em>Offline installer</em>





**<em>Online installer</em>***Reason*

The following are possible causes of this error:

- When your trial period expired
- When you don't have a license or an active trial
- You are not the license holder of your license
- Your account administrator has not yet assigned you a license.

*Suggested solution*

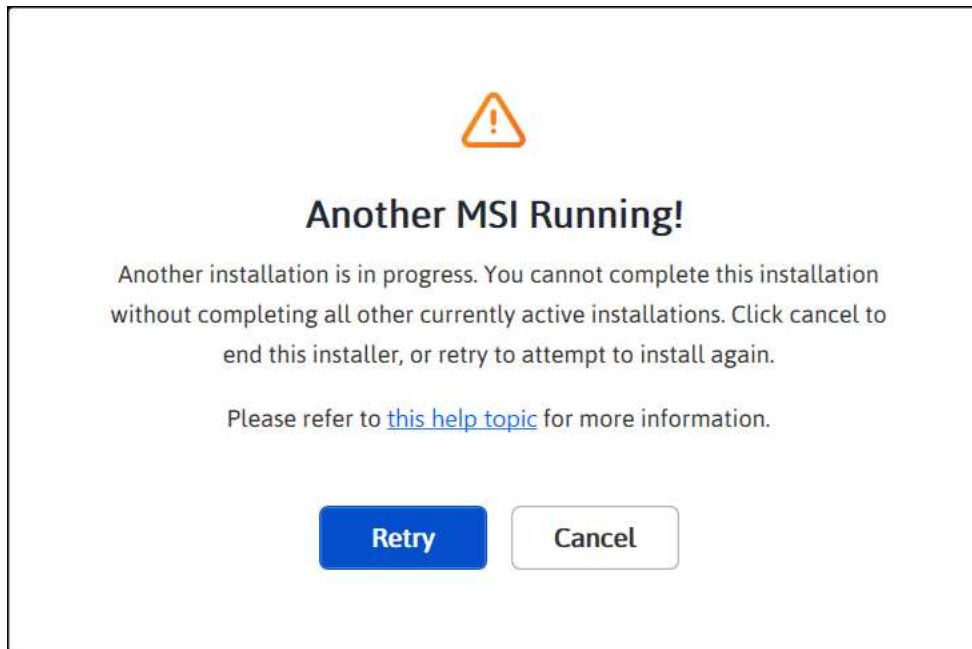
You can choose from the options listed below.

1. You can get a new license [here](#).
2. Contact your account administrator.
3. Send an email to to request a license.
4. You can reach out to our sales team by emailing .

*Unable to install because of another installation**Problem*

**Error Message:** Another installation is in progress. You cannot start this installation without completing all other currently active installations. Click cancel to end this installer or retry to attempt after currently active installation completed to install again.





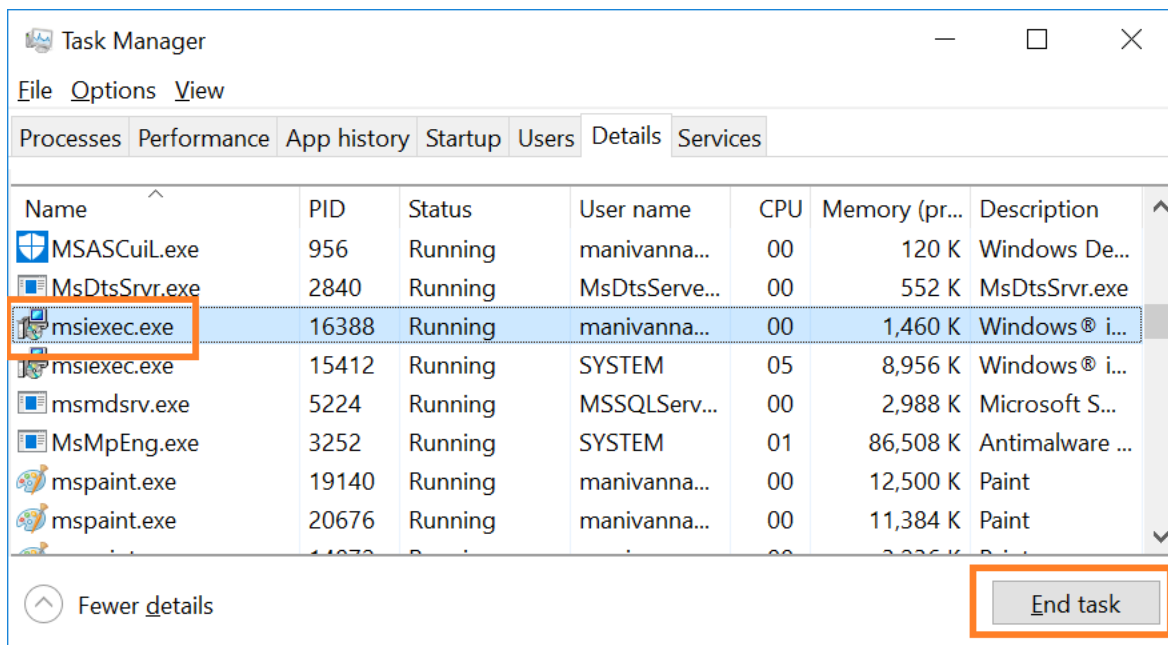
#### Reason

You are trying to install when another installation is already running in your machine.

#### Suggested solution

Open and kill the msixec process in the task manager and then continue to install Syncfusion. If the problem is still present, restart the computer and try Syncfusion installer.

1. Open the Windows Task Manager.
2. Browse the Details tab.
3. Select the msixec.exe and click **End task**.



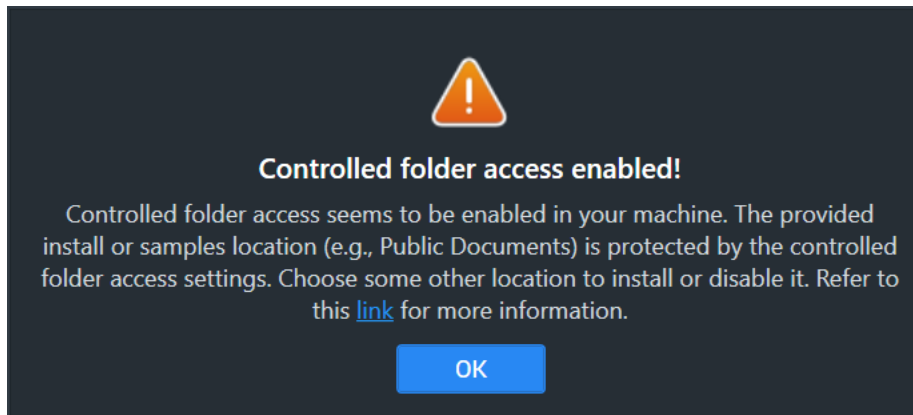


Unable to install due to controlled folder access

*Problem*

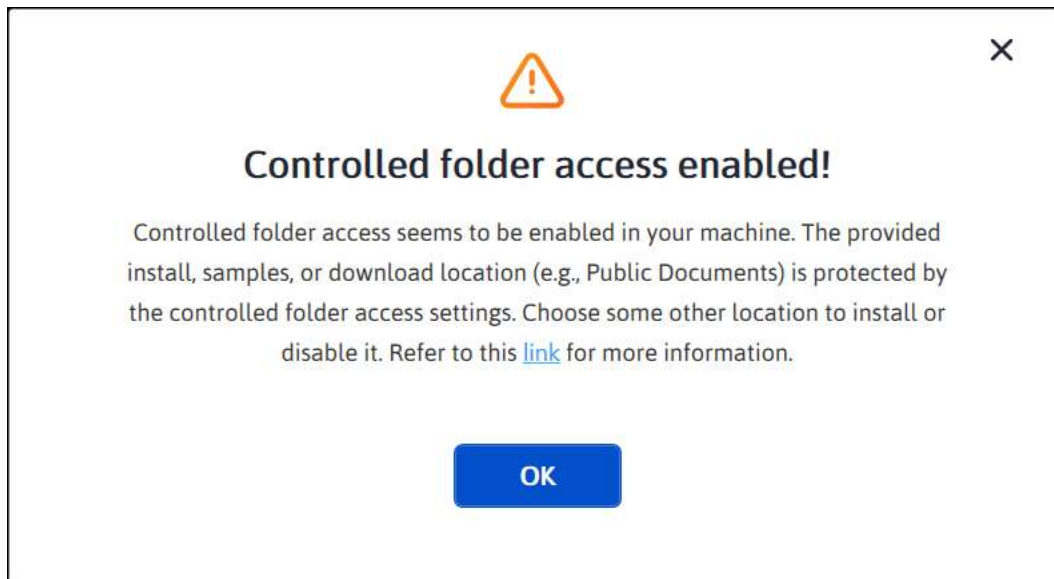
*Offline:*

**Error Message:** Controlled folder access seems to be enabled in your machine. The provided install or samples location (e.g., Public Documents) is protected by the controlled folder access settings.



*Online:*

**Error Message:** Controlled folder access seems to be enabled in your machine. The provided install, samples, or download location (e.g., Public Documents) is protected by the controlled folder access settings.



*Reason*

You have enabled controlled folder access settings on your computer.

*Suggested solution*

Select a different location to install or deactivate your machine's controlled folder access settings, and then try installing.



## Upgrading Syncfusion FileFormats

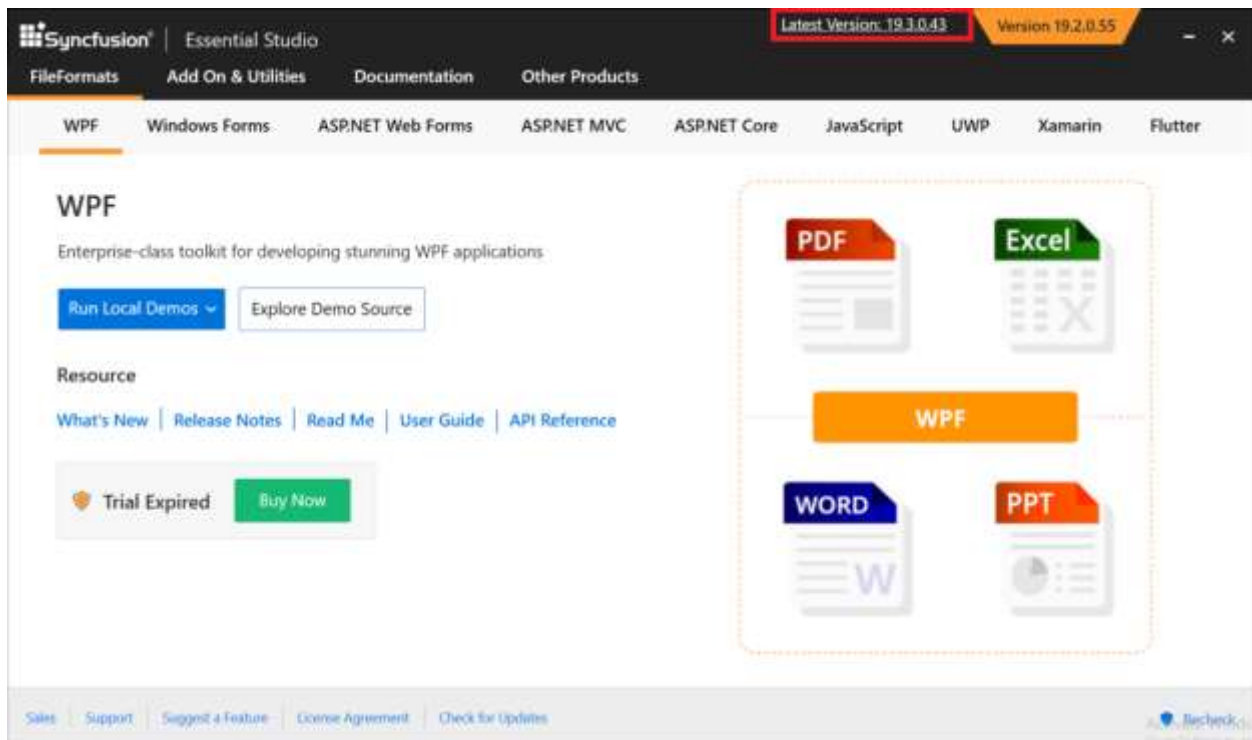
Syncfusion releases new volumes once every three months, with exciting new features. There will be one Service Pack release for this volume releases. Service Pack releases are provided to address major bug fixes in the volume releases.

You can upgrade to our latest version from any installed Syncfusion version.

See our **Upgrade Guide** for [PDF](#), [XlsIO](#), [Presentation](#) and [DocIO](#) FileFormats controls to learn more about the “Breaking Changes, Bug Fixes, Features and Known Issues” between your current version and the latest version you are trying to upgrade.

## Upgrading to the latest version

The most recent version of Syncfusion FileFormats can be downloaded and installed by clicking on the "Latest Version: {Version}" link at the top of the Syncfusion FileFormats Control Panel.



You can also upgrade to the latest version just by downloading and installing the products you require from [this](#) link. The existing installed versions are not required to be uninstalled.

It is not required to install the Volume release before installing the Service Pack release. As releases for Volume and Service Packs work independently, you can install the latest version with major bug fixes directly.

## Upgrade from trial version to license version

To upgrade from trial version, there are two possible solutions.

- Uninstall the trial version and install the fully licensed build from the [License & Downloads](#) section of our website.
- If you are using Syncfusion controls from [maven repository](#), replace the currently used trial license key with a paid license key that can be generated from the [License & Downloads](#) section



of our website. Refer to [this](#) topic for more information regarding registering the license in the application.

**Note:** License registration is not required if you reference Syncfusion Java packages from Licensed installer. These licensing changes applicable to all evaluators who refers the Syncfusion Java packages from evaluation installer and those who use Syncfusion controls from [maven repository](#).

### Available Syncfusion Java packages

Below are the Java packages available in syncfusion.

| Jar files                    | Short description   |
|------------------------------|---|
| syncfusion-docio             | This jar contains the core features needed for creating, reading, and editing a Word document programmatically. |
| syncfusion-javahelper        | This jar contains common functionalities required for the syncfusion file format libraries.                     |
| syncfusion-ej2-wordprocessor | This jar used for converting Word documents into Syncfusion Document Text (*.SFDI) format documents.            |

### Configure to download Syncfusion Java packages from Apache Maven

You can easily download the Syncfusion packages for Java using the [maven repository](#).

The following command shows how to mention the repository in Apache Maven.

#### XML

```
<repository>
<id>Syncfusion-Java</id>
<name>Syncfusion Java repo</name>
<url>https://jars.syncfusion.com/repository/maven-public/</url>
</repository>
```

The following command shows how to refer to the Syncfusion package, which needs to be used in your project as the dependency.

#### XML

```
<dependency>
<groupId>com.syncfusion</groupId>
<artifactId>syncfusion-docio</artifactId>
<version>18.4.0.30</version>
</dependency>
```

### Configure to download Syncfusion Java packages from Gradle

You can easily download the Syncfusion packages for Java using the [maven repository](#).

The following command shows how to mention the repository in Gradle.



```
repositories {  
    maven {  
        //Syncfusion maven repository to download the artifacts.  
        url "https://jars.syncfusion.com/repository/maven-public/"  
    }  
}
```

The following command shows how to refer to the Syncfusion package in Gradle, which needs to be used in your project as the dependency.

```
dependencies {  
    implementation 'com.syncfusion:syncfusion-docio:18.4.0.30'  
}
```

## Licensing

### Syncfusion Licensing in Java FileFormats

Starting from v19.1.0.x, if you reference Syncfusion Java packages from trial installer or from [maven repository](#) you must also include the Java platforms license key in your projects for the corresponding version.

#### *Difference between unlock key and license key*

Please note that this license key is different from the installer unlock key that you might have used in the past and needs to be separately generated from Syncfusion website. Refer [this](#) KB article to know more about difference between the Syncfusion Unlock Key and the Syncfusion License Key.

Trial message will be displayed as watermark in the generated documents, if Java packages referred from trial installer or from [maven repository](#)

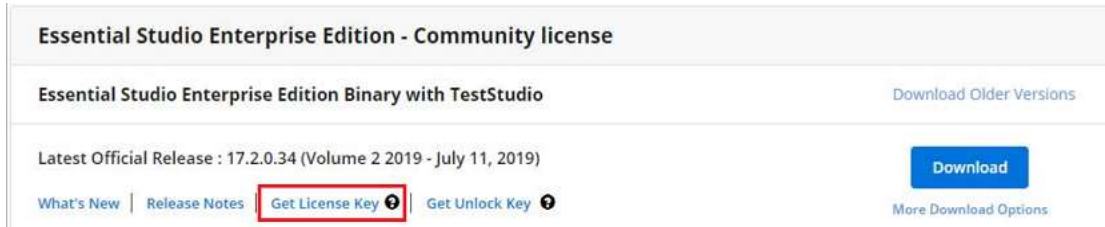
#### Example





## License Key Generation

License keys for Java can be generated from the [License & Downloads](#) or [Trail & Downloads](#) section from your Syncfusion account.



**Information:** \* Syncfusion license keys are **version and platform specific**, refer to the [KB](#) to generate the license key for the required version and platform.

- Refer this [KB](#) to know about which version of the Syncfusion license key should be used in the application.

## License Key Registration

The generated license key is just a string that needs to be registered before any Syncfusion control is initiated. The following code is used to register the license.

### C#

```
Syncfusion.Licensing.SyncfusionLicenseProvider.RegisterLicense("YOUR LICENSE KEY");
```

**Note:** Place the license key between double quotes. Also, ensure that Syncfusion.Licensing.dll is referenced in your project where the license key is being registered.

### Java

Recommended place to register the license for Java platform is given below.

Import 'syncfusion.licensing' package and register the license key in the **main method** of your console application.

### JAVA

```
// Refer the licensing package
import com.syncfusion.licensing.*;
static void main() {
    // Register Syncfusion license
    SyncfusionLicenseProvider.registerLicense("YOUR LICENSE KEY");
}
```

**Note:** License key registration is not required for Java before v19.1.

## Common Licensing Errors

Licensing error popup is displayed with various messages under different circumstances. Here are some ways to resolve different issues.



### License key not registered

The following error message will be shown if a Syncfusion license key has not been registered in your application.

**Error message:** This application was built using a trial version of Syncfusion Essential Studio. Please include a valid license to permanently remove this license validation message. You can also obtain a free 30 day evaluation license to temporarily remove this message during the evaluation period. Please refer to this help topic(<https://help.syncfusion.com/es/licensing/>) for more information.

**Solution:** Generate a valid license key from here [Licensed users](#) or [Trial users](#) for a specific version and platform.

### Invalid key

If the application is registered with an invalid key, another version of license key, or another platform's license key, the following error message will pop up when launching the application.

This error message will be shown if the license key is invalid. It is possible that the key is for another platform or for an older version.

**Error Message:** The included Syncfusion license is invalid. Please refer to this help topic(<https://help.syncfusion.com/es/licensing/invalid/>) for more information.

**Solution:** Generate a valid license key from here [Licensed users](#) or [Trial users](#) for a specific version and platform.

### Trial Expired

The following error message will be shown if the trial key has expired after 30 days.

**Error Message:** Your Syncfusion trial license has expired. Please refer to this help topic(<https://help.syncfusion.com/es/licensing/expired/>) for more information.

**Solution:** Purchase from [here](#) to get a valid Syncfusion license.

### Platform Mismatch

If the application is registered with another platform's license key, the following error message will pop up when launching the application.

**Error Message:**The included Syncfusion license is invalid (Platform mismatch). Please refer to this help topic(<https://help.syncfusion.com/es/licensing/platform-mismatch/>) for more information.

**Solution:** Generate a valid license key from here [Licensed users](#) or [Trial users](#) for a specific version and platform.

### Version Mismatch

If the application is registered with another version's license key, the following error message will pop up when launching the application.

**Error Message:**The included Syncfusion license ({Registered Version}) is invalid for version {Required version}. Please refer to this help topic(<https://help.syncfusion.com/es/licensing/version-mismatch/>) for more information.

**Solution:** Generate a valid license key from here [Licensed users](#) or [Trial users](#) for a specific version and platform. Kindly follow the [KB](#) to generate license key.



### Could not load Syncfusion.Licensing.dll assembly version...?

Make sure that all the referenced Syncfusion assemblies are of the same version. Try cleaning and rebuilding the application to resolve assembly conflict issues.

### Licensing FAQ

#### How to upgrade from Trial version after purchasing a license?

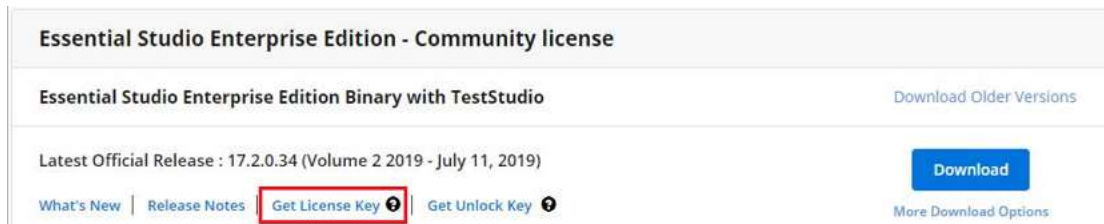
To upgrade from trial version, there are two possible solutions.

- Uninstall the trial version and install the fully licensed build from the [License & Downloads](#) section of our website.
- If you are using Syncfusion controls from [maven repository](#), replace the currently used trial license key with a paid license key that can be generated from the [License & Downloads](#) section of our website. Refer to [this](#) topic for more information regarding registering the license in the application.

**Note:** License registration is not required if you reference Syncfusion Java packages from Licensed installer. These licensing changes applicable to all evaluators who refers the Syncfusion Java packages from evaluation installer and those who use Syncfusion controls from [maven repository](#).

#### Where can I get a license key?

License keys can be generated from the [License & Downloads](#) or [Trail & Downloads](#) section of the Syncfusion website.



**Information:** \* Syncfusion license keys are **version and platform specific**, refer to the [KB](#) to generate the license key for the required version and platform.

- Refer this [KB](#) to know about which version of the Syncfusion license key should be used in the application.

## Word Library

### Overview of Word library (DocIO)

Essential DocIO is a native Java Word library that is used by developers to create, read, write, and convert the Microsoft Word documents in the Java application without Microsoft Word or interop dependencies.

It is a non-UI component that provides a full-fledged document instance model similar to the Microsoft Office COM libraries to iterate with the document elements explicitly and perform necessary manipulation. It is built from scratch in Java and does not require Microsoft Word to be installed in the machine. It supports Word 2007 and later versions documents.

### Key Features



- Support to [create Word document](#) from scratch.
- Support to open, modify, and save the existing Word documents.
- Advanced [Mail merge](#) support with different data sources.
- Ability to create or edit Word 2007 and later version documents, and convert them to commonly used file formats such as [WordML](#), [TXT](#), [HTML](#) and vice versa.
- Ability to create and manipulate [shapes](#), and [group shape](#) in the DOCX and WordML format documents.
- Ability to read and write the [built-in and custom document properties](#).
- Support to insert and edit the [form fields](#).
- Ability to insert the [bookmarks](#) and navigate corresponding bookmarks to insert, replace, and delete content.

### Compatible Microsoft Word Versions

- Microsoft Word 2007
- Microsoft Word 2010
- Microsoft Word 2013
- Microsoft Word 2016
- Microsoft Word 2019

**Tips:** Sometimes a Java application may throw an error like this "Java.Lang.OutOfMemoryError." Usually, this error occurs due to insufficient RAM heap space to allocate an object in the Java heap. To avoid this kind of problem, we suggest [increasing the JVM heap size](#).

**Note:** Currently, Essential DocIO supports .DOCX, .DOTX, .DOTM, .DOCM, .WordML, .RTF, .Txt, WordML, HTML. Please refer to the [supported and unsupported features](#) in the essential DocIO.

### External Jars Required

The following jar files are required to be referenced in your Java application.

1. syncfusion-docio
2. syncfusion-javahelper

Get the dependent jar files by installing [file formats controls](#). You can find the required jars in the build installed drive.

**Location:** {ProgramFilesFolder}\Syncfusion\Essential Studio\ {Platform}\ {version}\JarFiles

**Example:** C:\Program Files (x86)\Syncfusion\Essential Studio\FileFormats\18.3.0.35\JarFiles

### Supported Java versions

Syncfusion Java Word library supports Java SE 8.0(1.8) or above versions.

### Download Jars

You can download the jars from the Syncfusion [maven repository](#) to use our artifacts in your projects. It helps you to use the Syncfusion Java packages without installing Essential Studio or platform installation to development with Syncfusion controls.

### Frequently Asked questions

- [How to configure Syncfusion Java packages in Gradle?](#)



- [How to configure Syncfusion Java packages in Apache Maven?](#)

## Getting Started of Word library (DocIO)

In this page, you can see how to create a simple Word document by using Syncfusion Java Word library APIs. For creating and manipulating a Word document, the following jar files are required to be referenced in your Java application.

| Jar files             | Short description   |
|-----------------------|---|
| syncfusion-docio      | This jar contains the core features needed for creating, reading, manipulating a Word document. |
| syncfusion-javahelper | This jar contains common functionalities required for file format libraries.                    |

Get the dependent jar files by installing [file formats controls](#). You can find the required jars in the build installed drive.

**Location:** {ProgramFilesFolder}\Syncfusion\Essential Studio\ {Platform}\ {version}\JarFiles

**Example:** C:\Program Files (x86)\Syncfusion\Essential Studio\FileFormats\18.3.0.35\JarFiles

**Note:** 1. Starting with v19.1.0.x, if you reference Syncfusion Java packages from trial setup or from the maven repository, you also have to add "syncfusion.licensing" package reference and include a license key in your projects. Please refer to this [link](#) to know about registering Syncfusion license key in your application to use our components.

Import the following package in your Java application.

### JAVA

```
import com.syncfusion.docio.*;
import com.syncfusion.Javahelper.*;
```

## Creating a new Word document with few lines of code

The following code example explains how to create a new Word document with a few lines of the code.

### JAVA

```
//Creates an instance of WordDocument Instance (Empty Word Document).
WordDocument document = new WordDocument();
//Add a section and paragraph in the empty document.
document.ensureMinimal();
//Append text to the last paragraph of the document.
document.getLastParagraph().appendText("Hello World");
//Save and close the Word document.
document.save("Result.docx");
document.close();
```



### Creating a new Word document from scratch with basic elements

An entire Word document is represented by an instance of the `WordDocument` and it is the root element of DocIO's DOM. The Word document contains a collection of sections. A Word document must contain at least one section.

A section represents a group of paragraphs, tables, and more, that have a specific set of properties used to define the pages, a number of columns, headers, and footers, and more, that decides how the text appears. A section should contain at least one paragraph in this body.

The following code example explains how to add a section into a `WordDocument` instance.

#### JAVA

```
//Create an instance of WordDocument Instance (Empty Word Document).
WordDocument document = new WordDocument();
//Add a new section into the Word document.
IWSection section = document.addSection();
//Specifies the page margins.
section.getPageSetup().getMargins().setAll(50);
```

All the textual contents in a Word document are represented by the paragraphs. Within the paragraph, the textual contents are grouped into one or more child elements such as text range, field, and more. Each text range represents a region of text with a common set of rich text formatting.

The following code example explains how to add a paragraph into a Word document.

#### JAVA

```
//Add a new simple paragraph into the section.
IWParagraph firstParagraph = section.addParagraph();
//Set the paragraph's horizontal alignment as justify.
firstParagraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignme
nt.Justify);
//Add a text range into the paragraph.
IWTextRange firstTextRange = firstParagraph.appendText("AdventureWorks
Cycles,");
//set the font formatting of the text range.
firstTextRange.getCharacterFormat().setBoldBidi(true);
firstTextRange.getCharacterFormat().setFontName("Calibri");
firstTextRange.getCharacterFormat().setFontSize(14);
//Add another text range into the paragraph.
IWTextRange secondTextRange = firstParagraph.appendText(" the fictitious
company on which the AdventureWorks sample databases are based, is a large,
multinational manufacturing company.");
//set the font formatting of the text range.
secondTextRange.getCharacterFormat().setFontName("Calibri");
secondTextRange.getCharacterFormat().setFontSize(11);
```

The following code example shows how to add an image into the Word document.

#### JAVA

```
//Add another paragraph and aligns it as a center.
IWParagraph paragraph = section.addParagraph();
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Ce
nter);
```



```
//Set after spacing for paragraph.
paragraph.getParagraphFormat().setAfterSpacing(6);
//Add a picture into the paragraph.
IWPicture picture = paragraph.appendPicture(new
FileInputStream("DummyProfilePicture.jpg"));
//Specify the size of the picture.
picture.setHeight(86);
picture.setWidth(81);
```

A table is another important element in the Word that contains a set of paragraphs arranged in rows and columns. You can create a simple as well as a complex table by using the Essential DocIO's API. The following code example creates a simple table and adds content to it. Each table cell must contain at least one paragraph.

### JAVA

```
//Add a table into the Word document.
IWTable table = section.addTable();
//Create the specified number of rows and columns.
table.resetCells(2,2);
//Access the instance of the cell (first row, first cell).
WTableCell firstCell = table.getRows().get(0).getCells().get(0);
//Specifies the width of the cell.
firstCell.setWidth(150);
//Add a paragraph into the cell; a cell must have atleast 1 paragraph.
paragraph=firstCell.addParagraph();
IWTextRange textRange = paragraph.appendText("Profile picture");
textRange.getCharacterFormat().setBold(true);
//Access the instance of cell (first row, second cell).
WTableCell secondCell = table.getRows().get(0).getCells().get(1);
secondCell.setWidth(330);
paragraph=secondCell.addParagraph();
//Add text to the paragraph.
textRange=paragraph.appendText("Description");
textRange.getCharacterFormat().setBold(true);
firstCell=table.getRows().get(1).getCells().get(0);
firstCell.setWidth(150);
//Add image to the paragraph.
paragraph=firstCell.addParagraph();
//Set after spacing for paragraph.
paragraph.getParagraphFormat().setAfterSpacing(6);
IWPicture profilePicture = paragraph.appendPicture(new
FileInputStream("DummyProfilePicture.jpg"));
//Set the height and width for the image.
profilePicture.setHeight(98);
profilePicture.setWidth(95);
//Access the instance of cell (second row, second cell) and adds text.
secondCell=table.getRows().get(1).getCells().get(1);
secondCell.setWidth(330);
paragraph=secondCell.addParagraph();
textRange=paragraph.appendText("AdventureWorks Cycles, the fictitious
company on which the AdventureWorks sample databases are based, is a large,
multinational manufacturing company.");
```



Essential DocIO allows you to create simple and multi-level lists. The following code sample explains how to create a numbered and bulleted list.

#### **JAVA**

```
//Write the default numbered list.
paragraph = section.addParagraph();
//Set before spacing for paragraph.
paragraph.getParagraphFormat().setBeforeSpacing(6);
paragraph.appendText("Level 0");
//Apply the default numbered list formats.
paragraph.getListFormat().applyDefNumberedStyle();
//Applies list formatting.
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setLeft
Indent(36);
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setFirs
tLineIndent(-18);
paragraph.getListFormat().getCurrentListLevel().setNumberAlignment(ListNumbe
rAlignment.Left);
paragraph = section.addParagraph();
paragraph.appendText("Level 1");
//Specify the list format to continue from the last list.
paragraph.getListFormat().continueListNumbering();
//Increment the list level.
paragraph.getListFormat().increaseIndentLevel();
//Applies list formatting.
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setLeft
Indent(72);
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setFirs
tLineIndent(-18);
paragraph.getListFormat().getCurrentListLevel().setNumberAlignment(ListNumbe
rAlignment.Left);
paragraph = section.addParagraph();
paragraph.appendText("Level 0");
//Decrement the list level.
paragraph.getListFormat().continueListNumbering();
paragraph.getListFormat().decreaseIndentLevel();
//Applies list formatting.
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setLeft
Indent(36);
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setFirs
tLineIndent(-18);
paragraph.getListFormat().getCurrentListLevel().setNumberAlignment(ListNumbe
rAlignment.Left);
//Write the default bulleted list.
section.addParagraph();
paragraph = section.addParagraph();
paragraph.appendText("Level 0");
//Apply the default bulleted list formats.
paragraph.getListFormat().applyDefBulletStyle();
//Applies list formatting.
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setLeft
Indent(36);
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setFirs
tLineIndent(-18);
paragraph.getListFormat().getCurrentListLevel().setNumberAlignment(ListNumbe
rAlignment.Left);
```



```
paragraph = section.addParagraph();
paragraph.appendText("Level 1");
//Specify the list format to continue from the last list.
paragraph.getListFormat().continueListNumbering();
//Increment the list level.
paragraph.getListFormat().increaseIndentLevel();
//Applies list formatting.
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setLeft
Indent(72);
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setFirs
tLineIndent(-18);
paragraph.getListFormat().getCurrentListLevel().setNumberAlignment(ListNumbe
rAlignment.Left);
paragraph = section.addParagraph();
paragraph.appendText("Level 0");
//Specify the list format to continue from the last list.
paragraph.getListFormat().continueListNumbering();
//Decrement the list level.
paragraph.getListFormat().decreaseIndentLevel();
//Applies list formatting.
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setLeft
Indent(36);
paragraph.getListFormat().getCurrentListLevel().getParagraphFormat().setFirs
tLineIndent(-18);
paragraph.getListFormat().getCurrentListLevel().setNumberAlignment(ListNumbe
rAlignment.Left);
section.addParagraph();
```

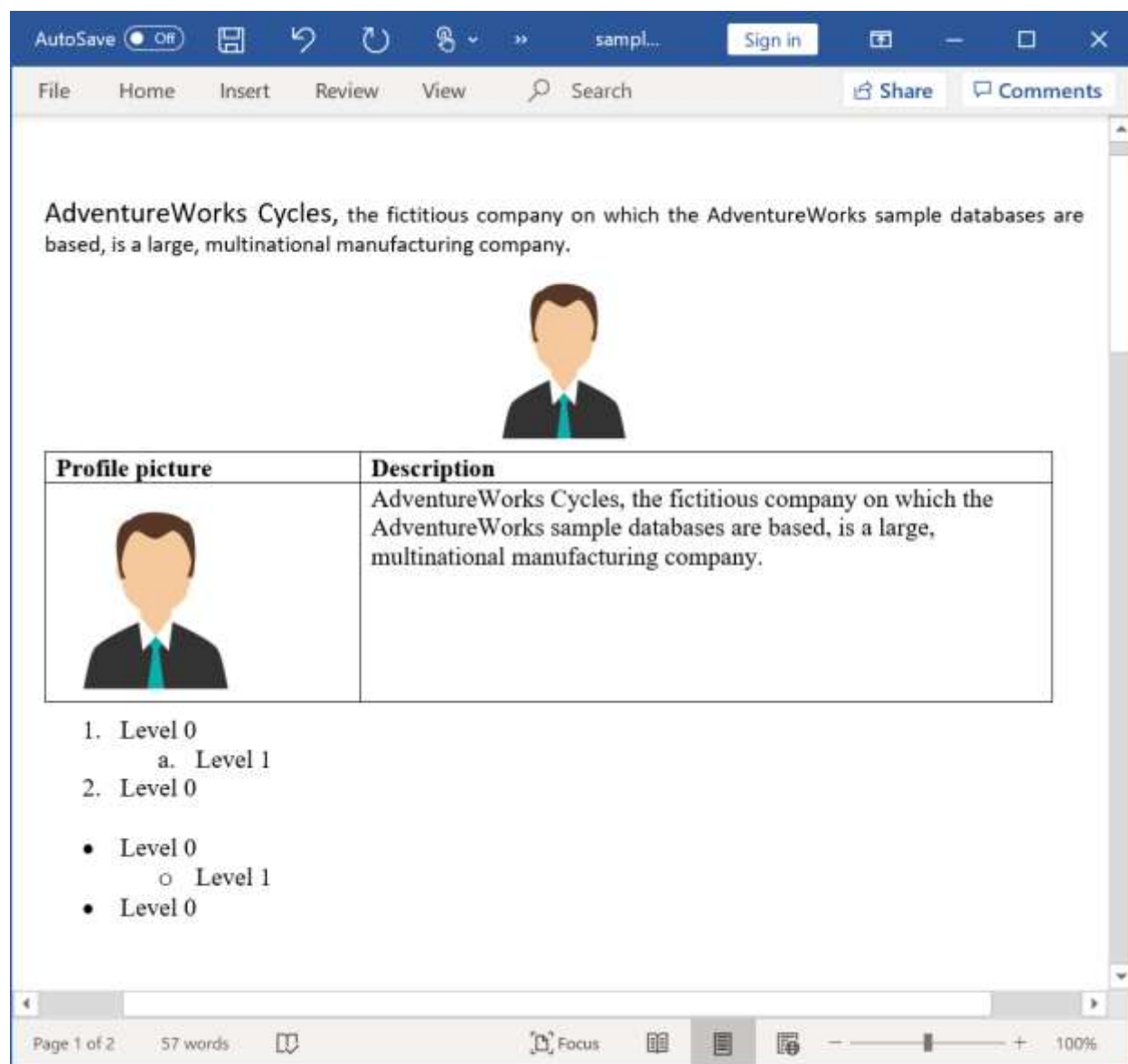
Finally, save the document in the file system and close its instance.

#### **JAVA**

```
//Save the document in the given name and format.
document.save(outputFileName, FormatType.Docx);
//Release the resources occupied by the WordDocument instance.
document.close();
```

The resultant Word document looks as follows.





### Performing Mail merge

Essential DocIO allows you to generate documents by filling data in the template document from the data source. Mail merge operation automatically maps the column name in the data source and names of the merge fields in the template Word document and fills the data.

The following data sources are supported by the Essential DocIO for performing the Mail merge.

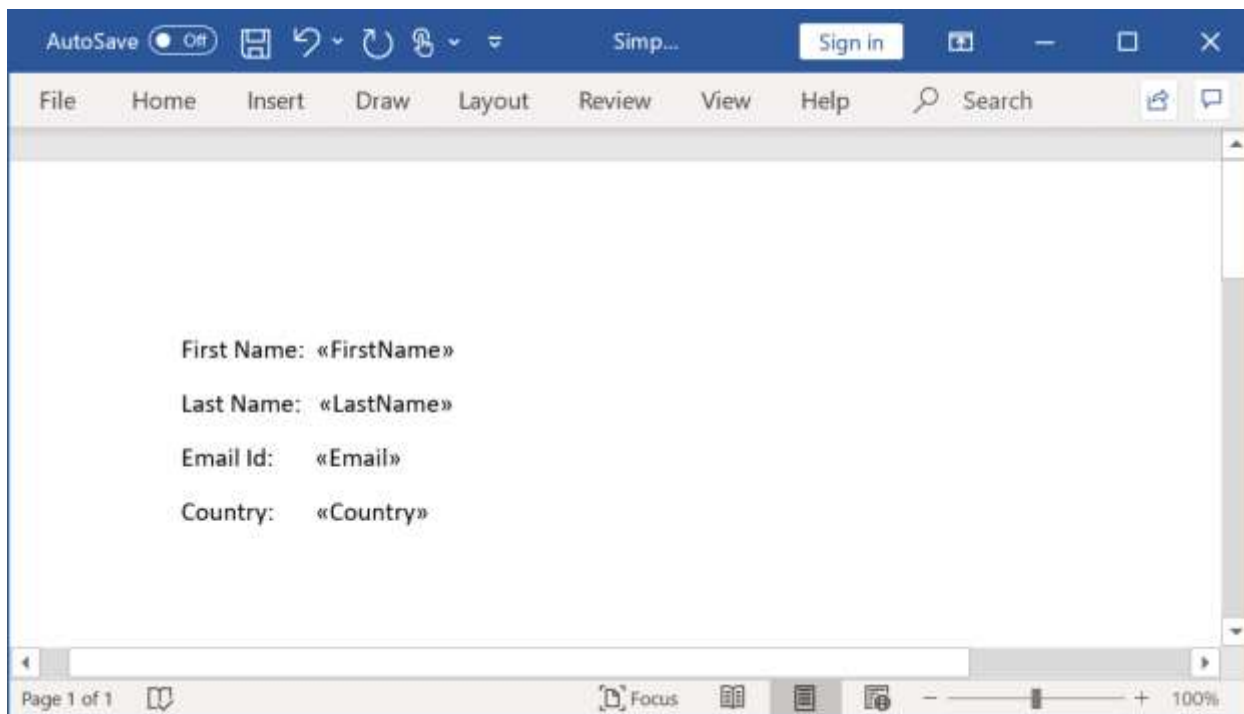
- String Arrays
- DataSet
- DataTable

Also, you can perform more than one Mail merge operations over the same template to generate documents as per your requirement.

Follow the given steps to perform a simple Mail merge in a Word document.



Let's consider that you have a template Word document with the merge fields as shown.



The `MailMerge` class provides various overloads for the `execute` method to perform a Mail merge from the various data source. The Mail merge operation replaces the matching merge fields with the respective data.

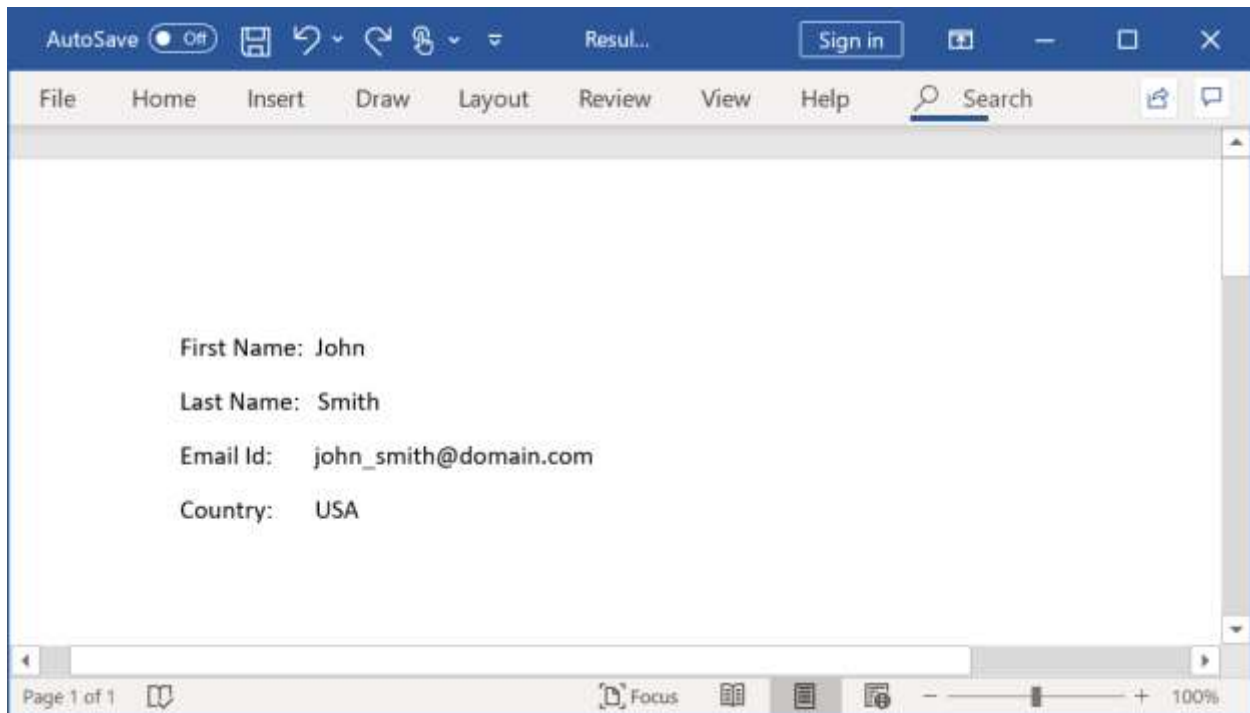
The following code example shows how to perform a simple Mail merge by using a string array.

#### **JAVA**

```
//Load the template document with the required merge fields.
WordDocument document = new
WordDocument("../SimpleMailMergeTemplate.docx");
//Initialize the string array with field names.
String[] fieldNames = new String[] {"FirstName", "LastName", "Email",
"Country"};
//Initialize the string array with field values.
String[] fieldValues = new String[] {"John", "Smith",
"john_smith@domain.com", "USA"};
//Executes the Mail merge operation that replaces the matching field names
with field values respectively.
document.getMailMerge().execute(fieldNames, fieldValues);
//Save and close the WordDocument instance.
document.save("Result.docx");
document.close();
```

The resultant Word document looks as follows.





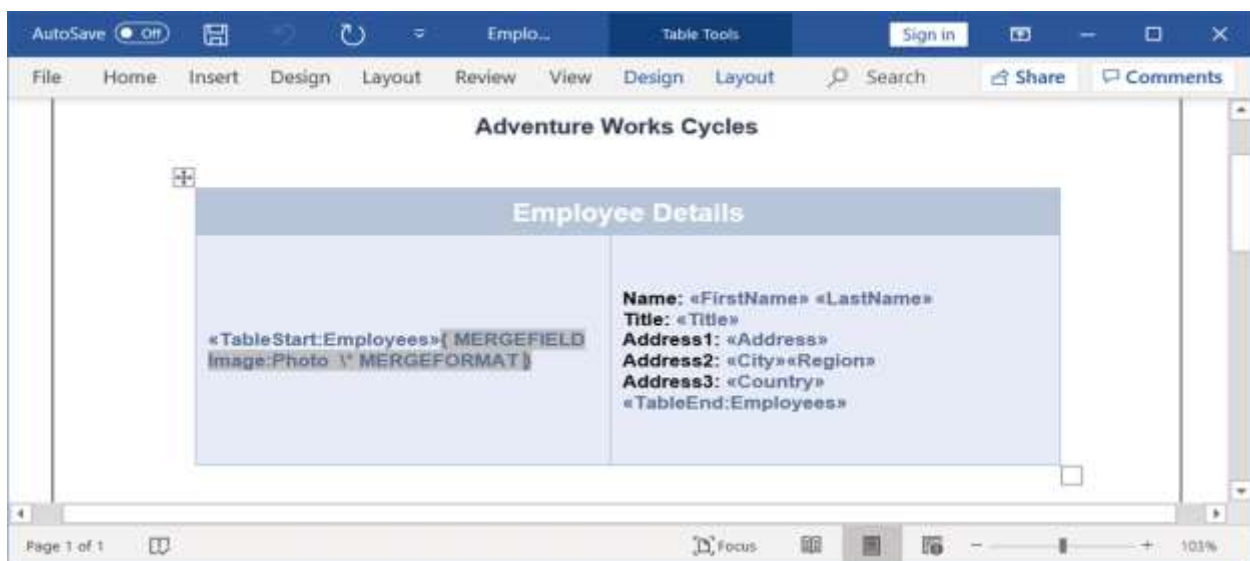
### Simple Mail merge with Group

You can perform a Mail merge with the group to append multiple records from the data source into a single document. The group is a part of the document enclosed by two special merge fields named «TableStart:TableName» and «TableEnd:TableName»

- «TableStart:TableName»: Denotes the start of the group.
- «TableEnd:TableName»: Denotes the end of the group.

The region between these two merge fields gets repeated for every record from the data source.

For example – let's consider that you have a template document as shown.





Here in this template, the employees are the group name and the exact same name should be used while performing the Mail merge through code. There are two special merge fields “TableStart:Employees” and “TableEnd:Employees” to denote the start and end of the Mail merge group.

To merge an image in the replace of a merge field, you need to add a prefix (“Image:”)the merge field name.

For example, the merge field name should be like “<<Image:Photo>>”(<<Image:MergeFieldName>>)

The following code example shows how to perform a Mail merge with objects.

#### JAVA

```
//Loads an existing Word document into DocIO instance.
WordDocument document = new WordDocument("EmployeesReportDemo.docx");
//Gets the employee details as IEnumerable collection.
ListSupport<Employee> employeeList = getEmployees();
//Uses the mail merge events handler for image fields.
document.getMailMerge().MergeImageField.add("mergeField_EmployeeImage", new
MergeImageFieldEventHandler() {
ListSupport<MergeImageFieldEventHandler> delegateList = new
ListSupport<MergeImageFieldEventHandler>(
MergeImageFieldEventHandler.class);
//Represents event handling for MergeFieldEventHandlerCollection.
public void invoke(Object sender, MergeImageFieldEventArgs args) throws
Exception
{
mergeField_EmployeeImage(sender, args);
}
//Represents the method that handles MergeField event.
public void dynamicInvoke(Object... args) throws Exception
{
mergeField_EmployeeImage((Object) args[0], (MergeImageFieldEventArgs)
args[1]);
}
//Represents the method that handles MergeField event to add collection
item.
public void add(MergeImageFieldEventHandler delegate) throws Exception
{
if (delegate != null)
delegateList.add(delegate);
}
//Represents the method that handles MergeField event to remove collection
item.
public void remove(MergeImageFieldEventHandler delegate) throws Exception
{
if (delegate != null)
delegateList.remove(delegate);
}
});
//Creates an instance of MailMergeDataTable by specifying MailMerge group
name and IEnumerable collection.
MailMergeDataTable dataSource = new
MailMergeDataTable("Employees",employeeList);
//Executes the mail merge for group.
document.getMailMerge().executeGroup(dataSource);
```



```
//Saves and closes the WordDocument instance.
document.save("Sample.docx");
document.close();
```

The following code example shows getEmployees method which is used to get data for mail merge.

#### **JAVA**

```
public ListSupport<Employee> getEmployees() throws Exception
{
    ListSupport<Employee> employees = new ListSupport<Employee>(Employee.class);
    employees.add(new Employee("Nancy","Smith","Sales Representative","505 -
    20th Ave. E. Apt. 2A","Seattle","WA","USA","Nancy.png"));
    employees.add(new Employee("Andrew","Fuller","Vice President, Sales","908 W.
    Capital Way","Tacoma","WA","USA","Andrew.png"));
    return employees;
}
```

The following code example shows how to bind the image from file system during Mail merge process by using MergeImageFieldEventHandler.

#### **JAVA**

```
private void mergeField_EmployeeImage(Object sender,
MergeImageFieldEventArgs args) throws Exception
{
    //Binds image from file system during mail merge.
    if ((args.getFieldName()).equals("Photo"))
    {
        String ProductFileName = args.getFieldValue().toString();
        //Gets the image from file system.
        FileStreamSupport imageStream = new FileStreamSupport(ProductFileName,
        FileMode.Open, FileAccess.Read);
        ByteArrayInputStream stream = new
        ByteArrayInputStream(imageStream.toArray());
        args.setImageStream(stream);
    }
}
```

The following code example provides supporting class for the above code.

#### **JAVA**

```
public class Employee
{
    private String _firstName;
    private String _lastName;
    private String _address;
    private String _city;
    private String _region;
    private String _country;
    private String _title;
    private String _photo;
    public String getFirstName() throws Exception
    {
        return _firstName;
    }
}
```



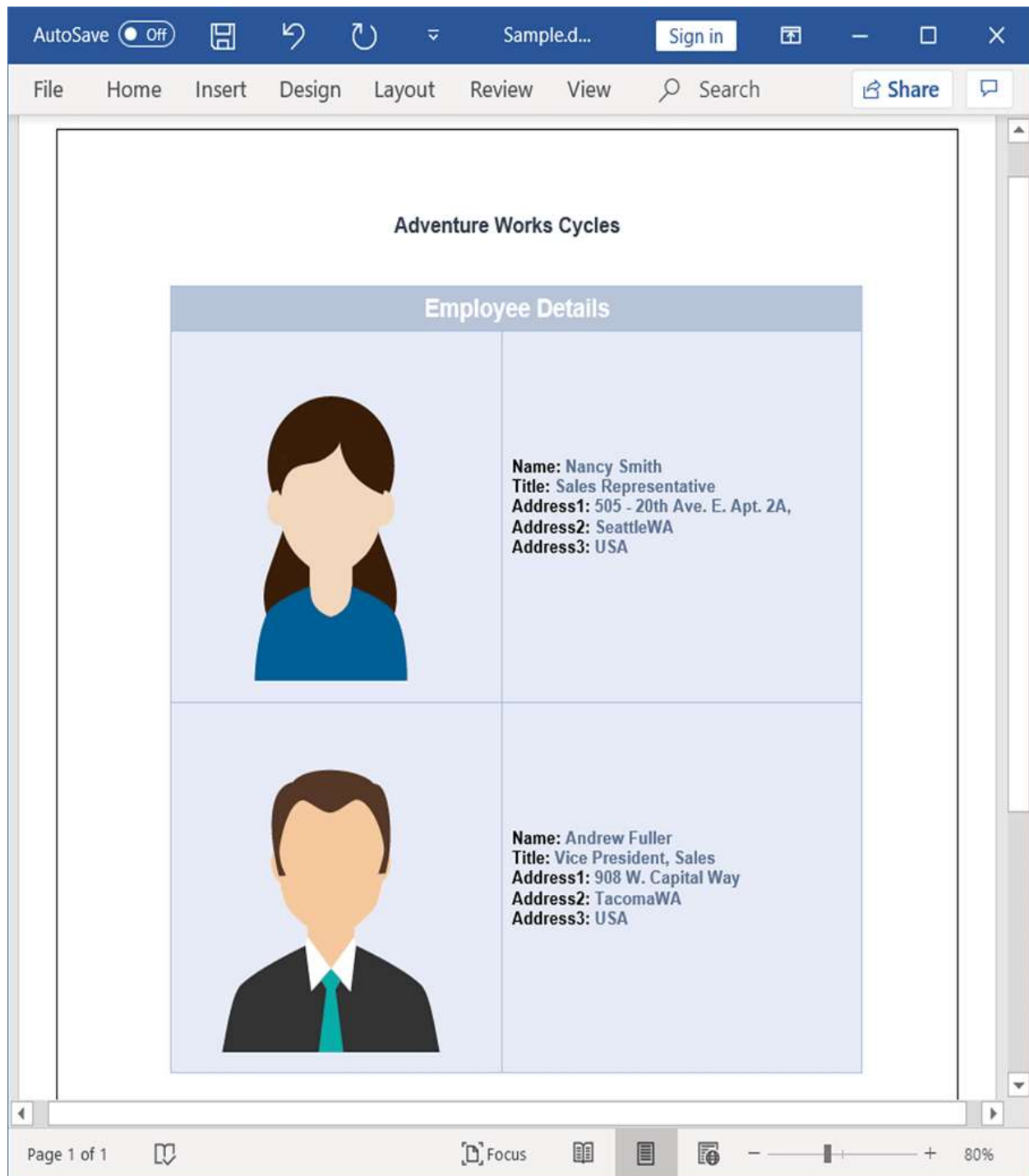
```
}  
public String setFirstName(String value) throws Exception  
{  
    _firstName=value;  
    return value;  
}  
public String getLastName() throws Exception  
{  
    return _lastName;  
}  
public String setLastName(String value) throws Exception  
{  
    _lastName=value;  
    return value;  
}  
public String getAddress() throws Exception  
{  
    return _address;  
}  
public String setAddress(String value) throws Exception  
{  
    _address=value;  
    return value;  
}  
public String getCity() throws Exception  
{  
    return _city;  
}  
public String setCity(String value) throws Exception  
{  
    _city=value;  
    return value;  
}  
public String getRegion() throws Exception  
{  
    return _region;  
}  
public String setRegion(String value) throws Exception  
{  
    _region=value;  
    return value;  
}  
public String getCountry() throws Exception{  
    return _country;  
}  
public String setCountry(String value) throws Exception  
{  
    _country=value;  
    return value;  
}  
public String getTitle() throws Exception  
{  
    return _title;  
}  
public String setTitle(String value) throws Exception  
{  
    _title=value;  
}
```



```
return value;
}
public String getPhoto() throws Exception
{
return _photo;
}
public String setPhoto(String image) throws Exception
{
_photo=image;
return image;
}
public Employee(String firstName,String lastName,String title,String
address,String city,String region,String country,String photoFilePath) throws
Exception
{
setFirstName(firstName);
setLastName(lastName);
setTitle(title);
setAddress(address);
setCity(city);
setRegion(region);
setCountry(country);
setPhoto((photoFilePath));
}
}
```

The resultant document looks as follows.

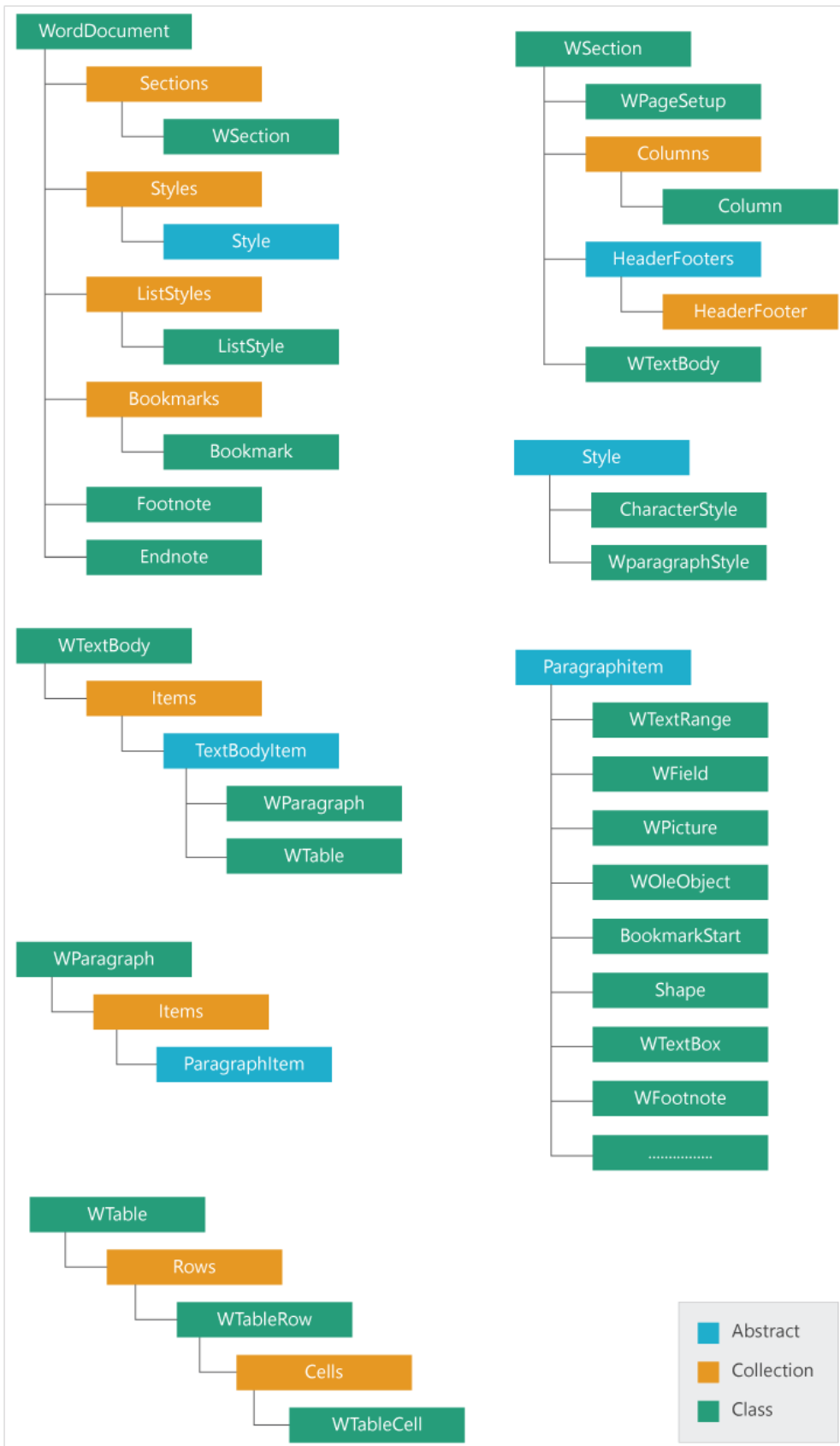




### Document Object Model representation

When an existing document is opened or a new document is created, the Java Word library creates a **Document Object Model (DOM)** of the document in main memory. This object model can be used to manipulate the document as needed.







## Loading & saving document

### Opening an existing document

You can open an existing Word document by using either the `open` method or the constructor of `WordDocument` class.

#### JAVA

```
//Open an existing document from the file system using the constructor of WordDocument class.  
WordDocument document = new WordDocument(fileName);
```

#### JAVA

```
//Create an empty Word document instance.  
WordDocument document = new WordDocument();  
//Load or open an existing word document using the open method of WordDocument class.  
document.open(fileName);
```

### Opening an existing document from Stream

You can open an existing document from the stream by using either the overload of `open` methods or the constructor of `WordDocument` class.

#### JAVA

```
//Open an existing document from the stream using the constructor of the WordDocument class.  
FileInputStream fileStreamPath = new FileInputStream("Input.docx");  
WordDocument document = new WordDocument(fileStreamPath,  
FormatTypeAutomatic);
```

#### JAVA

```
//Create an empty WordDocument instance.  
FileInputStream fileStreamPath = new FileInputStream("Input.docx");  
//Loads or opens an existing Word document using the open method of WordDocument class  
WordDocument document = new WordDocument();  
document.open(fileStreamPath, FormatTypeAutomatic);
```

### Opening the read only Word document

You can open the ready only documents or read-only streams using the `openReadOnly` method. If the Word document for reading is opened by any other application such as Microsoft Word, then the same document can be opened using the DocIO in ReadOnly mode. The following code sample explains the same.

#### JAVA

```
//Create an empty WordDocument instance.  
WordDocument document = new WordDocument();  
//Load or open an existing word document using the read only stream.  
document.openReadOnly("Template.docx", FormatType.Docx);
```



### Saving a Word document to file system

You can save the created or manipulated Word document to the file system using the `save` method of `WordDocument` class.

#### JAVA

```
//Create an empty WordDocument instance.
WordDocument document = new WordDocument();
//open an existing Word document using the Open method of WordDocument class.
document.open(fileName);
//To-Do some manipulation
//To-Do some manipulation
//Save the document in the file system.
document.save(outputFileName, FormatType.Docx);
```

### Saving a Word document to Stream

You can also save the created or manipulated word document to the stream by using the overloads of `save` methods.

#### JAVA

```
//Creates an empty WordDocument instance
WordDocument document = new WordDocument();
//Opens an existing Word document through Open method of WordDocument class.
document.open(fileName);
//To-Do some manipulation
//To-Do some manipulation
//Create an instance of the output stream.
ByteArrayOutputStream stream = new ByteArrayOutputStream();
//Save the document to stream.
document.save(stream, FormatType.Docx);
```

### Sending to a client browser

You can save and send the document to a client browser from a web site or web application by invoking the following shown overload of `save` method. This method explicitly makes use of an instance of [HttpResponse](#) as its parameter to stream the document to the client browser. So, this overload is suitable for a web application that references System.Web assembly.

#### JAVA

```
//Create an empty WordDocument instance.
WordDocument document = new WordDocument();
//Open an existing Word document using the Open method of WordDocument class.
document.open(fileName);
//To-Do some manipulation.
//To-Do some manipulation.
//Create an instance of the output stream.
ByteArrayOutputStream stream = new ByteArrayOutputStream();
//Save the document to the stream.
document.save(outputFileName, FormatType.Docx, Response,
HttpContentDisposition.Attachment);
```



### Closing a document

Once the document manipulation and save operation are completed, you should close the instance of **WordDocument**, in order to release all the memory consumed by the DocIO's DOM. The following code example shows how to close a **WordDocument** instance.

#### JAVA

```
//Create an empty WordDocument instance.
WordDocument document = new WordDocument();
//Opens an existing Word document using the Open method of WordDocument
class
document.Open(fileName);
//To-Do some manipulation.
//To-Do some manipulation.
//Create an instance of the output stream.
ByteArrayOutputStream stream = new ByteArrayOutputStream();
//Save the document to the stream.
document.save(stream, FormatType.Docx);
//Closes the document.
document.close();
```

### Working with Word document

#### Iterating through document elements

The following are the important points to be remembered while iterating the document elements

- Document consists of one or more sections.
- Section contains the contents present in Headers, Footers and main document through the instances of **WTextBody**.
- **WTextBody** contains three type of elements – either paragraph, table or block content control.

The following code example shows how to iterate throughout the Word document and remove the paragraph with a particular style.

#### JAVA

```
//Open an existing document from file system through constructor of
WordDocument class.
WordDocument document = new WordDocument("TestDocument.docx");
//Process the body contents for each section in the Word document.
for(Object section_tempObj : document.getSections())
{
    //Access the Body of section where all the contents in document are apart.
    WSection section = (WSection)section_tempObj;
    WTextBody sectionBody = section.getBody();
    iterateTextBody(sectionBody);
    //Consider that OddHeader and OddFooter are applied to this document.
    //Iterate through the TextBody of OddHeader and OddFooter.
    WHeadersFooters headersFooters = section.getHeadersFooters();
    iterateTextBody(headersFooters.getOddHeader());
    iterateTextBody(headersFooters.getOddFooter());
}
//Save and close the document instance.
document.save("Result.docx");
document.close();
```



The following code example provides supporting methods for the above code.

#### **JAVA**

```
private void iterateTextBody(WTextBody textBody) throws Exception
{
    //Iterate through each of the child items of WTextBody.
    for(int i = 0;i<textBody.getChildEntities().getCount();i++)
    {
        //IEntity is the basic unit in DocIO DOM.
        //Access the body items (should be either paragraph, table or block content
        control) as IEntity.
        IEntity bodyItemEntity = textBody.getChildEntities().get(i);
        //A Text body has 3 types of elements - Paragraph, Table and Block Content
        Control.
        //Decide the element type by using EntityType.
        switch(bodyItemEntity.getEntityType().toString())
        {
            case "Paragraph":
                //Check for particular style name and removes the paragraph from DOM.
                WParagraph paragraph = (WParagraph)bodyItemEntity;
                if(paragraph.getStyleName().equals("MyStyle"))
                {
                    int index = textBody.getChildEntities().indexOf(paragraph);
                    textBody.getChildEntities().removeAt(index);
                }
                break;
            case "Table":
                //Table is a collection of rows and cells.
                //Iterate through table's DOM.
                iterateTable((WTable)bodyItemEntity);
                break;
            case "BlockContentControl":
                //Iterate to the body items of Block Content Control.
                BlockContentControl blockContentControl =
                (BlockContentControl)bodyItemEntity;
                iterateTextBody(blockContentControl.getTextBody());
                break;
        }
    }
}
```

The following code example provides supporting methods for the above code.

#### **JAVA**

```
private void iterateTable(WTable table) throws Exception
{
    //Iterate the row collection in a table.
    for(Object row_tempObj : table.getRows())
    {
        //Iterate the cell collection in a table row.
        WTableRow row = (WTableRow)row_tempObj;
        for(Object cell_tempObj : row.getCells())
        {
            //Iterate the cell collection in a table row.
        }
    }
}
```



```
//Table cell is derived from (also a) TextBody.
//Reusing the code meant for iterating TextBody.
WTableCell cell = (WTableCell)cell_tempObj;
iterateTextBody(cell);
}
}
}
```

The following code example shows how to iterate throughout the paragraph and modify the hyperlink (Hyperlink)Uri and specific text (WTextRange)with another.

#### **JAVA**

```
//Open an existing document from file system through constructor of
WordDocument class.
WordDocument document = new WordDocument("TestDocument.docx");
//Process the body contents for each section in the Word document.
for(Object section_tempObj : document.getSections())
{
    //Access the Body of section where all the contents in document are apart.
    WSection section = (WSection)section_tempObj;
    WTextBody sectionBody = section.getBody();
    iterateTextBody(sectionBody);
    WHeadersFooters headersFooters = section.getHeadersFooters();
    //Consider that OddHeader & OddFooter are applied to this document.
    //Iterate through the TextBody of OddHeader and OddFooter.
    iterateTextBody(headersFooters.getOddHeader());
    iterateTextBody(headersFooters.getOddFooter());
}
//Save and close the document instance.
document.save("Result.docx");
document.close();
```

The following code example provides supporting methods for the above code.

#### **JAVA**

```
private void iterateTextBody(WTextBody textBody) throws Exception
{
    //Iterate through each of the child items of WTextBody.
    for(int i = 0;i<textBody.getChildEntities().getCount();i++)
    {
        //IEntity is the basic unit in DocIO DOM.
        //Access the body items (should be either paragraph, table or block content
        control) as IEntity.
        IEntity bodyItemEntity = textBody.getChildEntities().get(i);
        //A Text body has 3 types of elements - Paragraph, Table and Block Content
        Control.
        //Decide the element type by using EntityType.
        switch(bodyItemEntity.getEntityType().toString())
        {
            case "Paragraph":
                //Process the paragraph contents.
                //Iterate through the paragraph's DOM.
                WParagraph paragraph = (WParagraph)bodyItemEntity;
                terateParagraph(paragraph.getItems());
```



```

break;
case "Table":
    //Table is a collection of rows and cells.
    //Iterate through table's DOM.
    iterateTable((WTable)bodyItemEntity);
break;
case "BlockContentControl":
    //Iterate to the body items of Block Content Control.
    BlockContentControl blockContentControl =
        (BlockContentControl)bodyItemEntity;
    iterateTextBody(blockContentControl.getTextBody());
break;
}
}
}

```

The following code example provides supporting methods for the above code.

### JAVA

```

private void iterateTable(WTable table) throws Exception
{
    //Iterate the row collection in a table.
    for(Object row_tempObj : table.getRows())
    {
        //Iterate the cell collection in a table row.
        WTableRow row = (WTableRow)row_tempObj;
        for(Object cell_tempObj : row.getCells())
        {
            //Table cell is derived from (also a) TextBody.
            //Reusing the code meant for iterating TextBody.
            WTableCell cell = (WTableCell)cell_tempObj;
            iterateTextBody(cell);
        }
    }
}

```

The following code example provides supporting methods for the above code.

### JAVA

```

private void iterateParagraph(ParagraphItemCollection paraItems) throws
Exception
{
    for(int i = 0;i<paraItems.getCount();i++)
    {
        //A paragraph can have child elements such as text, image, hyperlink,
        symbols, etc.,.
        //Decide the element type by using EntityType.
        Entity entity = paraItems.get(i);
        switch(entity.getEntityType().toString())
        {
            case "TextRange":
                //Replace the text with another.
                WTextRange textRange = (WTextRange)entity;
                if(textRange.getText().equals("Andrew"))

```



```

{
    ((WTextRange)entity).setText("Fuller");
}
break;
case "Field":
    //Create hyperlink instance from field to manipulate the hyperlink.
    WField field = (WField)entity;
    //Modify the Uri of the hyperlink
    if(field.getFieldType().getEnumValue()==FieldType.FieldHyperlink.getEnumValue())
    {
        Hyperlink hyperlink = new Hyperlink((WField)entity);
        if(hyperlink.getType().getEnumValue()==HyperlinkType.WebLink.getEnumValue()
        && hyperlink.getTextToDisplay().equals("HTML"))
        {
            hyperlink.setUri("http://www.w3schools.com/");
        }
    }
    break;
case "TextBox":
    //Iterate to the body items of textbox.
    WTextBox textBox = (WTextBox)entity;
    iterateTextBody(textBox.getTextBoxBody());
    break;
case "Shape":
    //Iterate to the body items of shape.
    Shape shape = (Shape)entity;
    iterateTextBody(shape.getTextBody());
    break;
case "InlineContentControl":
    //Iterate to the paragraph items of inline content control.
    InlineContentControl inlineContentControl = (InlineContentControl)entity;
    iterateParagraph(inlineContentControl.getParagraphItems());
    break;
}
}
}

```

### Cloning a Word document

You can create a deep copy of a Word document by using `clone` method of `WordDocument` class. You can read the template document from file system or stream and create multiple document copies by cloning it. This improves the performance of document generation, as there is no need to read the Word document each time.

### JAVA

```

//Open an existing document.
WordDocument inputTemplateDoc = new WordDocument(fileName);
//Create a clone of Input Template.
WordDocument clonedDocument = inputTemplateDoc.clone();
//Save and close the cloned document instance.
clonedDocument.save("ClonedDocument.docx");
clonedDocument.close();
//Close the input template document instance.
inputTemplateDoc.close();

```



You can also create a deep copy of document elements such as sections, paragraphs, Tables, Text, Image, Shapes, TextBoxes and etc., The following code example illustrates how to clone the section and save each cloned section as a Word document.

#### **JAVA**

```
//Open a source document.
WordDocument sourceDocument = new WordDocument("SourceDocument.docx");
//Process the each section in the Word document.
for (int i = 0; i < sourceDocument.getSections().getCount(); i++)
{
    //Create new WordDocument instance to add cloned section.
    WordDocument destinationDocument = new WordDocument();
    //Clone and adds source document sections to the destination document.
    destinationDocument.getSections().add(sourceDocument.getSections().get(i).clone());
    //Save and close the document instance.
    destinationDocument.save("Section_" + i + ".docx");
    destinationDocument.close();
}
//Close the source document instance.
sourceDocument.close();
```

#### Merging Word documents

You can merge multiple Word documents into single Word document by using DocIO's capability of importing contents from one document to another. The imported contents are appended at the end of document. The following code example illustrates how to import the contents from source document into destination document where the contents are appended.

#### **JAVA**

```
//Open the source document.
WordDocument sourceDocument = new WordDocument(sourceFileName);
//Open the destination document.
WordDocument destinationDocument = new WordDocument(targetFileName);
//Import the contents of source document at the end of destination document.
destinationDocument.importContent(sourceDocument,
ImportOptions.UseDestinationStyles);
//Save the destination document.
destinationDocument.save(outputFileName, FormatType.Docx);
//close the document instances.
sourceDocument.close();
destinationDocument.close();
```

In the resultant document, the imported contents start from a new page followed by existing contents in a destination document. This is the default behavior.

When your requirement is to append the contents from the same page instead of starting from a new page, you need to set the break code of first section of Source document as NoBreak. The following code example illustrates the importing contents from the same page.

#### **JAVA**

```
//Open the source document.
```



```

WordDocument sourceDocument = new WordDocument(sourceFileName);
//Open the destination document.
WordDocument destinationDocument = new WordDocument(targetFileName);
//Set the break-code of First section of source document as NoBreak to avoid
imported from a new page.
sourceDocument.getSections().get(0).setBreakCode(SectionBreakCode.NoBreak);
//Import the contents of source document at the end of destination document.
destinationDocument.importContent(sourceDocument,
ImportOptions.UseDestinationStyles);
//Save the destination document.
destinationDocument.save(outputFileName, FormatType.Docx);
//Close the document instances.
sourceDocument.close();
destinationDocument.close();

```

### Maintain Imported List style information

The following code example shows how to maintain information about imported list styles in a Word document while cloning and merging multiple Word documents.

### JAVA

```

//Open the source document.
WordDocument sourceDocument = new WordDocument(sourceFileName);
//Open the destination document.
WordDocument destinationDocument = new WordDocument(targetFileName);
//Set true value to maintain imported list style cache to destination
document.
destinationDocument.getSettings().setMaintainImportedListCache(true);
//Process the body contents for each section in the Word document.
for(Object section_tempObj : sourceDocument.getSections())
{
//Access the body of section where all the contents in document are apart.
WSection section = (WSection)section_tempObj;
for(Object bodyItem_tempObj : section.getBody().getChildEntities())
{
TextBodyItem bodyItem = (TextBodyItem)bodyItem_tempObj;
destinationDocument.getLastSection().getBody().getChildEntities().add(bodyItem.clone());
}
}
//Close the source document.
sourceDocument.close();
//Set false value to exclude imported list style cache to destination
document.
destinationDocument.getSettings().setMaintainImportedListCache(false);
//Save the destination document.
destinationDocument.save(outputFileName, FormatType.Docx);
//Close the destination document.
destinationDocument.close();

```

### Working with Styles

A style is a predefined set of table, numbering, paragraph, and character properties that can be applied to regions within a document. DocIO provides the following functionalities related with styles.



- Access and modify the existing styles in the word document
- Create new paragraph style.
- Apply built-in styles.

### Access Styles

Paragraph and character styles present in the existing document are accessible through the **Styles** property of **WordDocument** class.

This following code example demonstrates how a style can be accessed and style properties like text color and first line indent can be updated.

#### JAVA

```
//Open an input Word template.
WordDocument document = new WordDocument(inputFileName);
//Access the styles collection that contains paragraph and character styles
in Word document.
IStyleCollection styleCollection = document.getStyles();
//Find the style with the name "Heading 1".
WParagraphStyle heading1ParagraphStyle =
(WParagraphStyle)styleCollection.findByName("Heading 1");
//Change the text color of style "Heading 1" as DarkBlue.
heading1ParagraphStyle.getCharacterFormat().setTextColor(ColorSupport.getDarkBlue());
//Change the first line indent of Paragraph as 36 points.
heading1ParagraphStyle.getParagraphFormat().setFirstLineIndent(36);
//Save and close the document instance.
document.save(outputFileName, FormatType.Docx);
document.close();
```

### Creating a new Paragraph Style

You can create a new paragraph style by using **WordDocument.addParagraphStyle** method and apply it by using **applyStyle** method of **WParagraph** class.

#### JAVA

```
//Open an input Word template.
WordDocument document = new WordDocument();
//This method adds a section and a paragraph in the document.
document.ensureMinimal();
//Add a new paragraph style named "MyStyle".
IWParagraphStyle myStyle = document.addParagraphStyle("MyStyle");
//Set the formatting of the style.
myStyle.getCharacterFormat().setFontSize(16f);
myStyle.getCharacterFormat().setTextColor(ColorSupport.getDarkBlue());
myStyle.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Right);
//Append the contents into the paragraph.
document.getLastParagraph().appendText("AdventureWorks Cycles, the fictitious company on which the AdventureWorks sample databases are based, is a large, multinational manufacturing company.");
//Apply the style to paragraph.
document.getLastParagraph().applyStyle("MyStyle");
//Save and close the document instance.
document.save(outputFileName, FormatType.Docx);
```



```
document.close();
```

### Applying built-in styles

DocIO provides a set of predefined styles. You can apply those predefined styles as shown in the following code example.

#### JAVA

```
//Open an input Word template.
WordDocument document = new WordDocument();
//This method adds a section and a paragraph in the document.
document.ensureMinimal();
IWParagraph paragraph = document.getLastParagraph();
//Append the content into the paragraph.
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.");
//Apply the style to paragraph.
paragraph.applyStyle(BuiltinStyle.Emphasis);
//Save and close the document instance.
document.save(outputFileName, FormatType.Docx);
document.close();
```

### Working with Word document properties

Document properties, also known as metadata, are details about a file that describe or identify it. You can also define the additional custom document properties for the documents by using DocIO Document properties that are classified as two categories.

- Built-in document properties - includes details such as title, author name, subject, and keywords that identify the document's topic or contents.
- Custom document properties - defines the user-defined document properties.

### Built-in document properties

The Built-in document properties of a word document is represented by `BuiltinDocumentProperties` property of `WordDocument` class. The following code example illustrates how to access and modify the Built-in document properties of the document.

#### JAVA

```
//Open an existing Word document.
WordDocument document = new WordDocument(inputFileName);
//Access the built-in document properties.
System.out.println(StringSupport.format("Title -
{0}", document.getBuiltinDocumentProperties().getTitle()));
System.out.println(StringSupport.format("Author -
{0}", document.getBuiltinDocumentProperties().getAuthor()));
//Modify or set the category and company Built-in document properties.
document.getBuiltinDocumentProperties().setCategory("Sales reports");
document.getBuiltinDocumentProperties().setCompany("Northwind traders");
document.save("outputFileName", FormatType.Docx);
document.close();
```



### *Adding Custom Document properties*

You add a new custom document properties through `add` method of `CustomProperties` class. The following code example illustrates how to add a new custom document properties.

#### **JAVA**

```
//Open an input word template.
WordDocument document = new WordDocument(inputFileName);
//Add the custom document properties of various data types.
document.getCustomDocumentProperties().add("PropertyA", "Value of A");
document.getCustomDocumentProperties().add("PropertyB", 12.5);
document.getCustomDocumentProperties().add("PropertyC", true);
document.getCustomDocumentProperties().add("PropertyD",
LocalDateTime.of(2015, 7, 20, 0, 0));
//Save and close the document.
document.save(outputFileName, FormatType.Docx);
document.close();
```

### *Accessing & Modifying Custom Document Properties*

You can access and modify an existing document property as shown in the following code example.

#### **JAVA**

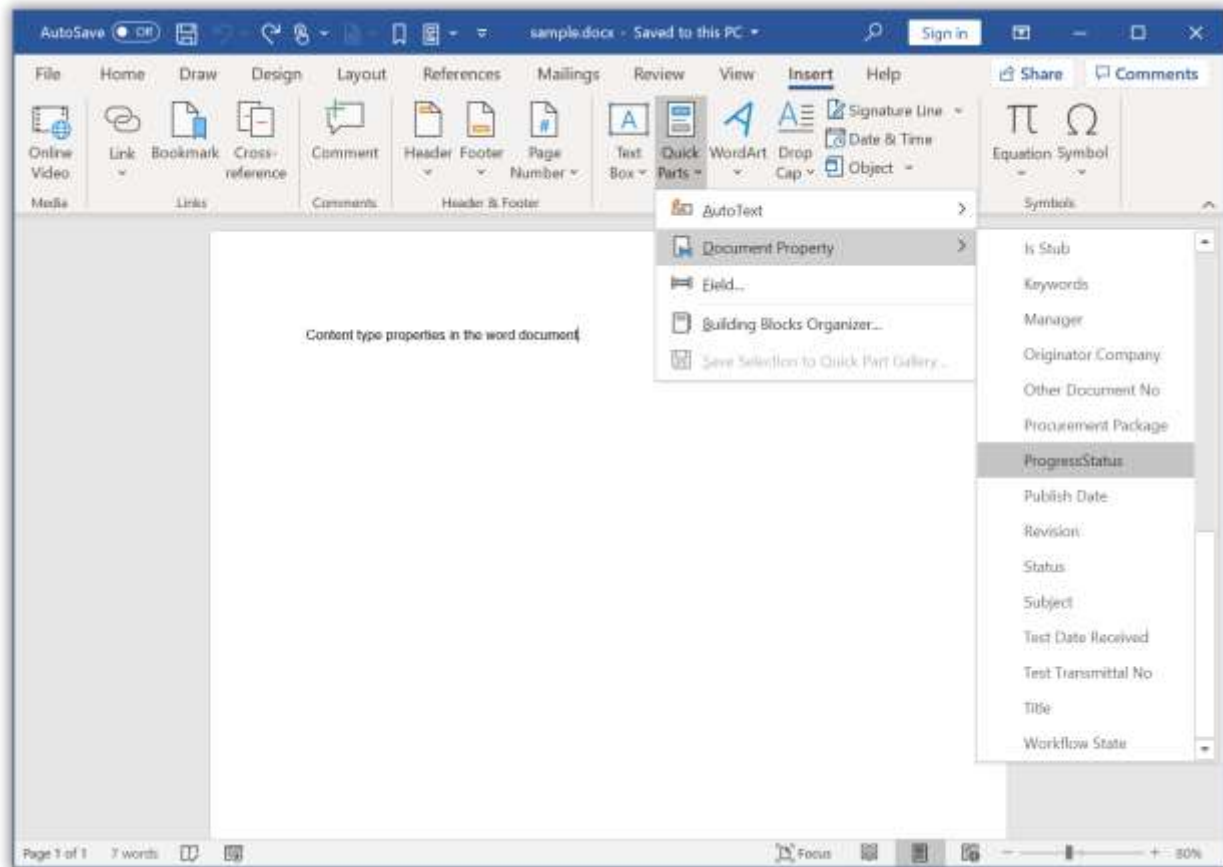
```
//Open an input word template.
WordDocument document = new WordDocument(inputFileName);
//Accesses an existing custom document property
DocumentProperty property =
document.getCustomDocumentProperties().get("PropertyA");
//Modifies the value of DocumentProperty instance
property.setValue("Hello world");
//Save and close the document.
document.save(outputFileName, FormatType.Docx);
document.close();
```

### *Working with Content Type Properties*

Content type properties refers the metadata stored in a Word document, such as author name, subject, and company. `DocIO` represents metadata with `MetaProperty` instance and you can access in the Word document (DOCX, WordML) by using the `ContentTypeProperties` collection of `WordDocument` class.

The following screenshots shows the content type property in the input Word document.





**Note:** You can use Content Type Properties only in documents that are saved in the DOCX or WordML Format.

#### *Accessing and modifying the Content Type Properties*

You can access and modify the value of existing metadata in the Word document (DOCX, WordML).

The following code example explains how to access and modify the value of an existing metadata in the Word document.

#### **JAVA**

```
// Loads the template document
WordDocument document = new WordDocument("Template.docx");
// Processes the metaproperty collection in the Word document
MetaProperties metaProperties = document.getContentProperties();
// Iterates through each of the child items of metaproperties
for (int i = 0; i < metaProperties.getCount(); i++)
{
    // Checks for particular display name of meta data and modifies its value
    switch (metaProperties.get(i).getDisplayname())
    {
        case "ProgressStatus":
            if (metaProperties.get(i).getType() == MetaPropertyType.Text &&
                !metaProperties.get(i).getIsReadOnly())
                metaProperties.get(i).setValue("Completed");
            break;
        case "Reviewed":
```



```

if (metaProperties.get(i).getType() == MetaPropertyType.Boolean &&
!metaProperties.get(i).getIsReadOnly())
metaProperties.get(i).setValue(true);
break;
case "Date":
if (metaProperties.get(i).getType() == MetaPropertyType.DateTime &&
!metaProperties.get(i).getIsReadOnly())
metaProperties.get(i).setValue(LocalDate.now(ZoneId.of("UTC")));
break;
case "Salary":
if ((metaProperties.get(i).getType() == MetaPropertyType.Number ||
metaProperties.get(i).getType() == MetaPropertyType.Currency) &&
!metaProperties.get(i).getIsReadOnly())
metaProperties.get(i).setValue(12000);
break;
case "Url":
if (metaProperties.get(i).getType() == MetaPropertyType.Url &&
!metaProperties.get(i).getIsReadOnly())
{
String[] value = { "https://www.syncfusion.com", "Syncfusion page" };
metaProperties.get(i).setValue(value);
}
break;
case "User":
if (metaProperties.get(i).getType() == MetaPropertyType.User &&
!metaProperties.get(i).getIsReadOnly())
{
String[] value = { "1234", "Syncfusion" };
metaProperties.get(i).setValue(value);
}
break;
default:
break;
}
}
// Saves the Word document
document.save("Sample.docx", FormatType.Docx);
// Closes the document
document.close();

```

### Setting the Background for a Word document

Essential DocIO allows to apply background such as color, gradient and picture to the Word document. A background of a Word document is represented by **Background** property of 'WordDocument' class.

The following code illustrates how to apply gradient as background to the document.

#### JAVA

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
WSection section = (WSection) document.addSection();
//Add new paragraph to the section
IWParagraph paragraph = section.addParagraph();
//Append text to the paragraph.
paragraph.appendText("Sample for applying document background");

```



```
//Set the background type as gradient.
document.getBackground().setType(BackgroundType.Gradient);
//Set color for gradient.
document.getBackground().getGradient().setColor1(ColorSupport.getLightGray());
document.getBackground().getGradient().setColor2(ColorSupport.getLightGreen());
//Set the shading style.
document.getBackground().getGradient().setShadingStyle(GradientShadingStyle.DiagonalUp);
document.getBackground().getGradient().setShadingVariant(GradientShadingVariant.ShadingDown);
//Save the document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code illustrates how to apply image as background for the document.

### JAVA

```
//Create a new Word document
WordDocument document = new WordDocument();
//Add new section to the document
IWSection section = document.addSection();
//Add new paragraph to the section
IWParagraph paragraph = section.addParagraph();
//Append text to the paragraph
paragraph.appendText("Sample for applying document background");
//Set the background type as picture
document.getBackground().setType(BackgroundType.Picture);
MemoryStreamSupport inputStream = new MemoryStreamSupport(
new FileInputStream(new File("Image.png")));
document.getBackground().setPicture(inputStream.toArray());
//Save the document
document.save("Sample.docx", FormatType.Docx);
//Close the document
document.close();
```

### Working with Alternate chunks

Updating Alternate chunk in the Word document, imports the content from the embedded alternate chunk into the main document. When saving the Word document containing alternate chunk as DOCX format document, the alternate chunk content preserved by default. But, when saving as DOC format or other formats, the alternate chunk content will not be preserved. You can use `updateAlternateChunks` method to preserve the alternate chunk content by importing into the main document.

The following examples show how to update the alternate chunk in the word document.

### JAVA

```
//Open an existing document from file system through constructor of
WordDocument class.
try(WordDocument document = new WordDocument("Sample.docx", FormatType.Docx))
{
//Update the alternate chunks in the document
```



```
document.updateAlternateChunks();
//Save and closes the document instance
document.save("Result.docx");
}
```

## Working with Sections

A section contains the contents present in the headers, footers, and the main document using the instances of `WTextBody`. A section also has a specific set of properties used to define the page settings, a number of columns, headers, and footers, and more that decide how the text appears. The `WTextBody` represents a group of paragraphs, tables, and more.

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Append the text to the created paragraph.
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

You can add the multiple sections to the document. When you add more than one section into the word document, the section starts from the next page by default.

You can also add a new section that starts on the same page by specifying the `BreakCode` as shown in the following code example.

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a paragraph to the created section.
IWParagraph paragraph = section.addParagraph();
String paraText = "AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.";
//Append the text to the created paragraph.
paragraph.appendText(paraText);
//Add the new section to the document.
section = document.addSection();
//Set a section break.
section.setBreakCode(SectionBreakCode.NoBreak);
//Add a paragraph to the created section.
paragraph = section.addParagraph();
//Append the text to the created paragraph.
paragraph.appendText(paraText);
//Save and close the Word document instance.
```



```
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

### Specifying Page Properties

Each section has its page setup properties such as page size, orientation, margins, borders, and more.

The following code example shows how to set the page setup properties.

#### JAVA

```
//Create a new Word document.  
WordDocument document = new WordDocument();  
//Add the section into the Word document.  
IWSection section = document.addSection();  
//Set the page setup options.  
section.getPageSetup().setOrientation(PageOrientation.Landscape);  
section.getPageSetup().getMargins().setAll(72);  
section.getPageSetup().getBorders().setLineWidth(2);  
//Add a paragraph to the created section.  
IWParagraph paragraph = section.addParagraph();  
//Append the text to the created paragraph.  
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which  
the AdventureWorks sample databases are based, is a large, multinational  
manufacturing company.");  
//Save and close the Word document instance.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

### Creating Multi-column document

You can split the contents into two or more columns by specifying the column width and spacing between columns.

The following code example shows how to display the contents in multiple columns.

#### JAVA

```
//Create a new Word document.  
WordDocument document = new WordDocument();  
//Add the section into the Word document.  
IWSection section = document.addSection();  
//Add the column into the section.  
section.addColumn(150, 20);  
//Add the column into the section.  
section.addColumn(150, 20);  
//Add the column into the section.  
section.addColumn(150, 20);  
//Add a paragraph to the created section.  
IWParagraph paragraph = section.addParagraph();  
//Add a paragraph to the created section.  
paragraph = section.addParagraph();  
String paraText = "AdventureWorks Cycles, the fictitious company on which  
the AdventureWorks sample databases are based, is a large, multinational  
manufacturing company.";  
//Append the text to the created paragraph.  
paragraph.appendText(paraText);  
//Add the column break.
```



```

paragraph.appendBreak(BreakType.ColumnBreak);
//Add a paragraph to the created section.
paragraph = section.addParagraph();
//Append the text to the created paragraph.
paragraph.appendText(paraText);
//Add the column break.
paragraph.appendBreak(BreakType.ColumnBreak);
//Add a paragraph to the created section.
paragraph = section.addParagraph();
//Append the text to the created paragraph.
paragraph.appendText(paraText);
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

### Creating document with different page settings

You can prefer to have more sections in a Word document when you need to have different page settings or headers and footers for a specific set of contents. The following code example shows how to create a Word document with the multiple sections whose page orientation is portrait and landscape respectively.

#### JAVA

```

//Creates a new Word document.
WordDocument document = new WordDocument();
//Add the section into the Word document.
IWSection section = document.addSection();
//Add a paragraph to the created section.
IWParagraph paragraph = section.addParagraph();
String paraText = "AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.";
//Append the text to the created paragraph.
paragraph.appendText(paraText);
//Set the page orientation as a portrait.
section.getPageSetup().setOrientation(PageOrientation.Portrait);
//Add the new section to the document.
section = document.addSection();
//Set the section break.
section.setBreakCode(SectionBreakCode.NewPage);
paragraph = section.addParagraph();
//Set the page orientation as a landscape
section.getPageSetup().setOrientation(PageOrientation.Landscape);
//Append the text to the paragraph.
paragraph.appendText(paraText);
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

### Working with Headers and Footers

The header and footer also represent the group of paragraphs and tables that occur at the top and bottom of the page respectively. The header and footer may vary for each section. The following are the types of Headers or Footers:



- FirstPageHeader: Represents the first-page header of the document.
- FirstPageFooter: Represents the first-page footer of the document.
- OddHeader: Represents the odd page header of the document and it is the default header for the section.
- OddFooter: Represents the odd page footer of the document and it is the default footer for the section.
- EvenHeader: Represents the even page header of the document.
- Even Footer: Represents the even page footer of the document.

The following code example illustrates how to add simple header and footer into a Word document.

#### **JAVA**

```
//Create a new document.
WordDocument document = new WordDocument();
//Add the first section to the document.
IWSection section = document.addSection();
//Add a paragraph to the section.
IWParagraph paragraph = section.addParagraph();
String paraText = "AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.";
//Append some text to the first page of the document.
paragraph.appendText("\r\r[ First Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);
//Append some text to the second page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Second Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);
//Append some text to the third page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Third Page ] \r\r" + paraText);
//Insert the default page header.
paragraph = section.getHeadersFooters().getOddHeader().addParagraph();
paragraph.appendText("[ Default Page Header ]");
//Insert the default page footer.
paragraph = section.getHeadersFooters().getOddHeader().addParagraph();
paragraph.appendText("[ Default Page Footer ]");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

You can have a specific header and footer contents for the first page in a Word document. The following code illustrates the same.

#### **JAVA**

```
//Create a new document.
WordDocument document = new WordDocument();
//Add the first section to the document.
IWSection section = document.addSection();
//Set the DifferentFirstPage as a true for inserting the header and footer
text.
section.getPageSetup().setDifferentFirstPage(true);
//Add a paragraph to the section.
```



```

IWParagraph paragraph = section.addParagraph();
String paraText = "AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.";
//Append some text to the first page of the document.
paragraph.appendText("\r\r[ First Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);
//Append some text to the second page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Second Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);
//Append some text to the third page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Third Page ] \r\r" + paraText);
//Insert the first page header.
paragraph = section.getHeadersFooters().getFirstPageHeader().addParagraph();
paragraph.appendText("[First Page Header ]");
//Insert the first page footer.
paragraph = section.getHeadersFooters().getFirstPageFooter().addParagraph();
paragraph.appendText("[ First Page Footer ]");
//Insert the default page header.
paragraph = section.getHeadersFooters().getOddHeader().addParagraph();
paragraph.appendText("[ Default Page Header ]");
//Insert the default page footer.
paragraph = section.getHeadersFooters().getOddFooter().addParagraph();
paragraph.appendText("[ Default Page Footer ]");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

A word document can have a different header and footer for the odd and even pages.

The following code example shows how to set a different header and footer for the odd and even pages of the document.

## JAVA

```

//Create a new document.
WordDocument document = new WordDocument();
//Add the first section to the document.
IWSection section = document.addSection();
//Set the DifferentOddAndEvenPages to true for inserting the header and
//footer text.
section.getPageSetup().setDifferentOddAndEvenPages(true);
//Add a paragraph to the section.
IWParagraph paragraph = section.addParagraph();
String paraText = "AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.";
//Append some text to the first page of the document.
paragraph.appendText("\r\r[ First Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);
//Append some text to the second page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Second Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);

```



```
//Append some text to the third page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Third Page ] \r\r" + paraText);
//Insert the odd page header.
paragraph = section.getHeadersFooters().getOddHeader().addParagraph();
paragraph.appendText("[ Odd Page Header ]");
//Insert the default page footer.
paragraph = section.getHeadersFooters().getOddFooter().addParagraph();
paragraph.appendText("[ Odd Page Footer ]");
//Insert the even page header.
paragraph = section.getHeadersFooters().getEvenHeader().addParagraph();
paragraph.appendText("[Even Page Header ]");
//Insert the even page footer.
paragraph = section.getHeadersFooters().getEvenFooter().addParagraph();
paragraph.appendText("[ Even Page Footer ]");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

You can use the previous section header and footer for the current section by using the `LinkToPrevious` property.

The following code example shows how to link the previous section header and footer for the current section.

#### **JAVA**

```
//Create a new document.
WordDocument document = new WordDocument();
//Add the first section to the document.
IWSection section = document.addSection();
//Insert the first section header.
section.getHeadersFooters().getHeader().addParagraph().appendText("[ First Section Header ]");
//Insert the first section footer.
section.getHeadersFooters().getFooter().addParagraph().appendText("[ First Section Footer ]");
//Add a paragraph to the section.
IWParagraph paragraph = section.addParagraph();
String paraText = "AdventureWorks Cycles, the fictitious company on which the AdventureWorks sample databases are based, is a large, multinational manufacturing company.";
//Append some text to the first page of the document.
paragraph.appendText("\r\r[ First Page ] \r\r" + paraText);
//Add the second section to the document.
section = document.addSection();
//Insert the second section header.
section.getHeadersFooters().getHeader().addParagraph().appendText("[ Second Section Header ]");
//Insert the second section footer.
section.getHeadersFooters().getFooter().addParagraph().appendText("[ Second Section Footer ]");
//Set the LinkToPrevious to true for retrieve the header and footer from the previous section.
section.getHeadersFooters().setLinkToPrevious(true);
//Append some text to the second page of the document.
```



```

paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Second Page ] \r\r" + paraText);
//Add the third section to the document.
section = document.addSection();
//Insert the third section header.
section.getHeadersFooters().getHeader().addParagraph().appendText("[ Third
Section Header ]");
//Insert the third section footer.
section.getHeadersFooters().getFooter().addParagraph().appendText("[ Third
Section Footer ]");
//Append some text to the third page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Third Page ] \r\r" + paraText);
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

### Adding Page Numbers

You can insert the current page number within the document contents. The following code example shows how to insert the current page number within the footer.

#### **JAVA**

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add the section into a Word document.
IWSection section = document.addSection();
section.getPageSetup().setPageStartingNumber(1);
section.getPageSetup().setRestartPageNumbering(true);
section.getPageSetup().setPageNumberStyle(PageNumberStyle.Arabic);
//Add a footer paragraph text to the document.
IWParagraph paragraph =
section.getHeadersFooters().getFooter().addParagraph();
paragraph.getParagraphFormat().getTabs().addTab(523f,
TabJustification.Right, TabLeader.NoLeader);
//Add text for the footer paragraph.
paragraph.appendText("Copyright Northwind Inc. 2001 - 2015");
//Add the page number field to the document.
paragraph.appendText("\tPage ");
paragraph.appendField("Page", FieldType.FieldPage);
//Add the paragraph.
paragraph = section.addParagraph();
//Append the text to the created paragraph.
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

The following code example shows how to add the current page number and the total number of pages in the header or footer.

#### **JAVA**



```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add the section into the Word document.
IWSection section = document.addSection();
section.getPageSetup().setPageStartingNumber(1);
section.getPageSetup().setRestartPageNumbering(true);
section.getPageSetup().setPageNumberStyle(PageNumberStyle.Arabic);
//Add a footer paragraph text to the document.
IWParagraph paragraph =
section.getHeadersFooters().getFooter().addParagraph();
paragraph.getParagraphFormat().getTabs().addTab(523f,
TabJustification.Right, TabLeader.NoLeader);
// Add the text for the footer paragraph.
paragraph.appendText("Copyright Northwind Inc. 2001 - 2015\t");
//Add the text.
paragraph.appendText(" Page ");
//Add the page number field to the document.
paragraph.appendField("CurrentPageNumber", FieldType.FieldPage);
// Add the text.
paragraph.appendText(" of ");
//Add the number of pages field to the document.
paragraph.appendField("TotalNumberOfPages", FieldType.FieldNumPages);
//Add the paragraph.
paragraph = section.addParagraph();
//Append the text to the created paragraph.
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

The following code example shows how to adjust the height of the header and footer.

#### **JAVA**

```
//Create a new document.
WordDocument document = new WordDocument();
//Add the first section to the document.
IWSection section = document.addSection();
//Specify the value to the header distance.
section.getPageSetup().setHeaderDistance(100);
//Specify the value to the footer distance.
section.getPageSetup().setFooterDistance(100);
//Add a paragraph to the section.
IWParagraph paragraph = section.addParagraph();
String paraText = "AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.";
//Append some text to the first page of the document.
paragraph.appendText("\r\r[ First Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);
//Append some text to the second page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Second Page ] \r\r" + paraText);
paragraph.getParagraphFormat().setPageBreakAfter(true);
```



```
//Append some text to the third page of the document.
paragraph = section.addParagraph();
paragraph.appendText("\r\r[ Third Page ] \r\r" + paraText);
//Insert the default page header.
paragraph = section.getHeadersFooters().getOddHeader().addParagraph();
paragraph.appendText("[ Default Page Header ]");
//Insert the default page footer.
paragraph = section.getHeadersFooters().getOddFooter().addParagraph();
paragraph.appendText("[ Default Page Footer ]");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Removing a Section

The following code example shows how to remove a particular section from the word document.

#### **JAVA**

```
//Open an input Word template.
WordDocument document = new WordDocument("inputFileName");
//Remove the second section from the collection.
document.getSections().removeAt(1);
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Working with Paragraph

Paragraph is the basic element in a Word document that contains a textual and graphical contents. Each paragraph has its own formatting such as line spacing, alignment, indentation, and more. Within a paragraph, the contents are represented by one or more child elements such as **WTextRange**, **WPicture**, and **Hyperlink** and more. The **ParagraphItem** is the base class for the child elements of paragraph. The following elements can be the child elements of a paragraph.

- Text: Represented by an instance of **WTextRange**.
- Image: Represented by an instance of **WPicture**.
- Comments: Represented by an instance of **WComment**.
- Hyperlink: Represented by an instance of **Hyperlink**.
- Symbols: Represented by an instance of **WSymbol**.
- Breaks: Represented by an instance of **Break**.
- Shapes: Represented by an instance of **Shape**.
- TextBox: Represented by an instance of **WTextBox**.
- Fields: Represented by an instance of **WField**.
- Form Fields: Represented by an instance of **WFormField**.
- Bookmarks: Represented by instances of **BookmarkStart** and **BookmarkEnd**.
- Absolute Tab: Represented by an instance of **WAbsoluteTab**.

The following code example explains how to add a new paragraph.

#### **JAVA**



```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add a new text to the paragraph.
paragraph.appendText("Adding new paragraph to the document");
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example shows how to modify an existing paragraph.

### **JAVA**

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx");
//Get the text body of the first section.
WTextBody textBody = document.getSections().get(0).getBody();
//Get the paragraph at index 1.
WParagraph paragraph = textBody.getParagraphs().get(1);
//Iterate through the child elements of the paragraph.
for(Object item_tempObj : paragraph.getChildEntities())
{
    ParagraphItem item = (ParagraphItem)item_tempObj;
    if(item instanceof WTextRange)
    {
        WTextRange text = (WTextRange)(item);
        //Modify the character format of the text.
        text.getCharacterFormat().setBold(true);
        break;
    }
}
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

### Applying paragraph formatting

As in the Microsoft Word, the DocIO provides support for all the paragraph formatting options such as line spacing, indentation, spacing before and after, keep follow, and more. The following code example explains how to apply the formatting to a paragraph.

**Note:** The `FirstLineIndent` can be used to update or retrieve both hanging and first line indents. Negative value for this property denotes the hanging indent and positive value denotes the first line indent of the paragraph.

### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
```



```
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add a new text to the paragraph.
IWTextRange firstText = paragraph.appendText("Paragraphs are the basic
elements of the Word document. It can contain text and images.");
//Apply the paragraph formatting.
paragraph.getParagraphFormat().setAfterSpacing(18f);
paragraph.getParagraphFormat().setBeforeSpacing(18f);paragraph.getParagraphF
ormat().setBackColor(ColorSupport.getLightGray());
paragraph.getParagraphFormat().setFirstLineIndent(10f);
paragraph.getParagraphFormat().setLineSpacing(10f);paragraph.getParagraphFor
mat().setHorizontalAlignment(HorizontalAlignment.Right);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

### Paragraph style

The paragraph style contains a definition for both font (text) and paragraph formatting that can be applied to the contents of an entire paragraph. DocIO supports various pre-defined styles and also provides the ability to create custom paragraph styles.

**Tips:** You can define a custom style or modify any built-in style to the required formatting, and apply this style to the part of Word document to be formatted. You can reduce the file size and code length by using styles instead of formatting each element explicitly.

The following code example explains how to use the predefined styles.

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph firstParagraph = section.addParagraph();
//Add a new text to the paragraph.
IWTextRange firstText = firstParagraph.appendText("Built-in styles can be
applied to the paragraph. Heading1 style is applied to this paragraph.");
//Apply the built-in style for the paragraph.
firstParagraph.applyStyle(BuiltinStyle.Heading1);
//Save the word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

### Custom paragraph style

The following code example explains how to create a custom paragraph style and apply it to a paragraph.

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
```



```

IWSection section = document.addSection();
//Create an user defined style.
IWParagraphStyle style = document.addParagraphStyle("User_defined_style");
style.getParagraphFormat().setBackColor(ColorSupport.getLightGray());
style.getParagraphFormat().setAfterSpacing(18f);
style.getParagraphFormat().setBeforeSpacing(18f);
style.getParagraphFormat().getBorders().setBorderType(BorderStyle.DotDash);
style.getParagraphFormat().getBorders().setLineWidth(0.5f);
style.getParagraphFormat().setLineSpacing(15f);
style.getCharacterFormat().setFontName("Calibri");
style.getCharacterFormat().setItalic(true);
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
IWTextRange text = paragraph.appendText("A new paragraph style is created
and is applied to this paragraph.");
//Apply the new style to the paragraph.
paragraph.applyStyle("User_defined_style");
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();

```

### Tab stop

A tab stop is a horizontal position that is set for aligning text of the paragraph. A tab character causes the carriage to go to the next tab stop.

Each paragraph has its own tab stop collection where the new tab stop can be added and existing tab stop can be removed.

The following code example explains how to add tab stops to the paragraph.

### JAVA

```

//Creates a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add a tab stop at position 11.
Tab firstTab = paragraph.getParagraphFormat().getTabs().addTab(11,
TabJustification.Left, TabLeader.Dotted);
//Add a tab stop at position 62.
paragraph.getParagraphFormat().getTabs().addTab(62, TabJustification.Left,
TabLeader.Single);
paragraph.appendText("This sample\t illustrates the use of tabs in the
paragraph. Tabs\t can be inserted or removed from the paragraph.");
//Remove the tab stop from the collection.
paragraph.getParagraphFormat().getTabs().removeByTabPosition(11);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();

```



### RTL paragraph

You can set RTL (Right-to-left) direction to the paragraph in a Word document. The following code example shows how to set RTL (Right-to-left) for a paragraph in Word document.

#### JAVA

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx");
//Get the text body of the first section.
WTextBody textBody = document.getSections().get(0).getBody();
//Get the paragraph at index 1.
WParagraph paragraph = textBody.getParagraphs().get(1);
//Get a value indicating whether the paragraph is right-to-left. True
indicates the paragraph direction is RTL.
boolean isRTL = paragraph.getParagraphFormat().getBidi();
//Set the RTL direction for a paragraph.
if(!isRTL)
paragraph.getParagraphFormat().setBidi(true);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Working with text

Text within a paragraph is represented by one or more instances of the **WTextRange**. Each **WTextRange** instance can have its own font (text) formatting.

The following code example explains how to append text to the paragraph.

#### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph firstParagraph = section.addParagraph();
//Add a new text to the paragraph.
IWTextRange firstText = firstParagraph.appendText("A new text is added to
the paragraph.");
firstText.getCharacterFormat().setFontSize(14);
firstText.getCharacterFormat().setBold(true);
firstText.getCharacterFormat().setTextColor(ColorSupport.getGreen());
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

Text in the paragraph can be modified or replaced with a new text. This can be achieved by iterating through the paragraph items.

The following code example explains how to replace the text of a text range.

#### JAVA

```
//Load the template document.
```



```

WordDocument document = new WordDocument("Template.docx");
//Get the last paragraph.
WParagraph lastParagraph = document.getLastParagraph();
//Iterate through the paragraph items to get the text range and modifies its
content.
for(int i = 0;i<lastParagraph.getChildEntities().getCount();i++)
{
if(lastParagraph.getChildEntities().get(i) instanceof WTextRange)
{
WTextRange textRange =
(WTextRange)(lastParagraph.getChildEntities().get(i));
textRange.setText("First text range of the last paragraph is replaced");
textRange.getCharacterFormat().setFontSize(14);
break;
}
}
//Save and close the Word document.
document.save("Sample.docx",FormatType.Docx);
document.close();

```

Text formatting enhances the appearance of text in the document. Text formatting includes font size, font color, font name, bold, italic, underline, etc.

The following code example explains how to apply formatting to the text.

#### JAVA

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph firstParagraph = section.addParagraph();
//Add a new text to the paragraph.
IWTextRange firstText = firstParagraph.appendText("This is the first text
range. ");
//Apply formatting for the first text range.
firstText.getCharacterFormat().setBold(true);
firstText.getCharacterFormat().setFontSize(14);
firstText.getCharacterFormat().setShadow(true);
firstText.getCharacterFormat().setSmallCaps(true);
IWTextRange secondText = firstParagraph.appendText("This the second text
range");
//Apply formatting for the second text range.
secondText.getCharacterFormat().setHighlightColor(ColorSupport.getGreenYellow());
secondText.getCharacterFormat().setUnderlineStyle(UnderlineStyle.DotDash);
secondText.getCharacterFormat().setItalic(true);
secondText.getCharacterFormat().setFontName("Times New Roman");
secondText.getCharacterFormat().setTextColor(ColorSupport.getGreen());
//Add a new paragraph to the section.
IWParagraph secondParagraph = section.addParagraph();
//Adds new text to the paragraph
IWTextRange thirdText = secondParagraph.appendText("ל?ו?ל?ו?ל?ו?");
thirdText.getCharacterFormat().setBidi(true);
//Sets language identifier for right to left characters.

```



```
thirdText.getCharacterFormat().setLocaleIdBidi((short)LocaleIDs.he_IL.getEnumValue());
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

## Working with Images

DocIO provides support for both inline and absolute positioned images.

- Inline images: The position of the image is constrained to the lines of text on the page.
- Absolute positioned images: The images can be positioned anywhere irrespective of the lines of text.

The following code example explains how to add an image to the paragraph.

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph firstParagraph = section.addParagraph();
//Add an image to the paragraph.
IWPicture picture = firstParagraph.appendPicture(new
FileInputStream("Image.png"));
//Set the height and width for the image.
picture.setHeight(100);
picture.setWidth(100);
//Saves the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

## Replace image

The image present in the document can be replaced with a new image. This can be achieved by iterating through the paragraph items.

The following code example explains how to replace an existing image.

### JAVA

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx");
WTextBody textbody = document.getSections().get(0).getBody();
//Iterate through the paragraphs of the textbody.
for(Object paragraph_tempObj : textbody.getParagraphs())
{
    WParagraph paragraph = (WParagraph)paragraph_tempObj;
    //Iterate through the child elements of the paragraph.
    for(Object item_tempObj : paragraph.getChildEntities())
    {
        ParagraphItem item = (ParagraphItem)item_tempObj;
        if(item instanceof WPicture)
```



```

{
WPicture picture = (WPicture)(item);
//Replace the image.
if (picture.getTitle() == "Bookmark")
{
FileInputStream file = new FileInputStream("Image.png");
byte[] byarr = new byte[(int) file.getChannel().size()];
file.read(byarr);
picture.loadImage(byarr);
}
}
}
}
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();

```

### Remove image

Images can be removed from the document by removing it from the paragraph items.

The following code example explains how to remove the image from the paragraph items.

### JAVA

```

//Load the template document.
WordDocument document = new WordDocument("Template.docx");
WTextBody textbody = document.getSections().get(0).getBody();
//Iterate through the paragraphs of the textbody.
for (Object paragraph_tempObj : textbody.getParagraphs())
{
WParagraph paragraph = (WParagraph)paragraph_tempObj;
//Iterate through the child elements of paragraph.
for (int i = 0; i < paragraph.getChildEntities().getCount(); i++)
{
if (paragraph.getChildEntities().get(i) instanceof WPicture)
{
//Remove the images from the paragraph.
paragraph.getItems().removeAt(i);
i--;
}
}
}
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();

```

### Format and rotate images

Absolute positioned images have properties such as position, wrap formats, and alignments. These properties are not applicable when the text wrapping style is inline. You can also rotate an image and apply flipping (horizontal and vertical) to it.

The following code example explains how various picture formats can be applied to the picture.

### JAVA



```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("This paragraph has picture. ");
//Append the new picture to the paragraph.
WPicture picture = (WPicture) paragraph.appendPicture(new
FileInputStream("Image.png"));
//Set the text wrapping style - When the wrapping style is inline, the
images are not absolutely positioned. It is added next to the text range.
picture.setTextWrappingStyle(TextWrappingStyle.Square);
//Set the horizontal and vertical origin.
picture.setHorizontalOrigin(HorizontalOrigin.Page);
picture.setVerticalOrigin(VerticalOrigin.Paragraph);
//Set the width and height for the paragraph.
picture.setWidth(150);
picture.setHeight(100);
//Set the horizontal and vertical position for the picture.
picture.setHorizontalPosition(200);
picture.setVerticalPosition(150);
picture.setName("PictureName");
//Set the horizontal and vertical alignments.
picture.setHorizontalAlignment(ShapeHorizontalAlignment.Center);
picture.setVerticalAlignment(ShapeVerticalAlignment.Bottom);
//Set the 90 degree rotation.
picture.setRotation(90);
//Set the horizontal flip.
picture.setFlipHorizontal(true);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();

```

#### Find an image by title

An Image with a specific title can be retrieved by iterating the paragraph items that can be used for further manipulations.

The following code example explains how images can be iterated from the document elements.

#### JAVA

```

//Create a new Word document.
WordDocument document = new WordDocument("Template.docx");
//Get the textbody content.
WTextBody textBody = document.getSections().get(0).getBody();
//Iterate through the textbody child entities.
for(Object item_tempObj : textBody.getChildEntities())
{
    TextBodyItem item = (TextBodyItem)item_tempObj;
    if(item instanceof WParagraph)
    {
        WParagraph paragraph = (WParagraph)(item);
        for(Object paraItem_tempObj : paragraph.getChildEntities())
        {

```



```

ParagraphItem paraItem = (ParagraphItem)paraItem_tempObj;
if(paraItem instanceof WPicture)
{
    //Get the image from its title and modifies its width and height.
    WPicture picture = (WPicture) (paraItem);
    if(picture.getTitle () == "Bookmark")
    {
        picture.setWidth(150);
        picture.setHeight(100);
    }
}
}
}
}
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();

```

### Add Image caption

You can add caption to an image and update the caption numbers (Sequence fields) using `addCaption` method.

The following code example shows how to add caption to an image.

### JAVA

```

//Create a new document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Set margin of the section.
section.getPageSetup().getMargins().setAll(72);
//Add a paragraph to the section.
IWParagraph paragraph = section.addParagraph();
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Center);
//Add an image to the paragraph.
IWPicture picture = paragraph.appendPicture(new
FileInputStream("Google.png"));
//Add an image caption.
IWParagraph lastParagraph = picture.addCaption("Figure",
CaptionNumberingFormat.Roman, CaptionPosition.AfterImage);
//Align the caption.
lastParagraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignmen
t.Center);
//Set after spacing.
lastParagraph.getParagraphFormat().setAfterSpacing(12f);
//Set before spacing.
lastParagraph.getParagraphFormat().setBeforeSpacing(1.5f);
//Add a paragraph to the section.
paragraph = section.addParagraph();
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Ce
nter);
//Add an image to the paragraph.
picture = paragraph.appendPicture(new FileInputStream("Yahoo.png"));

```



```
//Add an image caption.  
lastParagraph = picture.addCaption("Figure", CaptionNumberingFormat.Roman,  
CaptionPosition.AfterImage);  
//Align the caption.  
lastParagraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignmen  
t.Center);  
//Set before spacing.  
lastParagraph.getParagraphFormat().setBeforeSpacing(1.5f);  
//Update the fields in a Word document.  
document.updateDocumentFields();  
//Save and close the document.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

By executing the above code example, it generates output Word document as follows.



### Working with lists

Lists can organize and format the contents of a document in a hierarchical way. There are nine levels in the list, starting from level 0 to level 8. DocIO supports both built-in list styles and custom list styles. The following are the types of list supported in DocIO:

- Numbered list
- Bulleted list

The following code example explains how to create a simple bulleted list.

### JAVA

```
//Create a new Word document.  
WordDocument document = new WordDocument();  
//Add a new section to the document.  
IWSection section = document.addSection();
```



```
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Apply the default numbered list style.
paragraph.getListFormat().applyDefBulletStyle();
//Add text to the paragraph.
paragraph.appendText("List item 1");
//Continue the list defined.
paragraph.getListFormat().continueListNumbering();
//Add the second paragraph.
paragraph = section.addParagraph();
paragraph.appendText("List item 2");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Add a new paragraph.
paragraph = section.addParagraph();
paragraph.appendText("List item 3");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Save the Word document.
document.save("simple bulleted list.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example explains how to create a simple numbered list.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Apply the default numbered list style.
paragraph.getListFormat().applyDefNumberedStyle();
//Add the text to the paragraph.
paragraph.appendText("List item 1");
//Continue the list defined.
paragraph.getListFormat().continueListNumbering();
//Add the second paragraph.
paragraph = section.addParagraph();
paragraph.appendText("List item 2");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Add a new paragraph.
paragraph = section.addParagraph();
paragraph.appendText("List item 3");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example explains how to create a multilevel bulleted list.



**JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Apply the default numbered list style.
paragraph.getListFormat().applyDefBulletStyle();
//Add the text to the paragraph.
paragraph.appendText("List item 1 - Level 0");
//Continue the list defined.
paragraph.getListFormat().continueListNumbering();
//Add the second paragraph.
paragraph = section.addParagraph();
paragraph.appendText("List item 2 - Level 1");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Add a new paragraph.
paragraph = section.addParagraph();
paragraph.appendText("List item 3 - Level 2");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example explains how to create multilevel numbered list.

**JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Apply the default numbered list style.
paragraph.getListFormat().applyDefNumberedStyle();
//Add the text to the paragraph.
paragraph.appendText("List item 1 - Level 0");
//Continue the list defined.
paragraph.getListFormat().continueListNumbering();
//Add the second paragraph.
paragraph = section.addParagraph();
paragraph.appendText("List item 2 - Level 1");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Add a new paragraph.
```



```
paragraph = section.addParagraph();
paragraph.appendText("List item 3 - Level 2");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The list levels can be incremented or decremented by using the `increaseIndentLevel` and `decreaseIndentLevel` methods respectively. The following code example explains how to increase or decrease the list indent levels.

### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Apply the default numbered list style.
paragraph.getListFormat().applyDefNumberedStyle();
//Add the text to the paragraph.
paragraph.appendText("Multilevel numbered list - Level 0");
//Continue the list defined.
paragraph.getListFormat().continueListNumbering();
//Add the second paragraph
paragraph = section.addParagraph();
paragraph.appendText("Multilevel numbered list - Level 1");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Add a new paragraph.
paragraph = section.addParagraph();
paragraph.appendText("Multilevel numbered list - Level 0");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().decreaseIndentLevel();
//Add a new paragraph.
paragraph = section.addParagraph();
paragraph.appendText("Multilevel numbered list - Level 1");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```



The following code example explains how to create user defined list styles.

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new list style to the document.
ListStyle listStyle = document.addListStyle(ListType.Numbered,
"UserDefinedList");
WListLevel levelOne = listStyle.getLevels().get(0);
//Define the follow character, prefix, suffix, and start index for level 0.
levelOne.setFollowCharacter(FollowCharacterType.Tab);
levelOne.setNumberPrefix("(");
levelOne.setNumberSuffix(")");
levelOne.setPatternType(ListPatternType.LowRoman);
levelOne.setStartAt(1);
levelOne.setTabSpaceAfter(5);
levelOne.setNumberAlignment(ListNumberAlignment.Center);
WListLevel levelTwo = listStyle.getLevels().get(1);
//Define the follow character, suffix, pattern, and start index for level 1.
levelTwo.setFollowCharacter(FollowCharacterType.Tab);
levelTwo.setNumberPrefix("}");
levelTwo.setPatternType(ListPatternType.LowLetter);
levelTwo.setStartAt(2);
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add the text to the paragraph.
paragraph.appendText("User defined list - Level 0");
//Apply the default numbered list style.
paragraph.getListFormat().applyStyle("UserDefinedList");
//Add the second paragraph.
paragraph = section.addParagraph();
paragraph.appendText("User defined list - Level 1");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example explains how to create numbered list with prefix from previous level.

**Note:** The `NumberPrefix` value for the numbered list should meet the syntax “\u000N” to update the previous list level value as a prefix to the current list level. For example, it should be represented as (“\u0000.” or “\u0000.\u0001.”).

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new list style to the document.
```



```

ListStyle listStyle = document.addListStyle(ListType.Numbered,
"UserDefinedList");
WListLevel levelOne = listStyle.getLevels().get(0);
//Define the follow character, prefix from previous level, and start index
for level 0.
levelOne.setFollowCharacter(FollowCharacterType.Nothing);
levelOne.setPatternType(ListPatternType.Arabic);
levelOne.setStartAt(1);
WListLevel levelTwo = listStyle.getLevels().get(1);
//Define the follow character, prefix from previous level, pattern, and
start index for level 1.
levelTwo.setFollowCharacter(FollowCharacterType.Nothing);
levelTwo.setNumberPrefix("\u0000.");
levelTwo.setPatternType(ListPatternType.Arabic);
levelTwo.setStartAt(1);
WListLevel levelThree = listStyle.getLevels().get(2);
//Define the follow character, prefix from previous level, pattern, and
start index for level 1.
levelThree.setFollowCharacter(FollowCharacterType.Nothing);
levelThree.setNumberPrefix("\u0000.\u0001.");
levelThree.setPatternType(ListPatternType.Arabic);
levelThree.setStartAt(1);
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add a text to the paragraph.
paragraph.appendText("User defined list - Level 0");
//Apply the default numbered list style.
paragraph.getListFormat().applyStyle("UserDefinedList");
//Add the second paragraph.
paragraph = section.addParagraph();
paragraph.appendText("User defined list - Level 1");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Add the second paragraph.
paragraph = section.addParagraph();
paragraph.appendText("User defined list - Level 2");
//Continue the last defined list.
paragraph.getListFormat().continueListNumbering();
//Increase the level indent.
paragraph.getListFormat().increaseIndentLevel();
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();

```

### Get list value

You can get the string that represents the appearance of **list value of the paragraph** in the Word document using the `ListString` API.

This API holds the static string of the list value recently calculated while saving the document as Text. It is not updated automatically for each modification done in the Word document. Hence, you should either invoke the `getText()` method of `WordDocument` or save the Word document as Text to get the actual list value from this API.



The following example shows how to **get a string that represents the appearance of list value of the paragraph**.

#### JAVA

```
//Load an existing Word document.
WordDocument document = new WordDocument("Template.docx");
//Get the document text.
document.getText();
//Get the string that represents the appearance of list value of the
paragraph.
String listString = document.getLastParagraph().getListString();
//Save and close the WordDocument instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

**Note:** For a picture bulleted list, the `ListString` API is not valid and it will return an empty string.

#### Working with hyperlinks

Hyperlink is a reference to data that can link to external contents like images, files, webpage, and more. In Word document, a hyperlink may target to any one of the following sources:

- Webpage: Represents the web content.
- File: Represents the file in some location.
- Email: Represents an Email.
- Bookmark: Represents the bookmarks in the document.

Hyperlinks have two parts: the address and display content.

The following code example explains how to insert a web link.

#### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("Web Hyperlink: ");
paragraph = section.addParagraph();
//Append the web hyperlink to the paragraph.
IWField field = paragraph.appendHyperlink("http://www.syncfusion.com",
"Syncfusion", HyperlinkType.WebLink);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example illustrates how to add an email link.

#### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
```



```
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("Email hyperlink: ");
paragraph = section.addParagraph();
//Append an Email hyperlink to the paragraph.
paragraph.appendHyperlink("mailto:sales@syncfusion.com", "Sales" ,
HyperlinkType.EmailLink);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example explains how to add a file hyperlink.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("File Hyperlinks: ");
paragraph = section.addParagraph();
//Append the hyperlink field to the paragraph.
paragraph.appendHyperlink("Template.docx", "File", HyperlinkType.FileLink);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example explains how to add a bookmark hyperlink.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Create a new Bookmark.
paragraph.appendBookmarkStart("Introduction");
paragraph.appendText("Hyperlink");
paragraph.appendBookmarkEnd("Introduction");
paragraph.appendText("\nA hyperlink is a reference or navigation element in
a document to another section of the same document or to another document
that may be on or part of a (different) domain.");
paragraph = section.addParagraph();
paragraph.appendText("Bookmark Hyperlink: ");
paragraph = section.addParagraph();
//Append a Bookmark hyperlink to the paragraph.
paragraph.appendHyperlink("Introduction", "Bookmark",
HyperlinkType.Bookmark);
```



```
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The display content for the Hyperlinks can also be an image that may redirect to some other contents.

The following code example explains how to add image hyperlink.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("Image Hyperlink");
paragraph = section.addParagraph();
//Create a new image instance and load image.
WPicture picture = new WPicture(document);
picture.loadImage(new FileInputStream("Image.png"));
//Append a new image hyperlink to the paragraph.
paragraph.appendHyperlink("http://www.syncfusion.com", picture,
HyperlinkType.WebLink);
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Close the document.
document.close();
```

The following code example explains how to modify the URL of an existing hyperlink.

#### **JAVA**

```
//Load the template document.
WordDocument document = new WordDocument("Sample.docx", FormatType.Docx);
WParagraph paragraph = document.getLastParagraph();
//Iterate through the paragraph items.
for(Object item_tempObj : paragraph.getChildEntities())
{
ParagraphItem item = (ParagraphItem)item_tempObj;
if(item instanceof WField)
{
if(((WField)(item)).getFieldType() == FieldType.FieldHyperlink)
{
//Get the hyperlink field.
Hyperlink link = new Hyperlink((WField)(item));
if(link.getType() == HyperlinkType.WebLink)
{
//Modifies the URL of the hyperlink.
link.setUri("http://www.google.com");
link.setTextToDisplay("Google");
break;
}
}
}
}
```



```

}
//Save and close the Word document.
document.save("Sample.docx",FormatType.Docx);
document.close();

```

### Working with symbols

Symbols are used to add contents such as currencies, numbers, punctuations, and more. DocIO represents symbols with the `WSymbol` instance. Each symbol can be identified with their character codes.

The following code example explains how to add new symbol to the document.

#### **JAVA**

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("Example of adding symbols to the paragraph: ");
//Insert the symbol with character code 100.
paragraph.appendSymbol((byte) 100);
//Save and close the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

The following code example explains how to modify an existing symbol.

#### **JAVA**

```

//Load the template document.
WordDocument document = new WordDocument("Template.docx",FormatType.Docx);
//Get the textbody content.
WTextBody textbody = document.getSections().get(0).getBody();
//Iterate through the paragraphs.
for(Object paragraph_tempObj : textbody.getParagraphs())
{
    WParagraph paragraph = (WParagraph)paragraph_tempObj;
    for(Object item_tempObj : paragraph.getChildEntities())
    {
        ParagraphItem item = (ParagraphItem)item_tempObj;
        if(item instanceof WSymbol)
        {
            //Get the symbol from the paragraph items.
            WSymbol symbol = (WSymbol)(item);
            if(symbol.getCharacterCode() == (byte)100)
            {
                //Modify the character code.
                symbol.setCharacterCode((byte) 40);
                symbol.setFontName("Wingdings");
            }
        }
    }
}
//Save and close the Word document.

```



```
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

### Appending breaks

Breaks allow the document contents to split into multiple parts to customize the appearance of the contents. The following are the types of breaks supported in the DocIO:

- Page break: Starts the content in the next page.
- Line break: Starts the content in a new line.
- Column break: Starts the content in the next column.

The following code example explains how various types of breaks can be appended to the paragraphs.

#### JAVA

```
//Create a new Word document.  
WordDocument document = new WordDocument();  
//Add a new section to the document.  
IWSection section = document.addSection();  
//Add a new paragraph to the section.  
IWParagraph paragraph = section.addParagraph();  
paragraph.appendText("Before line break");  
//Add a line break to the paragraph.  
paragraph.appendBreak(BreakType.LineBreak);  
paragraph.appendText("After line break");  
IWParagraph pageBreakPara = section.addParagraph();  
pageBreakPara.appendText("Before page break");  
//Add a page break to the paragraph.  
pageBreakPara.appendBreak(BreakType.PageBreak);  
pageBreakPara.appendText("After page break");  
IWSection secondSection = document.addSection();  
//Add columns to the section.  
secondSection.addColumn(100, 2);  
secondSection.addColumn(100, 2);  
IWParagraph columnBreakPara = secondSection.addParagraph();  
columnBreakPara.appendText("Before column break");  
//Add a column break to the paragraph.  
columnBreakPara.appendBreak(BreakType.ColumnBreak);  
columnBreakPara.appendText("After column break");  
//Save and close the document instance.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

### Working with Text Box

Text box contains a group of textual and graphical contents. DocIO supports to create and manipulate the text box and its formatting by using the `WTextBox` instance.

The following code example explains how to add new text box to the paragraph.

#### JAVA

```
//Create a new Word document.  
WordDocument document = new WordDocument();  
//Add a new section to the document.
```



```

IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Append a new textbox to the paragraph.
IWTextBox textbox = paragraph.appendTextBox(150, 75);
//Add a new text to the textbox body.
IWParagraph textboxParagraph = textbox.getTextBoxBody().addParagraph();
textboxParagraph.appendText("Text inside text box");
textboxParagraph = textbox.getTextBoxBody().addParagraph();
//Add a new picture to textbox body.
IWPicture picture = textboxParagraph.appendPicture(new
FileInputStream("Image.png"));
picture.setHeight(75);
picture.setWidth(50);
//Save and close the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

### *Format and rotate text box*

Text box has its own formatting such as outline color, fill effects, text direction, wrap formats, and more. You can also rotate the text box and apply flipping (horizontal and vertical) to it.

The following code example explains how to apply formatting and rotation for the text box.

### **JAVA**

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add a new section to the document.
IWSection section = document.addSection();
//Add a new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Append a new textbox to the paragraph.
IWTextBox textbox = paragraph.appendTextBox(150, 75);
//Add a new text to the textbox body.
IWParagraph textboxParagraph = textbox.getTextBoxBody().addParagraph();
textboxParagraph.appendText("Text inside text box");
//Set the fill color and line width for the textbox.
textbox.getTextBoxFormat().setFillColor(ColorSupport.getLightGreen());
textbox.getTextBoxFormat().setLineWidth(2);
//Apply the textbox text direction.
textbox.getTextBoxFormat().setTextDirection(TextDirection.VerticalTopToBottom);
//Set the text wrapping style.
textbox.getTextBoxFormat().setTextWrappingStyle(TextWrappingStyle.InFrontOfText);
//Set the horizontal and vertical position.
textbox.getTextBoxFormat().setHorizontalPosition(200);
textbox.getTextBoxFormat().setVerticalPosition(200);
//Set the horizontal and vertical origin.
textbox.getTextBoxFormat().setVerticalOrigin(VERTICAL_ORIGIN_MARGIN);
textbox.getTextBoxFormat().setHorizontalOrigin(HORIZONTAL_ORIGIN_PAGE);
//Set the top and bottom margin values.
textbox.getTextBoxFormat().getInternalMargin().setBottom(5f);
textbox.getTextBoxFormat().getInternalMargin().setTop(5f);
//Set the 90 degree rotation.

```



```

textbox.getTextBoxFormat().setRotation(90);
//Set the horizontal flip.
textbox.getTextBoxFormat().setFlipHorizontal(true);
//Save and close the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

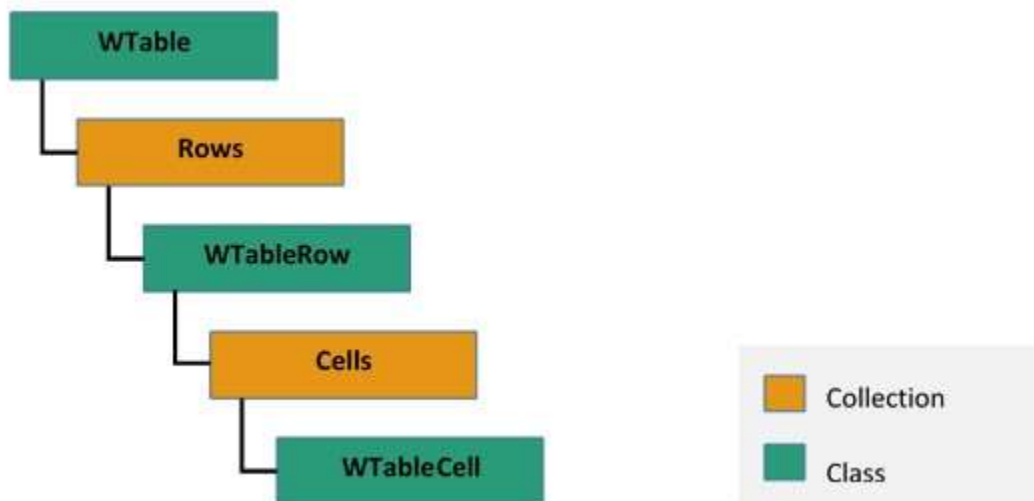
## Working with Tables in Word document

A table in Word document is used to arrange document content in rows and columns. **WTable** instance represents a table in Word document. A table must contain at least one row.

1. A row is a collection of cells and it is represented by an instance of **WTableRow**. Each row must contain at least one cell.
2. A cell can contain one or more paragraphs and tables. An instance of **WTableCell** represents a table cell. Each table cell must contain at least one paragraph.

**Note:** Adding more than 63 columns not supported in Word document using Microsoft Word application. It shows alert when you attempt to insert table with more than 64 columns, which is a one of the behaviors of Microsoft Word and Essential DocIO does the same.

The following image illustrates how a table in Word document is organized in EssentialDocIO's DOM:



The following code example illustrates how to create a simple table with predefined number of rows and cells.

### JAVA

```

//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
//Add a section into Word document.
IWSection section = document.addSection();
//Add a new paragraph into Word document and appends text into paragraph.
IWTextRange textRange = section.addParagraph().appendText("Price Details");
textRange.getCharacterFormat().setFontName("Arial");
textRange.getCharacterFormat().setFontSize(12);
textRange.getCharacterFormat().setBold(true);

```



```

section.addParagraph();
//Add a new table into Word document.
IWTable table = section.addTable();
//Specify the total number of rows & columns.
table.resetCells(3, 2);
//Access the instance of the cell (first row, first cell) and add the
content into cell.
textRange = table.get(0, 0).addParagraph().appendText("Item");
textRange.getCharacterFormat().setFontName("Arial");
textRange.getCharacterFormat().setFontSize(12);
textRange.getCharacterFormat().setBold(true);
//Access the instance of the cell (first row, second cell) and add the
content into cell.
textRange = table.get(0, 1).addParagraph().appendText("Price($)");
textRange.getCharacterFormat().setFontName("Arial");
textRange.getCharacterFormat().setFontSize(12);
textRange.getCharacterFormat().setBold(true);
//Access the instance of the cell (second row, first cell) and add the
content into cell.
textRange = table.get(1, 0).addParagraph().appendText("Apple");
textRange.getCharacterFormat().setFontName("Arial");
textRange.getCharacterFormat().setFontSize(10);
//Access the instance of the cell (second row, second cell) and add the
content into cell.
textRange = table.get(1, 1).addParagraph().appendText("50");
textRange.getCharacterFormat().setFontName("Arial");
textRange.getCharacterFormat().setFontSize(10);
//Access the instance of the cell (third row, first cell) and add the content
into cell.
textRange = table.get(2, 0).addParagraph().appendText("Orange");
textRange.getCharacterFormat().setFontName("Arial");
textRange.getCharacterFormat().setFontSize(10);
//Access the instance of the cell (third row, second cell) and add the
content into cell.
textRange = table.get(2, 1).addParagraph().appendText("30");
textRange.getCharacterFormat().setFontName("Arial");
textRange.getCharacterFormat().setFontSize(10);
//Save the document in the given name and format.
document.save("Table.docx", FormatType.Docx);
//Release the resources occupied by WordDocument instance.
document.close();

```

The following code example illustrates how to create a simple table by dynamically adding rows.

### JAVA

```

//Create an instance of WordDocument class
WordDocument document = new WordDocument();
IWSection section = document.addSection();
section.addParagraph().appendText("Price Details");
section.addParagraph();
//Add a new table into Word document
IWTable table = section.addTable();
//Add the first row into table
WTableRow row = table.addRow();
//Add the first cell into first row

```



```
WTableCell cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("Item");  
//Add the second cell into first row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("Price($)");  
//Add the second row into table  
row = table.addRow(true, false);  
//Add the first cell into second row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("Apple");  
//Add the second cell into second row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("50");  
//Add the third row into table  
row = table.addRow(true, false);  
//Add the first cell into third row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("Orange");  
//Add the second cell into third row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("30");  
//Add the fourth row into table  
row = table.addRow(true, false);  
//Add the first cell into fourth row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("Banana");  
//Add the second cell into fourth row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("20");  
//Add the fifth row to table  
row = table.addRow(true, false);  
//Add the first cell into fifth row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("Grapes");  
//Add the second cell into fifth row  
cell = row.addCell();  
//Specify the cell width  
cell.setWidth(200);  
cell.addParagraph().appendText("70");
```



```
//Save and close the document instance
document.save("Table.docx", FormatType.Docx);
document.close();
```

### Nested Table

You can create a nested table by adding a new table into a cell. The following code example illustrates how to add a table into a cell.

#### JAVA

```
//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
section.addParagraph().appendText("Price Details");
IWTable table = section.addTable();
table.resetCells(3, 2);
table.get(0, 0).addParagraph().appendText("Item");
table.get(0, 1).addParagraph().appendText("Price($)");
table.get(1, 0).addParagraph().appendText("Items with same price");
//Add a nested table into the cell (second row, first cell).
IWTable nestTable = table.get(1, 0).addTable();
//Create the specified number of rows and columns to nested table.
nestTable.resetCells(3, 1);
//Access the instance of the nested table cell (first row, first cell).
WTableCell nestedCell = nestTable.getRows().get(0).getCells().get(0);
//Specify the width of the nested cell.
nestedCell.setWidth(200);
//Add the content into nested cell.
nestedCell.addParagraph().appendText("Apple");
//Access the instance of the nested table cell (second row, first cell).
nestedCell = nestTable.getRows().get(1).getCells().get(0);
//Specify the width of the nested cell.
nestedCell.setWidth(200);
//Add the content into nested cell.
nestedCell.addParagraph().appendText("Orange");
//Access the instance of the nested table cell (third row, first cell).
nestedCell = nestTable.getRows().get(2).getCells().get(0);
//Specify the width of the nested cell.
nestedCell.setWidth(200);
//Add the content into nested cell.
nestedCell.addParagraph().appendText("Mango");
//Access the instance of the cell (second row, second cell).
nestedCell = table.getRows().get(1).getCells().get(1);
table.get(1, 1).addParagraph().appendText("85");
table.get(2, 0).addParagraph().appendText("Pomegranate");
table.get(2, 1).addParagraph().appendText("70");
//Save and close the document instance.
document.save("NestedTable.docx", FormatType.Docx);
document.close();
```

### Align text within a table

You can iterate the cells within a table and align text for each cell. Find more information about iterating the cells from [here](#)

The following code example illustrates how to align text within a table.



**JAVA**

```

private void alignCellContent(WTableCell tableCell,VerticalAlignment
verticalAlignment,HorizontalAlignment horizontalAlignment) throws Exception
{
    //Set vertical alignment to the cell.
    tableCell.getCellFormat().setVerticalAlignment(verticalAlignment);
    //Iterate body items in table cell and set horizontal alignment.
    alignCellContentForTextBody(tableCell,horizontalAlignment);
}
private void alignCellContentForTextBody(WTextBody
textBody,HorizontalAlignment horizontalAlignment) throws Exception
{
    for(int i = 0;i<textBody.getChildEntities().getCount();i++)
    {
        //IEntity is the basic unit in DocIO DOM.
        //Access the body items as IEntity.
        IEntity bodyItemEntity = textBody.getChildEntities().get(i);
        //A Text body has 3 types of elements - Paragraph, Table and Block Content
        Control.
        //Decide the element type by using EntityType.
        switch(bodyItemEntity.getEntityType().toString())
        {
            case "Paragraph":
                WParagraph paragraph = (WParagraph)bodyItemEntity;
                //Set horizontal alignment for paragraph.
                paragraph.getParagraphFormat().setHorizontalAlignment(horizontalAlignment);
                break;
            case "Table":
                //Table is a collection of rows and cells.
                //Iterate through table's DOM and set horizontal alignment.
                alignCellContentForTable((WTable)bodyItemEntity,horizontalAlignment);
                break;
            case "BlockContentControl":
                //Iterate to the body items of Block Content Control and set horizontal
                alignment.
                BlockContentControl blockContentControl =
                (BlockContentControl)bodyItemEntity;
                alignCellContentForTextBody(blockContentControl.getTextBody(),horizontalAlign
                ment);
                break;
            }
        }
    }
private void alignCellContentForTable(WTable table,HorizontalAlignment
horizontalAlignment) throws Exception
{
    //Iterate the row collection in a table.
    for(Object row_tempObj : table.getRows())
    {
        WTableRow row = (WTableRow)row_tempObj;
        //Iterate the cell collection in a table row.
        for(Object cell_tempObj : row.getCells())
        {
            //Iterate items in cell and set horizontal alignment.
            WTableCell cell = (WTableCell)cell_tempObj;
            alignCellContentForTextBody(cell,horizontalAlignment);
        }
    }
}

```



```

}
}
}

```

### Apply formatting to Table, Row and Cell

The following code example illustrates how to load an existing document and apply table formatting options such as Borders, LeftIndent, Paddings, IsAutoResize, etc.

**Note:** 1. `BorderStyle.None` is the default value of `BorderType` property in `Borders` class which will not show borders for the table or cell. It is equivalent to border style not defined and borders can be inherited from style or parent formats.

2. To hide border for a table or cell in the Word Document, you can set `BorderType` property with `BorderStyle.Cleared`. It means border style defined as no border (Don't show border) and shouldn't inherit from style or parent formats.

3. To show/display border for a table or cell in the Word Document, you can set `BorderType` property with `BorderStyle` values (except `BorderStyle.None` and `BorderStyle.Cleared`).

### JAVA

```

//Create an instance of WordDocument class (Empty Word Document)
WordDocument document = new WordDocument();
//Open an existing Word document into DocIO instance
document.open("Table.docx", FormatType.Docx);
//Access the instance of the first section in the Word document
WSection section = document.getSections().get(0);
//Access the instance of the first table in the section
WTable table = section.getTables().get(0);
//Specify the title for the table
table.setTitle("PriceDetails");
//Specify the description of the table
table.setDescription("This table shows the price details of various fruits");
//Specify the left indent of the table
table.setIndentFromLeft(50);
//Specify the background color of the table
table.getTableFormat().setBackColor(ColorSupport.fromArgb(192, 192, 192));
//Specify the horizontal alignment of the table
table.getTableFormat().setHorizontalAlignment(RowAlignment.Left);
//Specify the left, right, top and bottom padding of all the cells in the table
table.getTableFormat().getPaddings().setAll(10);
//Specify the auto resize of table to automatically resize all cell width based on its content
table.getTableFormat().setIsAutoResized(true);
//Specify the table top, bottom, left and right border line width
table.getTableFormat().getBorders().setLineWidth(2f);
//Specify the table horizontal border line width
table.getTableFormat().getBorders().getHorizontal().setLineWidth(2f);
//Specify the table vertical border line width
table.getTableFormat().getBorders().getVertical().setLineWidth(2f);
//Specify the tables top, bottom, left and right border color
table.getTableFormat().getBorders().setColor(ColorSupport.getRed());
//Specify the table Horizontal border color

```



```

table.getTableFormat().getBorders().getHorizontal().setColor(ColorSupport.getRed());
//Specify the table vertical border color
table.getTableFormat().getBorders().getVertical().setColor(ColorSupport.getRed());
//Specify the table borders border type
table.getTableFormat().getBorders().setBorderType(BorderStyle.Double);
//Accesses the instance of the first row in the table
WTableRow row = table.getRows().get(0);
//Specify the row height
row.setHeight(20);
//Specify the row height type
row.setHeightType(TableRowHeightType.AtLeast);
//Save and close the document instance
document.save("TableFormatting.docx", FormatType.Docx);
document.close();

```

### Applying cell formatting

The following code example illustrates how to load an existing document and apply cell formatting options such as VerticalAlignment, TextDirection, Paddings, Borders, etc.

### JAVA

```

//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
document.open("Table.docx", FormatType.Docx);
WSection section = document.getSections().get(0);
//Access the instance of the first row in the table.
WTable table = section.getTables().get(0);
//Specify the row height.
WTableRow row = table.getRows().get(0);
row.setHeight(20);
//Specify the row height type.
row.setHeightType(TableRowHeightType.AtLeast);
//Access the instance of the first cell in the row.
WTableCell cell = row.getCells().get(0);
//Specify the cell back ground color.
cell.getCellFormat().setBackColor(ColorSupport.fromArgb(192,192,192));
//Specify the same padding as table option as false to preserve current cell padding.
cell.getCellFormat().setSamePaddingsAsTable(false);
//Specify the left, right, top and bottom padding of the cell.
cell.getCellFormat().getPaddings().setLeft(5);
cell.getCellFormat().getPaddings().setRight(5);
cell.getCellFormat().getPaddings().setTop(5);
cell.getCellFormat().getPaddings().setBottom(5);
//Specify the vertical alignment of content of text.
cell.getCellFormat().setVerticalAlignment(VerticalAlignment.Middle);
//Disable the text wrap option to avoid displaying longer text on multiple lines.
cell.getCellFormat().setTextWrap(false);
//Access the instance of the second cell in the row.
cell=row.getCells().get(1);
cell.getCellFormat().setBackColor(ColorSupport.fromArgb(192,192,192));
cell.getCellFormat().setSamePaddingsAsTable(false);
//Specify the left, right, top and bottom padding of the cell.

```



```

cell.getCellFormat().getPadding().setAll(5);
cell.getCellFormat().setVerticalAlignment(VerticalAlignment.Middle);
//Disable the text wrap option to avoid displaying longer text on multiple
lines.
cell.getCellFormat().setTextWrap(false);
//Save and close the document instance.
document.save("TableCellFormatting.docx", FormatType.Docx);
document.close();

```

### Working with Table Style

A table style defines a set of table, row, cell and paragraph level formatting that can be applied to a table. `WTableStyle` instance represents table style in a Word document.

**Note:** Essential DocIO currently provides support for table styles in DOCX and WordML formats alone. The visual appearance is also preserved in Word to HTML conversion.

The following code example illustrates how to apply the built-in table styles to the table.

#### JAVA

```

//Create an instance of WordDocument class.
WordDocument document = new WordDocument("Table.docx", FormatType.Docx);
WSection section = document.getSections().get(0);
WTable table = section.getTables().get(0);
//Apply "LightShading" built-in style to table.
table.applyStyle(BuiltinTableStyle.LightShading);
//Save and close the document instance.
document.save("TableStyle.docx", FormatType.Docx);
document.close();

```

### Table style options

Once you have applied a table style, you can enable or disable the special formatting of the table. There are six options: first column, last column, banded rows, banded columns, header row and last row.

The following code example illustrates how to enable and disable the special table formatting options of the table styles

#### JAVA

```

//Create an instance of WordDocument class.
WordDocument document = new WordDocument("Table.docx", FormatType.Docx);
WSection section = document.getSections().get(0);
WTable table = section.getTables().get(0);
//Apply "LightShading" built-in style to table.
table.applyStyle(BuiltinTableStyle.LightShading);
//Enable special formatting for banded columns of the table.
table.setApplyStyleForBandedColumns(true);
//Enable special formatting for banded rows of the table.
table.setApplyStyleForBandedRows(true);
//Disables special formatting for first column of the table.
table.setApplyStyleForFirstColumn(false);
//Enable special formatting for header row of the table.
table.setApplyStyleForHeaderRow(true);
//Enable special formatting for last column of the table.
table.setApplyStyleForLastColumn(true);
//Disable special formatting for last row of the table.

```



```
table.setApplyStyleForLastRow(false);
//Save and close the document instance.
document.save("TableStyle.docx",FormatType.Docx);
document.close();
```

### Custom table style

The following code example illustrates how to apply a custom table style to table.

### JAVA

```
//Creates an instance of WordDocument class.
WordDocument document = new WordDocument("Table.docx", FormatType.Docx);
WSection section = document.getSections().get(0);
WTable table = section.getTables().get(0);
//Adds a new custom table style.
WTableStyle tableStyle = (WTableStyle)document.addTableStyle("CustomStyle");
//Applies formatting for whole table.
tableStyle.getTableProperties().setRowStripe(1);
tableStyle.getTableProperties().setColumnStripe(1);
tableStyle.getTableProperties().getPaddings().setTop(0);
tableStyle.getTableProperties().getPaddings().setBottom(0);
tableStyle.getTableProperties().getPaddings().setLeft(5.4f);
tableStyle.getTableProperties().getPaddings().setRight(5.4f);
//Applies conditional formatting for first row.
ConditionalFormattingStyle firstRowStyle =
tableStyle.getConditionalFormattingStyles().add(ConditionalFormattingType.Fi
rstRow);
firstRowStyle.getCharacterFormat().setBold(true);
firstRowStyle.getCharacterFormat().setTextColor(ColorSupport.fromArgb(255,25
5,255,255));
firstRowStyle.getCellProperties().setBackColor(ColorSupport.getBlue());
//Applies conditional formatting for first column.
ConditionalFormattingStyle firstColumnStyle =
tableStyle.getConditionalFormattingStyles().add(ConditionalFormattingType.Fi
rstColumn);
firstColumnStyle.getCharacterFormat().setBold(true);
//Applies conditional formatting for odd row.
ConditionalFormattingStyle oddRowBandingStyle =
tableStyle.getConditionalFormattingStyles().add(ConditionalFormattingType.Od
dRowBanding);
oddRowBandingStyle.getCellProperties().setBackColor(ColorSupport.getWhiteSmo
ke());
//Applies the custom table style to the table.
table.applyStyle("CustomStyle");
//Saves and closes the document instance
document.save("TableStyle.docx",FormatType.Docx);
document.close();
```

### Merging cells vertically and horizontally

You can combine two or more table cells located in the same row or column into a single cell.

The following code example illustrates how to apply horizontal merge to specified range of cells in a specified row.

### JAVA



```
//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
section.addParagraph().appendText("Horizontal merging of Table cells");
IWTable table = section.addTable();
table.resetCells(5,5);
//Specify the horizontal merge from second cell to fifth cell in third row.
table.applyHorizontalMerge(2,1,4);
//Save and close the document instance.
document.save("HorizontalMerge.docx",FormatType.Docx);
document.close();
```

The following code example illustrates how to apply vertical merge to specified range of rows in a specified column.

#### **JAVA**

```
//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
section.addParagraph().appendText("Vertical merging of Table cells");
IWTable table = section.addTable();
table.resetCells(5,5);
//Specify the vertical merge to the third cell, from second row to fifth row.
table.applyVerticalMerge(2,1,4);
//Save and close the document instance.
document.save("VerticalMerge.docx",FormatType.Docx);
document.close();
```

The following code example illustrate how to create a table that contains horizontal merged cells.

#### **JAVA**

```
//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
section.addParagraph().appendText("Horizontal merging of Table cells");
IWTable table = section.addTable();
table.resetCells(2,2);
//Add content to table cell.
table.get(0,0).addParagraph().appendText("First row, First cell");
table.get(0,1).addParagraph().appendText("First row, Second cell");
table.get(1,0).addParagraph().appendText("Second row, First cell");
table.get(1,1).addParagraph().appendText("Second row, Second cell");
//Specifies the horizontal merge start to first row, first cell.
table.get(0,0).getCellFormat().setHorizontalMerge(CellMerge.Start);
//Modify the cell content.
table.get(0,0).getParagraphs().get(0).setText("Horizontally merged cell");
//Specify the horizontal merge continue to second row second cell.
table.get(0,1).getCellFormat().setHorizontalMerge(CellMerge.Continue);
//Save and close the document instance.
document.save("HorizontalMerge.docx",FormatType.Docx);
document.close();
```



The following code example illustrates how to create a table with vertical merged cells.

#### JAVA

```
//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
section.addParagraph().appendText("Vertical merging of Table cells");
IWTable table = section.addTable();
table.resetCells(2,2);
//Add content to table cells.
table.get(0,0).addParagraph().appendText("First row, First cell");
table.get(0,1).addParagraph().appendText("First row, Second cell");
table.get(1,0).addParagraph().appendText("Second row, First cell");
table.get(1,1).addParagraph().appendText("Second row, Second cell");
//Specify the vertical merge start to first row first cell.
table.get(0,0).getCellFormat().setVerticalMerge(CellMerge.Start);
//Modify the cell content.
table.get(0,0).getParagraphs().get(0).setText("Vertically merged cell");
//Specify the vertical merge continue to second row first cell.
table.get(1,0).getCellFormat().setVerticalMerge(CellMerge.Continue);
//Save and close the document instance.
document.save("VerticalMerge.docx",FormatType.Docx);
document.close();
```

#### Specifying table header row to repeat on each page

You can specify one or more rows in a table to be repeated as header row at the top of each page, when the table spans across multiple pages.

- In the case of a single header row, it must be the first row in the table.
- In the case of multiple header rows, then header rows must be consecutive from the first row of the table.

**Note:** Heading rows do not have any effect with nested tables in Microsoft Word as well as DocIO

The following code example illustrates how to create a table with a single header row.

#### JAVA

```
//Create an instance of WordDocument class.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
IWTable table = section.addTable();
table.resetCells(50,1);
WTableRow row = table.getRows().get(0);
//Specify the first row as a header row of the table.
row.setIsHeader(true);
row.setHeight(20);
row.setHeightType(TableRowHeightType.AtLeast);
row.getCells().get(0).addParagraph().appendText("Header Row");
for(int i = 1;i<50;i++)
{
row=table.getRows().get(i);
row.setHeight(20);
row.setHeightType(TableRowHeightType.AtLeast);
```



```

row.getCells().get(0).addParagraph().appendText("Text in Row" + i);
}
//Save and close the document instance.
document.save("TableWithHeaderRow.docx",FormatType.Docx);
document.close();

```

### Keeping rows from breaking across pages

You can enable or disable the table row content to split across multiple pages, when the row contents do not fit in a previous page.

The following code example illustrates how to disable all the table rows from splitting across multiple pages.

#### JAVA

```

//Creates an instance of WordDocument class.
WordDocument document = new WordDocument("Template.docx");
WSection section = document.getSections().get(0);
WTable table = section.getTables().get(0);
//Disable breaking across pages for all rows in the table.
for(Object row_tempObj : table.getRows())
{
WTableRow row = (WTableRow)row_tempObj;
row.getRowFormat().setIsBreakAcrossPages(false);
}
//Save and close the document instance.
document.save("Result.docx",FormatType.Docx);
document.close();

```

### Iterating through table elements

The following code example illustrates how to iterate through the table and apply back color to a particular cell.

#### JAVA

```

//Create an instance of WordDocument class.
WordDocument document = new WordDocument("Template.docx");
WSection section = document.getSections().get(0);
WTable table = section.getTables().get(0);
//Iterate the rows of the table.
for(Object row_tempObj : table.getRows())
{
WTableRow row = (WTableRow)row_tempObj;
//Iterate through the cells of rows.
for(Object cell_tempObj : row.getCells())
{
WTableCell cell = (WTableCell)cell_tempObj;
//Iterate through the paragraphs of the cell.
for(Object paragraph_tempObj : cell.getParagraphs())
{
WParagraph paragraph = (WParagraph)paragraph_tempObj;
//When the paragraph contains text Panda then apply green as back color to cell.
if(paragraph.getText().contains("panda"))
cell.getCellFormat().setBackColor(ColorSupport.getGreen());
}
}
}

```



```
}  
}  
}  
//Save and close the document instance.  
document.save("Sample.docx",FormatType.Docx);  
document.close();  
document.Close();
```

### Removing the table

You can remove a table from a text body by its instance or by its index position in the text body item collection. The following code example shows how to remove a table in Word document.

#### JAVA

```
//Create an instance of WordDocument class.  
WordDocument document = new WordDocument("Template.docx");  
//Access the instance of the first section in the Word document.  
WSection section = document.getSections().get(0);  
//Access the instance of the first table in the section.  
WTable table = section.getTables().get(0);  
//Remove a table from the text body.  
section.getBody().getChildEntities().remove(table);  
//Save the Word document.  
document.save("Sample.docx",FormatType.Docx);  
//Close the document.  
document.close();
```

### Working with Bookmarks

A bookmark identifies a location or a selection of text within a document that you can name and identify for future reference.

In Essential DocIO, bookmark is represented by Bookmark instance that is a pair of BookmarkStart and BookmarkEnd. BookmarkStart represents start point of a bookmark and BookmarkEnd represents end point of a bookmark. Every Word document contains a collection of bookmarks that are accessible through the Bookmarks property of WordDocument class.

#### Adding a bookmark

The following code example shows how to add a bookmark in Word document.

#### JAVA

```
//Create an instance of WordDocument class (Empty Word Document).  
WordDocument document = new WordDocument();  
//Add a new section into the Word Document.  
IWSection section = document.addSection();  
//Add a new paragraph into Word document and appends text into paragraph.  
IWParagraph paragraph = section.addParagraph();  
paragraph.appendText("Northwind Database");  
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Center);  
//Add a paragraph into section.  
paragraph = section.addParagraph();  
//Add a new bookmark start into paragraph with name "Northwind".  
paragraph.appendBookmarkStart("Northwind");  
//Add a text between the bookmark start and end into paragraph.
```



```
paragraph.appendText("The Northwind sample database (Northwind.mdb) is included with all versions of Access. It provides data you can experiment with and database objects that demonstrate features you might want to implement in your own databases.");  
//Add a new bookmark end into paragraph with name " Northwind ".  
paragraph.appendBookmarkEnd("Northwind");  
//Add a text after the bookmark end.  
paragraph.appendText(" Using Northwind, you can become familiar with how a relational database is structured and how the database objects work together to help you enter, store, manipulate, and print your data.");  
//Save the document in the given name and format.  
document.save("Bookmarks.docx", FormatType.Docx);  
//Release the resources occupied by WordDocument instance.  
document.close();
```

### Obtaining a bookmark instance

The following code example shows how to retrieve an instance of bookmark from a Word document.

#### **JAVA**

```
//Load an existing Word document into DocIO instance.  
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);  
//Get the bookmark instance by using FindByName method of BookmarkCollection with bookmark name.  
Bookmark bookmark = document.getBookmarks().findByName("Northwind");  
//Access the bookmark start's owner paragraph by using bookmark and changes its back color.  
bookmark.getBookmarkStart().getOwnerParagraph().getParagraphFormat().setBackColor(ColorSupport.getAliceBlue());  
//Save and close the Word document.  
document.save("Result.docx", FormatType.Docx);  
document.close();
```

### Removing a Bookmark from Word document

The following code example shows how to remove a bookmark from Word document.

#### **JAVA**

```
//Load an existing Word document into DocIO instance.  
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);  
//Get the bookmark instance by using FindByName method of BookmarkCollection with bookmark name.  
Bookmark bookmark = document.getBookmarks().findByName("Northwind");  
//Remove the bookmark named "Northwind" from Word document.  
document.getBookmarks().remove(bookmark);  
//Save and close the Word document.  
document.save("Result.docx", FormatType.Docx);  
document.close();
```

### Retrieving contents within a bookmark

BookmarkNavigator is used for navigating to a bookmark in a Word document. You can retrieve, replace and delete the content of a specified bookmark by using BookmarkNavigator.



You can get the content between bookmark start and bookmark end of the specified bookmark in two ways:

1. You can use `getBookmarkContent` method for retrieving content as collection of body items when the bookmark start and bookmark end are preserved in a single section.
2. You can use `getContent` method for retrieving content as collection of sections when the bookmark start and bookmark end are preserved in different sections.

The following code example shows how to retrieve the specified bookmark content by using `getBookmarkContent` method in a Word document.

#### **JAVA**

```
//Open an input Word template.
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind");
//Get the bookmark content.
TextBodyPart part = bookmarkNavigator.getBookmarkContent();
//Add the retrieved content into another new section.
document.addSection();
for (int i = 0; i < part.getBodyItems().getCount(); i++)
document.getLastSection().getBody().getChildEntities().add(part.getBodyItems()
().get(i));
//Save and close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();
```

The following code example shows how to retrieve the specified bookmark content by using `getContent` method in a Word document.

#### **JAVA**

```
//Load the template document with bookmark "Northwind" whose start and end
are preserved in different section.
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind");
//Get the bookmark content as WordDocumentPart.
WordDocumentPart wordDocumentPart = bookmarkNavigator.getContent();
//Save the WordDocumentPart as separate Word document.
WordDocument newDocument = wordDocumentPart.getAsWordDocument();
//Close the WordDocumentPart instance.
wordDocumentPart.close();
//Close the template Word document.
document.close();
newDocument.save("Result.docx", FormatType.Docx);
//Release the resources hold by WordDocument instance.
newDocument.close();
```



### Retrieving bookmark contents within a table

You can select the column range for bookmarks inside the tables in Word documents by using `FirstColumn` and `LastColumn` properties.

- Note:**
1. `FirstColumn` and `LastColumn` properties are valid to select table cells, only when the respective bookmark end and start is present within the same row or next rows of the same table.
  2. `FirstColumn` property denotes the top left corner cell and `LastColumn` property denotes the bottom right corner cell of rectangular selection region since you can only select the content as a rectangular selection by using bookmarks within the table.
  3. `FirstColumn` property selects from the first cell of the respective row when this property value is negative (or) greater than the cells of a row (or) greater than the `LastColumn` value.
  4. `LastColumn` property selects till last cell of the respective row when this property value is negative (or) greater than the cells of a row (or) less than the `FirstColumn` value.

The following code example shows how to retrieve the bookmark content of a specified column range from a table in a Word document.

#### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a section and a paragraph in the document.
document.ensureMinimal();
//Insert a new table with bookmark.
IWTable table = CreateTable(document);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"BkmkInTable".
bookmarkNavigator.moveToBookmark("BkmkInTable");
//Set the column index where the bookmark starts within the table.
bookmarkNavigator.getCurrentBookmark().setFirstColumn((short) 1);
//Set the column index where the bookmark ends within the table.
bookmarkNavigator.getCurrentBookmark().setLastColumn((short) 4);
//Get the bookmark content.
TextBodyPart part = bookmarkNavigator.getBookmarkContent();
//Add new section.
document.addSection();
//Add the retrieved content into another new section.
for (int i = 0; i < part.getBodyItems().getCount(); i++)
document.getLastSection().getBody().getChildEntities().add(part.getBodyItems().get(i));
//Save and close the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

The following code example shows how to create table with bookmark.

#### JAVA

```
public IWTable CreateTable(WordDocument document) throws Exception
```



```

{
//Add a new table into Word document.
IWTable table = document.getLastSection().addTable();
//Specify the total number of rows & columns.
table.resetCells(5, 5);
//Access the instance of the cells and adds the content into cells.
table.get(0, 0).addParagraph().appendText("Supplier ID");
table.get(0, 1).addParagraph().appendText("Company Name");
IWParagraph paragraph =
table.getRows().get(0).getCells().get(2).addParagraph();
//Append a bookmark start in third cell of first row.
paragraph.appendBookmarkStart("BkmkInTable");
paragraph.appendText("Contact Name");
table.get(0, 3).addParagraph().appendText("Address");
table.get(0, 4).addParagraph().appendText("City");
table.get(1, 0).addParagraph().appendText("1");
table.get(1, 1).addParagraph().appendText("Exotic Liquids");
table.get(1, 2).addParagraph().appendText("Charlotte Cooper");
table.get(1, 3).addParagraph().appendText("49 Gilbert St.");
table.get(1, 4).addParagraph().appendText("London");
table.get(2, 0).addParagraph().appendText("2");
table.get(2, 1).addParagraph().appendText("New Orleans Cajun Delights");
table.get(2, 2).addParagraph().appendText("Shelley Burke");
table.get(2, 3).addParagraph().appendText("P.O. Box 78934");
table.get(2, 4).addParagraph().appendText("New Orleans");
table.get(3, 0).addParagraph().appendText("3");
table.get(3, 1).addParagraph().appendText("Grandma Kelly's Homestead");
table.get(3, 2).addParagraph().appendText("Regina Murphy");
table.get(3, 3).addParagraph().appendText("707 Oxford Rd.");
table.get(3, 4).addParagraph().appendText("Ann Arbor");
table.get(4, 0).addParagraph().appendText("4");
table.get(4, 1).addParagraph().appendText("Tokyo Traders");
paragraph = table.getRows().get(4).getCells().get(2).addParagraph();
//Append a bookmark end in third cell of last row.
paragraph.appendBookmarkEnd("BkmkInTable");
paragraph.appendText("Yoshi Nagase");
table.get(4, 3).addParagraph().appendText("9-8 Sekimai Musashino - shi");
table.get(4, 4).addParagraph().appendText("Tokyo");
return table;
}

```

### Inserting content into a bookmark

You can insert table, paragraph, simple text and paragraph item at the start or end location of the current bookmark by using bookmark navigator.

The following code example shows how to insert a simple text by using BookmarkNavigator.

### JAVA

```

//Load the template document with bookmark "Northwind".
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind");

```



```
//Insert a new text before the bookmark end of the bookmark.
bookmarkNavigator.insertText(" Northwind Database is a set of tables
containing data fitted into predefined categories.");
//Save and close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();
```

The following code example shows how to insert a paragraph item by using BookmarkNavigator.

#### **JAVA**

```
//Load the template document with bookmark "Northwind".
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind"
bookmarkNavigator.moveToBookmark("Northwind", false, true);
//Insert a new picture after the bookmark end.
WPicture picture = (WPicture)
bookmarkNavigator.insertParagraphItem(ParagraphItemType.Picture);
FileInputStream file = new FileInputStream("Northwind.png");
byte[] byarr = new byte[(int) file.getChannel().size()];
file.read(byarr);
picture.loadImage(byarr);
//Set the height and width for the image.
picture.setWidthScale(50);
picture.setHeightScale(50);
//Save and close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();
```

The following code example shows how to insert a paragraph by using BookmarkNavigator.

#### **JAVA**

```
//Load the template document with bookmark "Northwind".
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Load the template document with bookmark "Northwind".
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind", false, true);
//Insert a new paragraph before the bookmark start.
IWParagraph paragraph = new WParagraph(document);
paragraph.appendText("Northwind Database is a set of tables containing data
fitted into predefined categories.");
bookmarkNavigator.insertParagraph(paragraph);
//Save and close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();
```

The following code example shows how to insert a table by using BookmarkNavigator.



**JAVA**

```

WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind", false, false);
//Insert a new paragraph before the bookmark end.
IWParagraph paragraph = new WParagraph(document);
paragraph.appendText("Northwind Database Contains the following tables:");
bookmarkNavigator.insertParagraph(paragraph);
//Insert a new table before the bookmark end.
WTable table = new WTable(document);
table.resetCells(3, 2);
table.get(0, 0).addParagraph().appendText("Suppliers");
table.get(0, 1).addParagraph().appendText("2");
table.get(1, 0).addParagraph().appendText("Customers");
table.get(1, 1).addParagraph().appendText("1");
table.get(2, 0).addParagraph().appendText("Employees");
table.get(2, 1).addParagraph().appendText("3");
bookmarkNavigator.insertTable(table);
//Save and close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();

```

The following code example shows how to insert a TextBodyPart by using BookmarkNavigator.

**JAVA**

```

//Load the template document with bookmark "Northwind".
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind");
//Get the bookmark content.
TextBodyPart textBodyPart = bookmarkNavigator.getBookmarkContent();
document.addSection();
IWParagraph paragraph = document.getLastSection().addParagraph();
paragraph.appendText("Northwind Database is a set of tables containing data
fitted into predefined categories.");
//Add the new bookmark into Word document.
paragraph.appendBookmarkStart("bookmark_empty");
paragraph.appendBookmarkEnd("bookmark_empty");
//Move the virtual cursor to the location after the start of the bookmark
"bookmark_empty".
bookmarkNavigator.moveToBookmark("bookmark_empty", true, true);
//Insert the text body part after the bookmark start.
bookmarkNavigator.insertTextBodyPart(textBodyPart);
//Save and close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();

```



### Deleting content from a bookmark

You can delete the contents between bookmark start and end of the specified bookmark in a Word document.

The following code example shows how to remove the contents of a specified bookmark from Word document.

#### **JAVA**

```
//Load an existing Word document into DocIO instance.
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind");
//Delete bookmark content without deleting the format in the target
document.
bookmarkNavigator.deleteBookmarkContent(false);
//Save and close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();
```

### Replacing content in a bookmark

You can replace the contents of an existing bookmark with simple text, TextBodyPart, WordDocumentPart.

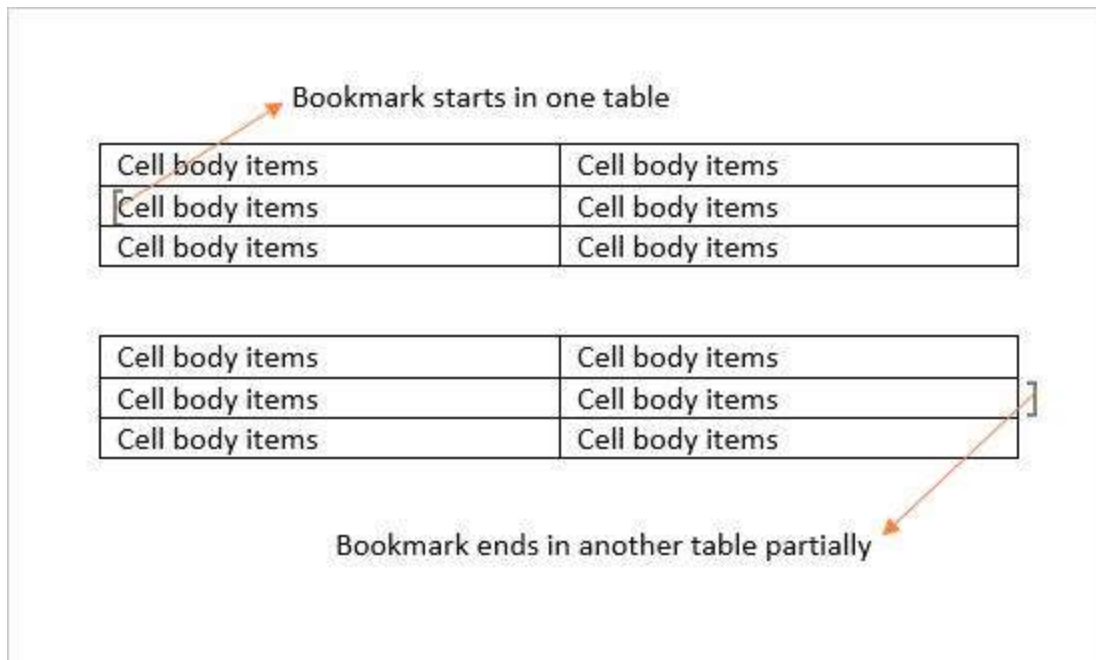
#### **Note:**

You cannot replace the multi section contents into a bookmark within table in Word documents. Use "for loop" instead of "foreach loop" to iterate through document elements when replacing the bookmark contents to avoid "collection modified exception", as there is a chance for modification in the document elements on replacing the bookmark contents.

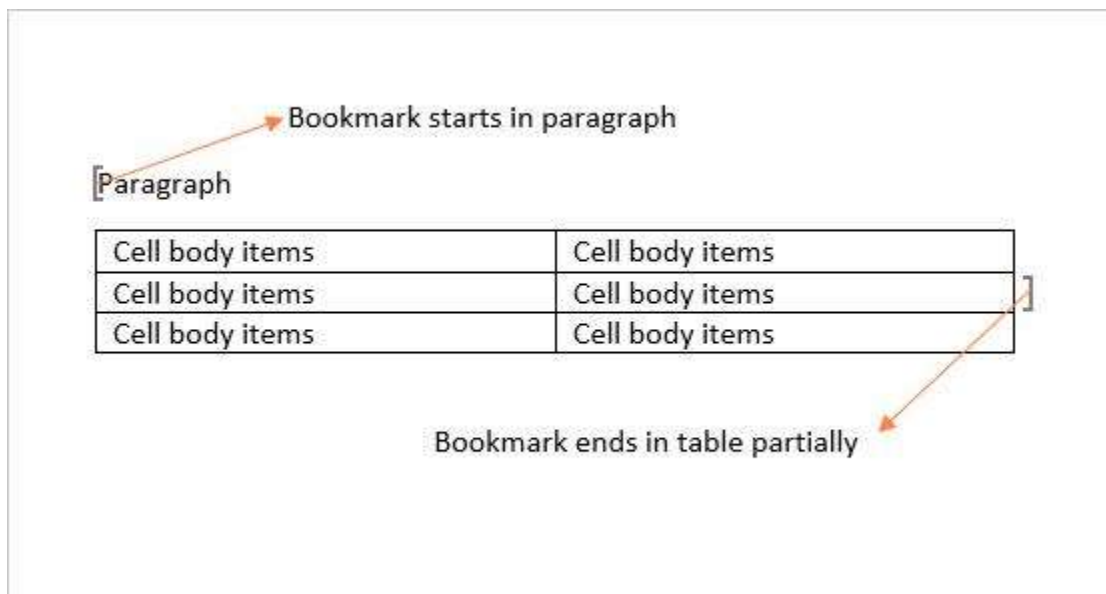
As per Microsoft Word behavior, you cannot replace the bookmark contents when the bookmark start and end is not in a same table as following cases:

Case 1





## Case 2



The following code example shows how to replace a specified bookmark content by using `replaceBookmarkContent` method in Word document.

**JAVA**

```
//Load an existing Word document into DocIO instance.
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Create the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new BookmarksNavigator(document);
//Move the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind");
//Get the bookmark content.
```



```

TextBodyPart textBodyPart = bookmarkNavigator.getBookmarkContent();
document.addSection();
IWParagraph paragraph = document.getLastSection().addParagraph();
paragraph.appendText("Northwind Database is a set of tables containing data
fitted into predefined categories.");
//Add the new bookmark into Word document.
paragraph.appendBookmarkStart("bookmark_empty");
paragraph.appendBookmarkEnd("bookmark_empty");
//Move the virtual cursor to the location before the end of the bookmark
"bookmark_empty".
bookmarkNavigator.moveToBookmark("bookmark_empty");
//Replace the bookmark content with text body part.
bookmarkNavigator.replaceBookmarkContent(textBodyPart);
//Save and close the Word document.
document.save("Result .docx", FormatType.Docx);
document.close();

```

The following code example shows how to replace a specified bookmark content by using `replaceContent` method in Word document.

#### JAVA

```

//Loads the template document with bookmark "Northwind" whose start and end
are preserved in different section.
WordDocument templateDocument = new WordDocument("Template.docx",
FormatType.Docx);
//Creates the bookmark navigator instance to access the bookmark.
BookmarksNavigator bookmarkNavigator = new
BookmarksNavigator(templateDocument);
//Moves the virtual cursor to the location before the end of the bookmark
"Northwind".
bookmarkNavigator.moveToBookmark("Northwind");
//Gets the bookmark content as WordDocumentPart.
WordDocumentPart wordDocumentPart = bookmarkNavigator.getContent();
//Loads the Word document with bookmark NorthwindDB.
WordDocument document = new WordDocument("Bookmarks.docx", FormatType.Docx);
//Creates the bookmark navigator instance to access the bookmark.
bookmarkNavigator = new BookmarksNavigator(document);
//Moves the virtual cursor to the location before the end of the bookmark
"NorthwindDB".
bookmarkNavigator.moveToBookmark("NorthwindDB");
//Replaces the bookmark content with word body part.
bookmarkNavigator.replaceContent(wordDocumentPart);
//Close the WordDocumentPart instance.
wordDocumentPart.close();
//Closes the template document.
templateDocument.close();
document.save("Result.docx", FormatType.Docx);
document.close();

```

### Working with document Fields

Fields in a Word document are placeholders for data that might change on field update. Fields are represented by the `WField` and `WFieldMark` instances in `DocIO`. A field in a Word document contains field codes, field separator, field result, and field end.



To learn various types of Microsoft Word supported fields and their syntax, refer to the [MSDN article](#)

The entire field code is included in Document Object Model(DOM). Hence, adding a field will automatically include the following elements in DOM:

1. **WField**: Represents the starting of a Field.
2. **ParagraphItem**: Represents the Field code.
3. **WFieldMark**: Represents the Field separator.
4. **ParagraphItem**: Represents the Field result.
5. **WFieldMark**: Represents the end of a Field.

### Adding fields

You can add a field to a Word document by using the **appendField** method of **WParagraph** class.

The following code example explains how to add a field to the Word document.

#### JAVA

```
//Creates an instance of WordDocument class (Empty Word Document).
WordDocument document = new WordDocument();
//Adds a new section to the Word Document.
IWSection section = document.addSection();
//Adds a new paragraph to Word document and appends text into paragraph.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("Today's Date: ");
//Adds the new Date field to Word document with field name and its type.
WField field = (WField)paragraph.appendField("Date", FieldType.FieldDate);
//Field code used to describe how to display the date.
field.setFieldCode(StringSupport.concat("DATE \@", "\"MMMM d, yyyy\""));
//Saves the document in the given name and format.
document.save("Sample.docx", FormatType.Docx);
//Releases the resources occupied by WordDocument instance.
document.close();
```

### Formatting fields

You can format the field instances added to the Word document by iterating the items from field start to end.

The following code example explains how to format the field in Word document.

#### JAVA

```
//Creates an instance of a WordDocument.
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document.
document.ensureMinimal();
//Adds the new Page field to Word document with field name and its type.
IWField field = document.getLastParagraph().appendField("Page",
FieldType.FieldPage);
IEntity entity = field;
//Iterates to sibling items until Field End.
while (entity.getNextSibling() != null)
{
    if (entity instanceof WTextRange)
        //Sets character format for text ranges.
}
```



```

((WTextRange) entity).getCharacterFormat().setFontSize((float) 6);
else if ((entity instanceof WFieldMark)
&& ((WFieldMark) entity).getType().getEnumValue() ==
FieldMarkType.FieldEnd.getEnumValue())
break;
//Gets next sibling item.
entity = entity.getNextSibling();
}
// Saves and closes the Word document instance.
document.save("Template.docx", FormatType.Docx);
document.close();

```

### Updating fields

Field updating engine calculates the resultant value based on the field code information and updates the field result with a new value. You can update the following fields by using DocIO:

- = (formula field)
- DATE
- TIME
- DOCVARIABLE
- DOCPROPERTY
- COMPARE
- IF
- NEXTIF
- MERGEREC
- MERGESEQ
- SECTION
- Numpages
- TITLE
- Cross-Reference
- SEQ

The following code example explains how to update the fields present in Word document.

#### **JAVA**

```

//Loads an existing Word document into DocIO instance.
WordDocument document = new WordDocument("Input.docx", FormatType.Docx);
//Updates the fields present in a document.
document.updateDocumentFields();
//Saves and closes the Word document instance.
document.save("Result.docx", FormatType.Docx);
document.close();

```

### IF field

IF field compares two values and updates the field result with true text, when comparison succeeds otherwise false text.

To learn more about IF field and its syntax in Microsoft Word, refer to the [MSDN article](#)

The following code example explains how to add an If field to a Word document.



**JAVA**

```
//Creates an instance of a WordDocument.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("If field which uses string of characters in
expression");
paragraph=section.addParagraph();
//Creates the new instance of IF field.
WIFfield field = (WIFfield)paragraph.appendField("If", FieldType.FieldIf);
//Specifies the expression, true and false statement in field code.
field.setFieldCode("IF \"True\" = \"True\" \"The given statement is
Correct\" \"The given statement is Wrong\"");
paragraph = section.addParagraph();
paragraph.appendText("If field which uses numbers in expression");
paragraph = section.addParagraph();
//Creates the new instance of IF field
field = (WIFfield)paragraph.appendField("If", FieldType.FieldIf);
//Specifies the expression, true and false statement in field code.
field.setFieldCode("IF 100 >= 1000 \"The given statement is Correct\" \"The
given statement is Wrong\"");
//Updates the document fields.
document.updateDocumentFields();
// Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

**Document variables**

The DocVariable field displays the value of a specified document variable in the Word document. The document variables can be added or modified using the Variables property of WordDocument class.

The following code example explains how to add a DocVariable field to a Word document.

**JAVA**

```
//Creates an instance of a WordDocument.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("First Name of the customer: ");
//Adds the DocVariable field with Variable name and its type.
paragraph.appendField("FirstName",FieldType.FieldDocVariable);
paragraph = section.addParagraph();
paragraph.appendText("Last Name of the customer: ");
//Adds the DocVariable field with Variable name and its type.
paragraph.appendField("LastName", FieldType.FieldDocVariable);
//Adds the value for variable in WordDocument.Variable collection.
document.getVariables().add("FirstName", "Jeff");
document.getVariables().add("LastName", "Smith");
//Updates the document fields.
document.updateDocumentFields();
// Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```



### Cross reference

A cross-reference refers to an item that appears in another location in a document. You can create cross-reference to bookmarks in a document by using the `appendCrossReference` method of `WParagraph` class.

**Note:** The Essential DocIO supports creating and updating the cross-reference fields only for bookmarks in a document.

The following code example explains how to append cross reference for bookmark in a Word document.

#### JAVA

```
//Creates an instance of a WordDocument.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
IWParagraph paragraph = section.addParagraph();
//Adds text, bookmark start and end in the paragraph.
paragraph.appendBookmarkStart("Title");
paragraph.appendText("Northwind Database");
paragraph.appendBookmarkEnd("Title");
paragraph = section.addParagraph();
paragraph.appendText("The Northwind sample database (Northwind.mdb) is
included with all versions of Access. It provides data you can experiment
with and database objects that demonstrate features you might want to
implement in your own databases.");
section = document.addSection();
section.addParagraph();
paragraph = (WParagraph)section.addParagraph();
//Gets the collection of bookmark start in the word document.
ListSupport<Entity> items =
document.getCrossReferenceItems(ReferenceType.Bookmark);
paragraph.appendText("Bookmark Cross Reference starts here ");
//Appends the cross reference for bookmark "Title" with ContentText as
reference kind.
paragraph.appendCrossReference(ReferenceType.Bookmark,
ReferenceKind.ContentText, items.get(0), true, false, false, "");
//Updates the document Fields.
document.updateDocumentFields();
// Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Unlink fields

You can replace the field with its most recent result in the Word document by unlinking the field using the `unlink` API. When you unlink a field, its current result is converted to text or a graphic and can no longer be updated automatically.

The following code example shows how to unlink the fields in Word document.

#### JAVA

```
//Creates an instance of WordDocument class.
WordDocument document = new WordDocument();
//Adds a new section into the Word Document.
IWSection section = document.addSection();
//Adds a new paragraph into Word document and appends text into paragraph.
```



```

IWParagraph paragraph = section.addParagraph();
paragraph.appendText("Today's Date: ");
//Adds the new Date field in Word document with field name and its type.
WField field = (WField)paragraph.appendField("Date", FieldType.FieldDate);
//Updates the field.
field.update();
//Unlink the field.
field.unlink();
//Saves the document in the given name and format.
document.save("Sample.docx", FormatType.Docx);
//Releases the resources occupied by WordDocument instance.
document.close();

```

**Note:** XE (Index Entry) fields cannot be unlinked.

### Sequence Field

You can use the Sequence (SEQ) field to automatically numbers the chapters, tables, figures, and other items in a Word document. When you add, delete, or move an item in Word document (along with SEQ fields), you can update the remaining SEQ fields with a new sequence.

You can format the SEQ field using below switches.

\c -- Repeats the closest preceding sequence number.

\h -- Hides the field result unless a general-formatting-switch is also present.

\n -- Inserts the next sequence number for the specified items. This is the default switch.

\r -- Resets the sequence number to the number following "r".

\s -- Resets the sequence number at the heading level following the "s".

### Apply Number format

You can apply the number format for the sequence field using `NumberFormat` property.

The following code example shows how to apply the number format for sequence field.

### JAVA

```

//Creates a new document.
WordDocument document = createDocument();
//Accesses sequence field in the document.
WSeqField field =
(WSeqField) ((WParagraph) document.getLastSection().getBody().getChildEntities()
.get(0)).getChildEntities().get(0);
//Applies the number format for sequence field.
field.setNumberFormat(CaptionNumberingFormat.Roman);
//Accesses sequence field in the document.
field =
(WSeqField) ((WParagraph) document.getLastSection().getBody().getChildEntities()
.get(1)).getChildEntities().get(0);
//Applies the number format for sequence field.
field.setNumberFormat(CaptionNumberingFormat.Roman);
//Accesses sequence field in the document.
field =
(WSeqField) ((WParagraph) document.getLastSection().getBody().getChildEntities()
.get(2)).getChildEntities().get(0);
//Applies the number format for sequence field.

```



```
field.setNumberFormat(CaptionNumberingFormat.Roman);  
//Updates the document fields.  
document.updateDocumentFields();  
//Saves and closes the Word document.  
document.save("Sample.docx");  
document.close();
```

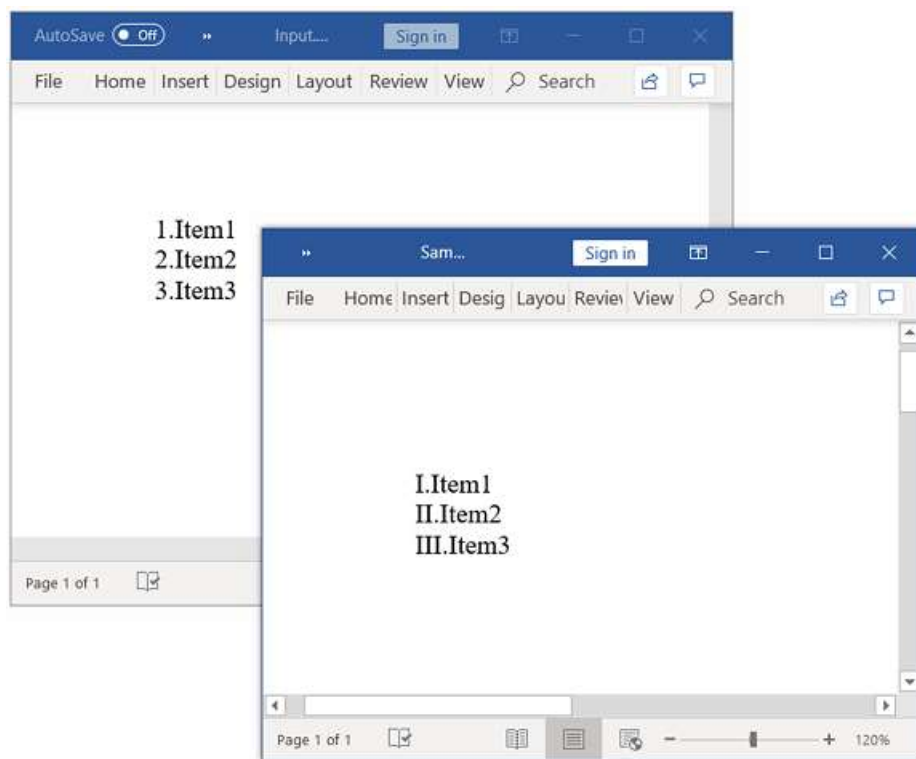
The following code example provides supporting methods for the above code.

#### JAVA

```
private WordDocument createDocument() throws Exception  
{  
    //Creates a new document.  
    WordDocument document = new WordDocument();  
    //Adds a new section to the document.  
    IWSection section = document.addSection();  
    //Sets margin of the section.  
    section.getPageSetup().getMargins().setAll((float)72);  
    //Adds a paragraph to the section.  
    IWParagraph paragraph = section.addParagraph();  
    paragraph.appendField("List",FieldType.FieldSequence);  
    paragraph.appendText(".Item1");  
    //Adds a paragraph to the section.  
    paragraph = section.addParagraph();  
    paragraph.appendField("List",FieldType.FieldSequence);  
    paragraph.appendText(".Item2");  
    //Adds a paragraph to the section.  
    paragraph = section.addParagraph();  
    paragraph.appendField("List",FieldType.FieldSequence);  
    paragraph.appendText(".Item3");  
    return document;  
}
```

By executing the above code example, it generates output Word document as follows.





### Refer Bookmark

You can refer the sequence field elsewhere in the document by including bookmark name through **BookmarkName** property. The referred sequence field may be present anywhere in the Word document, but not beyond the respective bookmark end.

For example, you can refer the image caption numbers in sentence by including the bookmark name in sequence field.

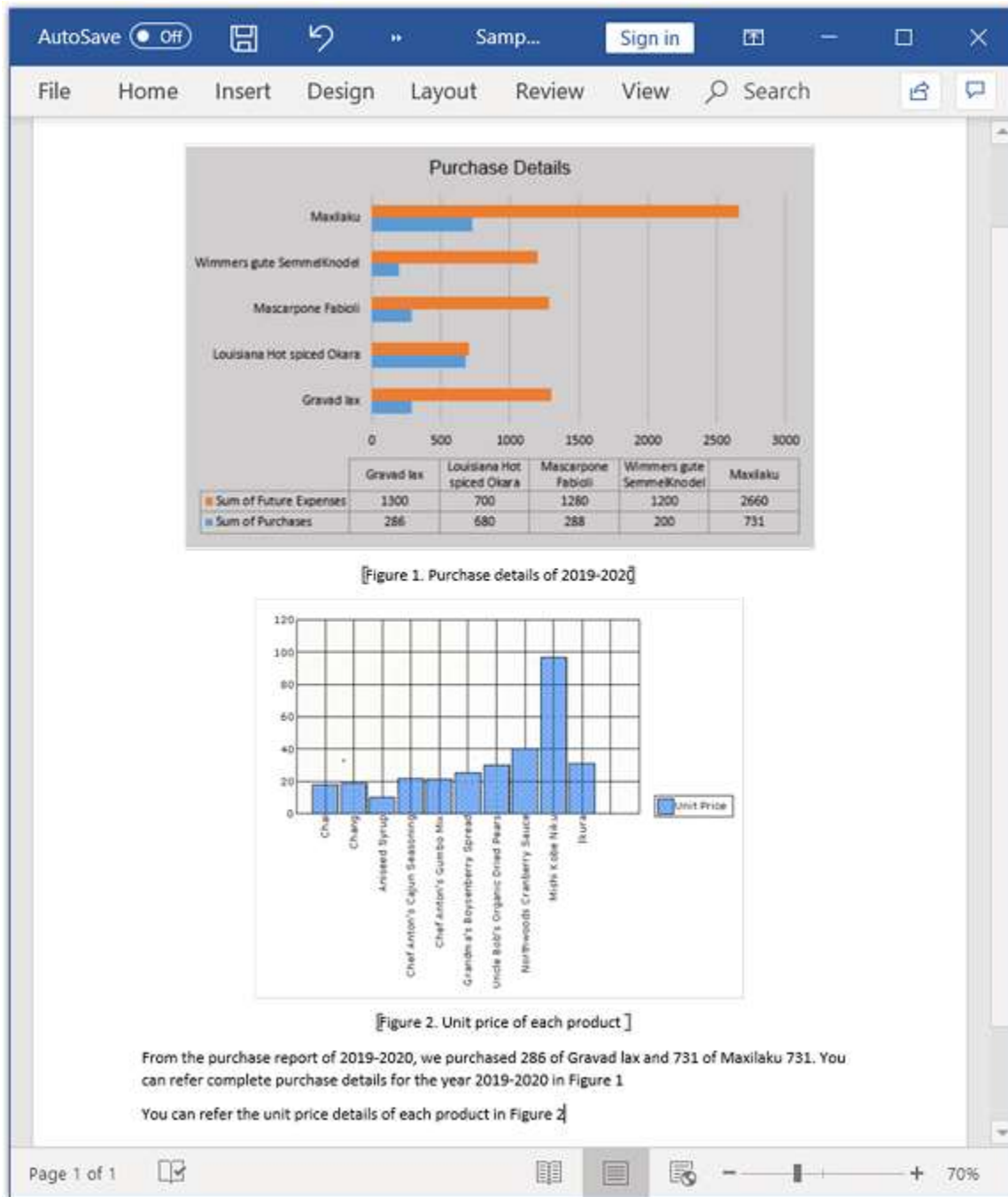
The following code example shows how to refer the bookmark in sequence field.

### JAVA

```
//Opens an existing word document.
WordDocument document = new WordDocument("Template.docx");
//Accesses sequence field in the document.
WParagraph paragraph =
(WParagraph)document.getLastSection().getBody().getChildEntities().get(4);
WSeqField seqField = (WSeqField)paragraph.getChildEntities().get(12);
//Adds bookmark reference to the sequence field.
seqField.setBookmarkName("BkmkPurchase");
//Accesses sequence field in the document.
paragraph = (WParagraph)document.getLastSection().getParagraphs().get(5);
seqField=(WSeqField)paragraph.getChildEntities().get(1);
//Adds bookmark reference to the sequence field.
seqField.setBookmarkName("BkkmUnitPrice");
//Updates the document fields.
document.updateDocumentFields();
//Saves and closes the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```



By executing the above code example, it generates output Word document as follows.



### Reset numbering

You can reset the numbering for sequence field (\r) using `ResetNumber` property and reset the numbering based on heading level (\s) in the Word document using `ResetHeadingLevel` property.

The following code example shows how to reset the numbering for sequence field.

### JAVA

```
//Creates a Word document.
```



```

WordDocument document = createDocument();
//Accesses sequence field in the document.
IWTable table =
(WTable)document.getLastSection().getBody().getChildEntities().get(1);
WSeqField field =
((WSeqField)((WParagraph)table.get(0,1).getChildEntities().get(0)).getChildE
ntities().get(1));
//Resets the number for sequence field.
field.setResetNumber(1001);
//Accesses sequence field in the document.
field =
((WSeqField)((WParagraph)table.get(1,1).getChildEntities().get(0)).getChildE
ntities().get(1));
//Resets the number for sequence field.
field.setResetNumber(1002);
//Accesses sequence field in the document.
field =
((WSeqField)((WParagraph)table.get(2,1).getChildEntities().get(0)).getChildE
ntities().get(1));
//Resets the number for sequence field.
field.setResetNumber(1003);
table =
(WTable)document.getLastSection().getBody().getChildEntities().get(3);
field =
((WSeqField)((WParagraph)table.get(0,1).getChildEntities().get(1)).getChildE
ntities().get(1));
//Resets the heading level for sequence field.
field.setResetHeadingLevel(1);
//Updates the document fields.
document.updateDocumentFields();
//Saves and closes the Word document.
document.save("Sample.docx");
document.close();

```

The following code example provides supporting methods for the above code.

#### **JAVA**

```

private WordDocument createDocument() throws Exception
{
//Creates a new word document.
WordDocument document = new WordDocument();
//Adds new section to the document.
IWSection section = document.addSection();
//Sets margin of the section.
section.getPageSetup().getMargins().setAll((float)72);
//Adds new paragraph to the section.
IWParagraph paragraph = (WParagraph)section.addParagraph();
//Adds text range.
IWTextRange textRange = paragraph.appendText("Adventure Works Cycles");
textRange.getCharacterFormat().setFontSize((float)16);
textRange.getCharacterFormat().setBold(true);
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Ce
nter);
//Adds a new table into Word document.
IWTable table = section.addTable();

```



```

//Specifies the total number of rows & columns.
table.resetCells(3,2);
//First row.
table.get(0,0).addParagraph().appendPicture(new
FileInputStream("Nancy.png"));
paragraph = table.get(0,1).addParagraph();
paragraph.appendText("Employee Id: ");
paragraph.appendField("Id",FieldType.FieldSequence);
table.get(0,1).addParagraph().appendText("Name: Nancy Davolio");
table.get(0,1).addParagraph().appendText("Title: Sales Representative");
table.get(0,1).addParagraph().appendText("Address: 507 - 20th Ave. E.");
table.get(0,1).addParagraph().appendText("Zip Code: 98122");
//Second row.
table.get(1,0).addParagraph().appendPicture(new
FileInputStream("Andrews.png"));
paragraph = table.get(1,1).addParagraph();
paragraph.appendText("Employee ID: ");
paragraph.appendField("Id",FieldType.FieldSequence);
table.get(1,1).addParagraph().appendText("Name: Andrew Fuller");
table.get(1,1).addParagraph().appendText("Title: Vice President, Sales");
table.get(1,1).addParagraph().appendText("Address1: 908 W. Capital Way, ");
table.get(1,1).addParagraph().appendText("TacomaWA USA");
//Third row.
table.get(2,0).addParagraph().appendPicture(new
FileInputStream("Janet.png"));
paragraph = table.get(2,1).addParagraph();
paragraph.appendText("Employee ID: ");
paragraph.appendField("Id",FieldType.FieldSequence);
table.get(2,1).addParagraph().appendText("Name: Janet Leverling");
table.get(2,1).addParagraph().appendText("Title: Sales Representative");
table.get(2,1).addParagraph().appendText("Address1: 722 Moss Bay Blvd, ");
table.get(2,1).addParagraph().appendText("KirklandWA USA");
//Adds new Paragraph to the section.
paragraph = section.addParagraph();
paragraph.appendBreak(BreakType.PageBreak);
//Adds text range.
paragraph.appendText("Product Overview");
paragraph.applyStyle(BuiltinStyle.Heading1);
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Ju
stify);
//Adds a new table into Word document
table = section.addTable();
//Specifies the total number of rows & columns
table.resetCells(3,2);
//Accesses the instance of the cell and adds the content into cell.
//First row.
table.get(0,0).addParagraph().appendPicture(new FileInputStream("Mountain-
200.png"));
table.get(0,1).addParagraph().appendText("Mountain-200");
paragraph = table.get(0,1).addParagraph();
paragraph.appendText("Product No: ");
paragraph.appendField("Id",FieldType.FieldSequence);
table.get(0,1).addParagraph().appendText("Size: 38");
table.get(0,1).addParagraph().appendText("Weight: 25");
table.get(0,1).addParagraph().appendText("Price: $2,294.99");
//Second row.
table.get(1,0).addParagraph().appendText("Mountain-300");

```

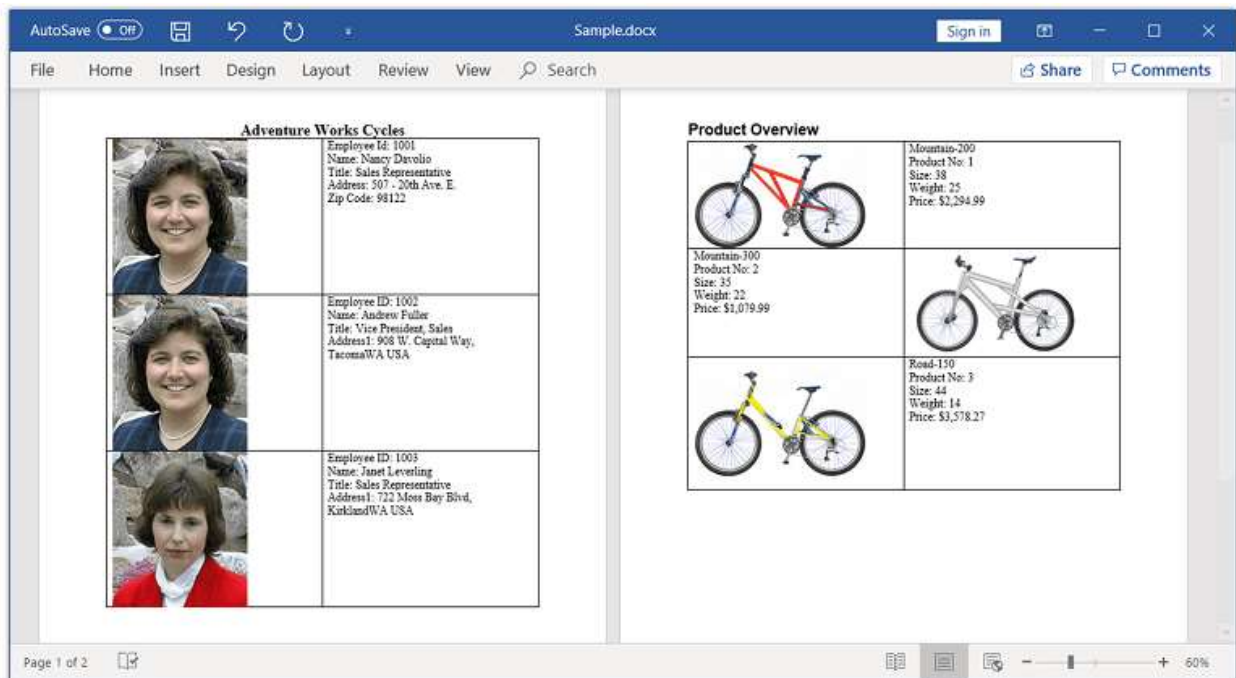


```

paragraph = table.get(1,0).addParagraph();
paragraph.appendText("Product No: ");
paragraph.appendField("Id",FieldType.FieldSequence);
table.get(1,0).addParagraph().appendText("Size: 35");
table.get(1,0).addParagraph().appendText("Weight: 22");
table.get(1,0).addParagraph().appendText("Price: $1,079.99");
table.get(1,1).addParagraph().appendPicture(new FileInputStream("Mountain-
300.png"));
table.get(2,0).addParagraph().appendPicture(new FileInputStream("Road-
550.png"));
table.get(2,1).addParagraph().appendText("Road-150");
//Third row.
paragraph = table.get(2,1).addParagraph();
paragraph.appendText("Product No: ");
paragraph.appendField("Id",FieldType.FieldSequence);
table.get(2,1).addParagraph().appendText("Size: 44");
table.get(2,1).addParagraph().appendText("Weight: 14");
table.get(2,1).addParagraph().appendText("Price: $3,578.27");
return document;
}

```

By executing the above code example, it generates output Word document as follows.



### Repeat nearest number

You can insert the closest preceding sequence number (\c) using `RepeatNearestNumber` property.

For example, if you need to display total number of products in a page, you can repeat the closest preceding sequence number which referred for products.

The following code example shows how to repeat the closest preceding sequence number in the Word document.

### JAVA



```
//Creates a Word document.
WordDocument document = createDocument();
//Accesses sequence field in the document.
WSeqField field =
(WSeqField)((WParagraph)document.getLastSection().getHeadersFooters().getHeader().getChildEntities().get(0)).getChildEntities().get(1);
//Enables a flag to repeat the nearest number for sequence field.
field.setRepeatNearestNumber(true);
//Updates the document fields.
document.updateDocumentFields();
//Saves and closes the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

The following code example provides supporting methods for the above code.

### **JAVA**

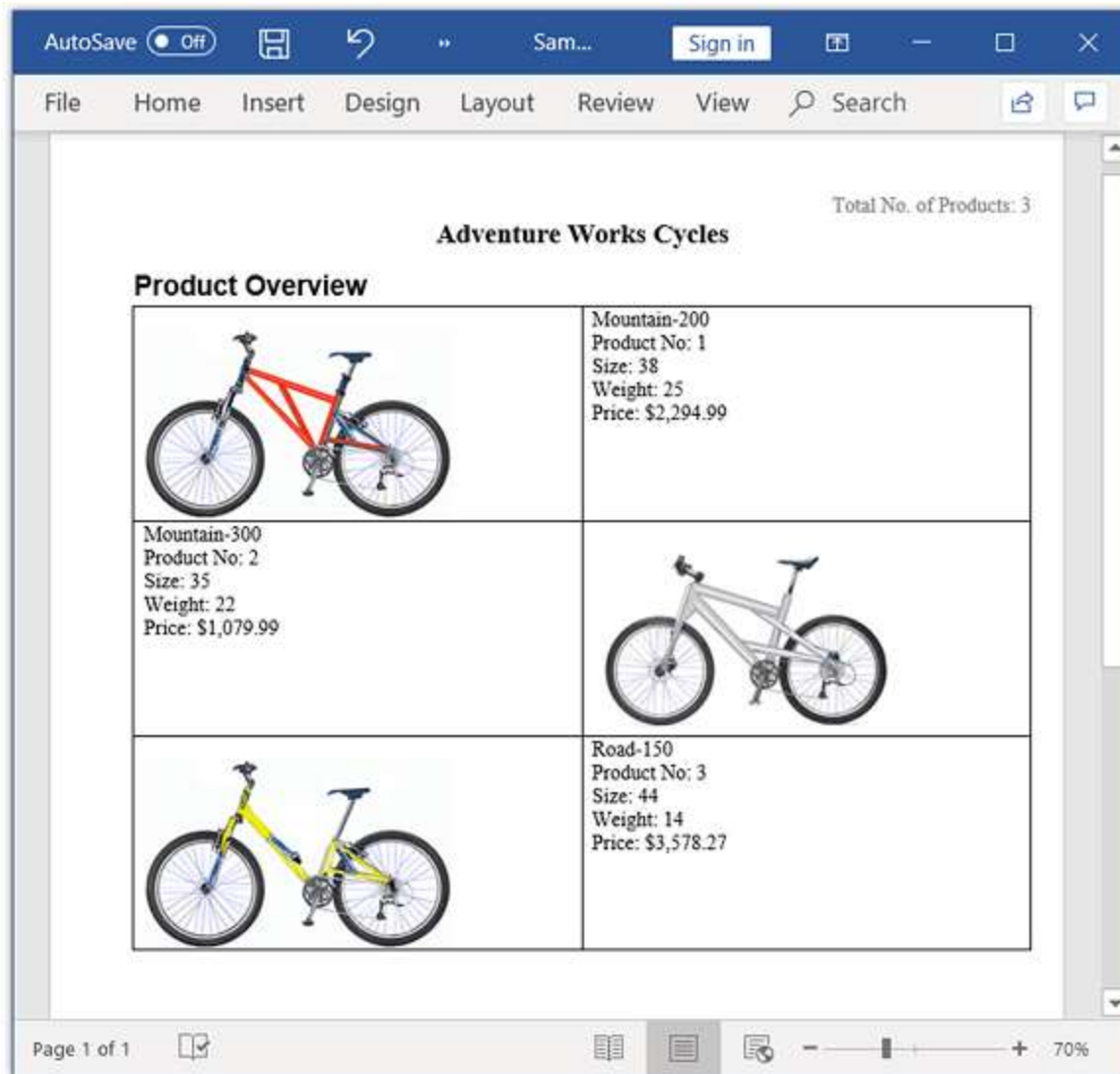
```
private WordDocument createDocument() throws Exception
{
//Creates a new document.
WordDocument document = new WordDocument();
//Adds a new section to the document
IWSection section = document.addSection();
//Inserts the default page header.
IWParagraph paragraph =
section.getHeadersFooters().getOddHeader().addParagraph();
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Right);
paragraph.appendText("Total No. of Products: ");
paragraph.appendField("Product count",FieldType.FieldSequence);
//Adds a paragraph to the section.
paragraph = section.addParagraph();
IWTextRange textRange = paragraph.appendText("Adventure Works Cycles");
paragraph.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Center);
textRange.getCharacterFormat().setFontSize((float)16);
textRange.getCharacterFormat().setBold(true);
//Adds a paragraph to the section.
section.addParagraph().appendText("Product Overview");
document.getLastParagraph().applyStyle(BuiltinStyle.Heading1);
//Adds a new table into Word document
IWTable table = section.addTable();
//Specifies the total number of rows & columns.
table.resetCells(3,2);
//Accesses the instance of the cell and adds the content into cell
//First row.
table.get(0,0).addParagraph().appendPicture(new FileInputStream("Mountain-200.png"));
table.get(0,1).addParagraph().appendText("Mountain-200");
paragraph=table.get(0,1).addParagraph();
paragraph.appendText("Product No: ");
paragraph.appendField("Product count",FieldType.FieldSequence);
table.get(0,1).addParagraph().appendText("Size: 38");
table.get(0,1).addParagraph().appendText("Weight: 25");
table.get(0,1).addParagraph().appendText("Price: $2,294.99");
```



```
//Second row.
table.get(1,0).addParagraph().appendText("Mountain-300");
paragraph=table.get(1,0).addParagraph();
paragraph.appendText("Product No: ");
paragraph.appendField("Product count",FieldType.FieldSequence);
table.get(1,0).addParagraph().appendText("Size: 35");
table.get(1,0).addParagraph().appendText("Weight: 22");
table.get(1,0).addParagraph().appendText("Price: $1,079.99");
table.get(1,1).addParagraph().appendPicture(new FileInputStream("Mountain-
300.png"));
//Third row.
table.get(2,0).addParagraph().appendPicture(new FileInputStream("Road-
550.png"));
table.get(2,1).addParagraph().appendText("Road-150");
paragraph=table.get(2,1).addParagraph();
paragraph.appendText("Product No: ");
paragraph.appendField("Product count",FieldType.FieldSequence);
table.get(2,1).addParagraph().appendText("Size: 44");
table.get(2,1).addParagraph().appendText("Weight: 14");
table.get(2,1).addParagraph().appendText("Price: $3,578.27");
return document;
}
```

By executing the above code example, it generates output Word document as follows.





### Hide

You can hide the field result of the sequence field using `HideResult` property.

For example, if you need to consider the sequence numbering for list of products, but not need to print the numbering for retired products, then you can hide those sequence fields alone.

The following code example shows how to hide the field result of sequence field.

### JAVA

```
//Creates a Word document.
WordDocument document = createDocument();
//Accesses sequence field in the document.
WTable table =
(WTable)document.getLastSection().getBody().getChildEntities().get(1);
WSeqField field =
((WSeqField)((WParagraph)table.get(2,1).getChildEntities().get(0)).getChildren().get(0));
//Enables a flag to to hide the sequence field result.
field.setHideResult(true);
```



```
//Accesses sequence field in the document.
field=((WSeqField)((WParagraph)table.get(4,1).getChildEntities().get(0)).get
ChildEntities().get(0));
//Enables a flag to hide the sequence field result.
field.setHideResult(true);
//Updates the document fields.
document.updateDocumentFields();
//Saves and closes the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

The following code example provides supporting methods for the above code.

### **JAVA**

```
private WordDocument createDocument() throws Exception
{
    //Creates a new Word document.
    WordDocument document = new WordDocument();
    //Adds a new section to the document.
    IWSection section = document.addSection();
    //Adds a paragraph to the section.
    IWParagraph paragraph = section.addParagraph();
    paragraph.appendText("Syncfusion Product Details");
    paragraph.applyStyle(BuiltinStyle.Heading1);
    //Adds a new table .
    IWTable table = section.addTable();
    //Specifies the total number of rows & columns
    table.resetCells(6,4);
    //Accesses the instance of the cell and add the content into cell.
    //First row.
    table.get(0,0).addParagraph().appendText("S.No");
    table.get(0,1).addParagraph().appendText("Platform Id");
    table.get(0,2).addParagraph().appendText("Platform");
    table.get(0,3).addParagraph().appendText("Status ");
    table.get(1,0).addParagraph().appendText("1.");
    //Second row.
    table.get(1,1).addParagraph().appendField("PlatformCount",FieldType.FieldSequence);
    table.get(1,2).addParagraph().appendText("ASP.NET Core");
    table.get(1,3).addParagraph().appendText("Live");
    //Third row.
    table.get(2,0).addParagraph().appendText("2.");
    table.get(2,1).addParagraph().appendField("PlatformCount",FieldType.FieldSequence);
    table.get(2,2).addParagraph().appendText("LightSwitch");
    table.get(2,3).addParagraph().appendText("Retired");
    //Fourth row.
    table.get(3,0).addParagraph().appendText("3.");
    table.get(3,1).addParagraph().appendField("PlatformCount",FieldType.FieldSequence);
    table.get(3,2).addParagraph().appendText("ASP.NET MVC");
    table.get(3,3).addParagraph().appendText("Live");
    //Fifth row.
    table.get(4,0).addParagraph().appendText("4.");
```

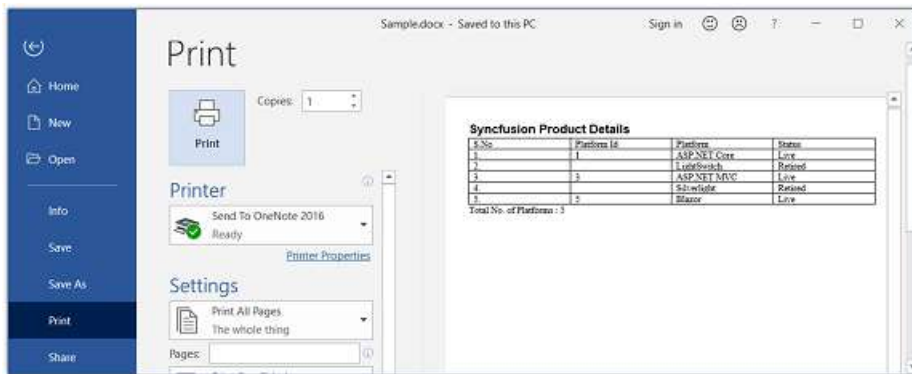


```

table.get(4,1).addParagraph().appendField("PlatformCount",FieldType.FieldSequence);
table.get(4,2).addParagraph().appendText("Silverlight ");
table.get(4,3).addParagraph().appendText("Retired");
//Sixth row.
table.get(5,0).addParagraph().appendText("5.");
table.get(5,1).addParagraph().appendField("PlatformCount",FieldType.FieldSequence);
table.get(5,2).addParagraph().appendText("Blazor");
table.get(5,3).addParagraph().appendText("Live");
section.addParagraph().appendText("Total No. of Platforms : 5");
return document;
}

```

By executing the above code example, it generates output Word document as follows.



#### Next Sequence number

You can insert the next sequence number for the specified items using `InsertNextNumber` property. This is the default.

The following code example shows how to insert the next sequence number for the specified item.

#### JAVA

```

//Opens an existing word document.
WordDocument document = new WordDocument("Template.docx");
//Accesses sequence field in the document.
WParagraph paragraph =
(WParagraph)document.getLastSection().getBody().getChildEntities().get(4);
WSeqField field = (WSeqField)paragraph.getChildEntities().get(12);
//Enables a flag to insert next number for sequence field.
field.setInsertNextNumber(true);
//Updates the document fields.
document.updateDocumentFields();
//Saves and closes the Word document.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

#### Working with Find and Replace

You can search a particular text you like to change and replace it with another text or part of the document.



### Finding contents in a Word document

You can find the first occurrence of a particular text within a single paragraph in the document by using `Find` method and its next occurrence by using `FindNext` method. You can also find a particular text pattern in the document.

The following code example illustrates how to find a particular text and its next occurrence in the document.

#### JAVA

```
//Loads the template document
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Finds the first occurrence of a particular text in the document
TextSelection textSelection = document.find("as graphical contents", false,
true);
//Gets the found text as single text range
WTextRange textRange = textSelection.getAsOneRange();
//Modifies the text
textRange.setText("Replaced text");
//Sets highlight color
textRange.getCharacterFormat().setHighlightColor(ColorSupport.getYellow());
//Finds the next occurrence of a particular text from the previous paragraph
textSelection = document.findNext(textRange.getOwnerParagraph(),
"paragraph", true, false);
//Gets the found text as single text range
WTextRange range = textSelection.getAsOneRange();
//Sets bold formatting
range.getCharacterFormat().setBold(true);
//Saves and closes the document
document.save("Sample.docx", FormatType.Docx);
document.close();
```

You can find all the occurrence of a particular text within a single paragraph in the document by using `FindAll` method.

The following code example illustrates how to find all the occurrences of a particular text in the document.

#### JAVA

```
//Loads the template document
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Finds all the occurrences of a particular text
TextSelection[] textSelections = document.findAll("paragraph", false, true);
for(Object textSelection_tempObj : textSelections)
{
TextSelection textSelection = (TextSelection)textSelection_tempObj;
WTextRange textRange = textSelection.getAsOneRange();
textRange.getCharacterFormat().setHighlightColor(ColorSupport.getYellowGreen
());
}
//Saves and closes the document
document.save("Sample.docx", FormatType.Docx);
document.close();
```



You can find the first occurrence of a particular text extended to several paragraphs in the document by using `FindSingleLine` method and its next occurrence by using `FindNextSingleLine` method.

The following code example illustrates how to find a particular text extended to several paragraphs in the Word document.

#### JAVA

```
//Loads the template document
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Finds the first occurrence of a particular text extended to several
paragraphs in the document
TextSelection[] textSelections = document.findSingleLine("First paragraph
Second paragraph", true, false);
WParagraph paragraph = null;
for(Object textSelection_tempObj : textSelections)
{
    //Gets the found text as single text range and set highlight color
    TextSelection textSelection = (TextSelection)textSelection_tempObj;
    WTextRange textRange = textSelection.getAsOneRange();
    textRange.getCharacterFormat().setHighlightColor(ColorSupport.getYellowGreen
());
    paragraph=textRange.getOwnerParagraph();
}
//Finds the next occurrence of a particular text extended to several
paragraphs in the document
textSelections=document.findNextSingleLine(paragraph,"First paragraph Second
paragraph",true,false);
for(Object textSelection_tempObj : textSelections)
{
    //Gets the found text as single text range and sets italic formatting
    TextSelection textSelection = (TextSelection)textSelection_tempObj;
    WTextRange text = textSelection.getAsOneRange();
    text.getCharacterFormat().setItalic(true);
}
//Saves and closes the document
document.save("Sample.docx", FormatType.Docx);
document.close();
```

#### Find and replace text with other text

You can find text in a Word document and replace it with other text. Unlike the `find` method, the `replace` method replaces all occurrences of the text. You can customize it to replace only the first occurrence of a text by setting the `setReplaceFirst` property of the `WordDocument` class to true.

The following code example illustrates how to replace all occurrences of a misspelled word with the correctly spelled word.

#### JAVA

```
// Opens the input Word document
WordDocument document = new WordDocument("Template.docx");
// Finds all occurrences of a misspelled word and replaces with properly
spelled word
document.replace("Cyles", "Cycles", true, true);
//Saves and closes the document
document.save("Sample.docx");
```



```
document.close();
```

### Find and replace text with an image

You can find placeholder text in a Word document and replace it with any desired image.

The following code example illustrates how to find and replace text in a word document with an image

#### JAVA

```
//Opens the input Word document
WordDocument document = new WordDocument("Template.docx");
//Finds all the image placeholder text in the Word document
TextSelection[] textSelections =
document.findAll(Pattern.compile(MatchSupport.trimPattern("^/(.*)")));
for (int i = 0; i < textSelections.length; i++)
{
    // Replaces the image placeholder text with desired image
    WParagraph paragraph = new WParagraph(document);
    WPicture picture = (WPicture) paragraph.appendPicture(new
    FileInputStream(textSelections[i].getSelectedText() + ".png"));
    TextSelection newSelection = new TextSelection(paragraph, 0, 1);
    TextBodyPart bodyPart = new TextBodyPart(document);
    bodyPart.getBodyItems().add(paragraph);
    document.replace(textSelections[i].getSelectedText(), bodyPart, true, true);
}
//Saves and closes the document
document.save("Sample.docx");
document.close();
```

### Find and replace a pattern of text with a merge field

You can find and replace a pattern of text in a Word document with merge fields using Regex.

The following code example illustrates how to create a mail merge template by replacing a pattern of text (enclosed within '«' and '»') in a Word document with the desired merge fields.

#### JAVA

```
// Opens the input Word document
WordDocument document = new WordDocument("Template.docx");
// Finds all the placeholder text enclosed within '«' and '»' in the Word
document
TextSelection[] textSelections = document.findAll(
Pattern.compile(MatchSupport.trimPattern("«([?i)image(?-i)]*:[a-zA-Z0-9
]*:[a-zA-Z0-9 ]+)»")));
String[] searchedPlaceholders = new String[textSelections.length];
for (int i = 0; i < textSelections.length; i++)
{
    searchedPlaceholders[(int) i] = textSelections[(int) i].getSelectedText();
}
for (int i = 0; i < searchedPlaceholders.length; i++)
{
    WParagraph paragraph = new WParagraph(document);
    // Replaces the placeholder text enclosed within '«' and '»' with desired
    merge field
    paragraph.appendField(StringSupport.trimEnd(StringSupport.trimStart(searched
    Placeholders[i], '«'), '»'), FieldType.FieldMergeField);
}
```



```

TextSelection newSelection = new TextSelection(paragraph, 0,
paragraph.getItems().getCount());
TextBodyPart bodyPart = new TextBodyPart(document);
bodyPart.getBodyItems().add(paragraph);
document.replace(searchedPlaceholders[(int) i], bodyPart, true, true, true);
}
//Saves and closes the document
document.save("Sample.docx");
document.close();

```

### Find and replace text with a table

You can find placeholder text in a Word document and replace it with a table.

The following code example illustrates how to do this.

#### JAVA

```

// Opens the input Word document
WordDocument document = new WordDocument("Template.docx");
// Creates a new table
WTable table = new WTable(document);
table.resetCells(1, 6);
table.get(0, 0).setWidth(52f);
table.get(0, 0).addParagraph().appendText("Supplier ID");
table.get(0, 1).setWidth(128f);
table.get(0, 1).addParagraph().appendText("Company Name");
table.get(0, 2).setWidth(70f);
table.get(0, 2).addParagraph().appendText("Contact Name");
table.get(0, 3).setWidth(92f);
table.get(0, 3).addParagraph().appendText("Address");
table.get(0, 4).setWidth(66.5f);
table.get(0, 4).addParagraph().appendText("City");
table.get(0, 5).setWidth(56f);
table.get(0, 5).addParagraph().appendText("Country");
// Imports data to the table
importDataToTable(table);
// Applies the built-in table style (Medium Shading 1 Accent 1) to the table
table.applyStyle(BuiltinTableStyle.MediumShading1Accent1);
TextBodyPart bodyPart = new TextBodyPart(document);
bodyPart.getBodyItems().add(table);
// Replaces the table placeholder text with a new table
document.replace("[Suppliers table]", bodyPart, true, true, true);
// Saves and closes the document
document.save("Sample.docx");

```

The following code example provides supporting method for the above code.

#### JAVA

```

private void importDataToTable(WTable table) throws Exception
{
    FileStreamSupport fs = new FileStreamSupport("Suppliers.xml", FileMode.Open,
    FileAccess.Read);
    XmlReaderSupport reader = XmlReaderSupport.create(fs);
    if (reader == null)
    throw new Exception("reader");
}

```



```

while (reader.getNodeType().getEnumValue() !=
XmlNodeType.Element.getEnumValue())
reader.read();
if (reader.getLocalName() != "SuppliersList")
throw new Exception(StringSupport.concat("Unexpected xml tag ",
reader.getLocalName()));
reader.read();
while (reader.getNodeType().getEnumValue() ==
XmlNodeType.Whitespace.getEnumValue())
reader.read();
while (reader.getLocalName() != "SuppliersList")
{
if (reader.getNodeType().getEnumValue() ==
XmlNodeType.Element.getEnumValue())
{
switch (reader.getLocalName())
{
case "Suppliers":
WTableRow tableRow = table.addRow(true);
importDataToRow(reader, tableRow);
break;
}
}
else
{
reader.read();
if ((reader.getLocalName() == "SuppliersList") && reader.getNodeType() ==
XmlNodeType.EndElement)
break;
}
}
reader.close();
fs.close();
}

```

The following code example provides supporting method for the above code.

#### JAVA

```

private void importDataToRow(XmlReaderSupport reader, WTableRow tableRow)
throws Exception
{
if (reader == null)
throw new Exception("reader");
while (reader.getNodeType().getEnumValue() !=
XmlNodeType.Element.getEnumValue())
reader.read();
if (reader.getLocalName() != "Suppliers")
throw new Exception(StringSupport.concat("Unexpected xml tag ",
reader.getLocalName()));
reader.read();
while (reader.getNodeType().getEnumValue() ==
XmlNodeType.Whitespace.getEnumValue())
reader.read();
while (reader.getLocalName() != "Suppliers")
{

```



```

if (reader.getNodeType().getEnumValue() ==
XmlNodeType.Element.getEnumValue())
{
switch (reader.getLocalName())
{
case "SupplierID":
tableRow.getCells().get(0).addParagraph().appendText(reader.readContentAsString());
break;
case "CompanyName":
tableRow.getCells().get(1).addParagraph().appendText(reader.readContentAsString());
break;
case "ContactName":
tableRow.getCells().get(2).addParagraph().appendText(reader.readContentAsString());
break;
case "Address":
tableRow.getCells().get(3).addParagraph().appendText(reader.readContentAsString());
break;
case "City":
tableRow.getCells().get(4).addParagraph().appendText(reader.readContentAsString());
break;
case "Country":
tableRow.getCells().get(5).addParagraph().appendText(reader.readContentAsString());
break;
default:
reader.skip();
break;
}
}
else
{
reader.read();
if ((reader.getLocalName() == "Suppliers") && reader.getNodeType() ==
XmlNodeType.EndElement)
break;
}
}
}

```

### Find and replace text in Word document with another document

You can find and replace text with another Word document.

The following code example illustrates how to merge or combine Word documents by replacing text with another document (the content of a subheading).

#### JAVA

```

// Opens the Word template document
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
TextSelection[] textSelections =
document.findAll(Pattern.compile(MatchSupport.trimPattern("\\[(.*)\\]")));

```



```

for (int i = 0; i < textSelections.length; i++)
{
    WordDocument subDocument = new
    WordDocument(StringSupport.trimEnd(StringSupport.trimStart(textSelections[i]
    .getSelectedText(), '['), ']') + ".docx", FormatType.Docx);
    document.replace(textSelections[(int) i].getSelectedText(), subDocument,
    true, true);
    subDocument.close();
}
// Saves the Word document
document.save("Sample.docx");
// Closes the document
document.close();

```

### Find and replace text extending to several paragraphs

Apart from finding text in a paragraph, you can also find and replace text that extends to several paragraphs in a Word document. You can find the first occurrence of the text that extends to several paragraphs by using the `findSingleLine` method. Find the next occurrences of the text by using the `findNextSingleLine` method. Similarly, you can replace text that extends to several paragraphs by using `replaceSingleLine` method.

The following code example illustrates how to replace text that extends to several paragraphs.

#### JAVA

```

// Opens the input Word document
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
WordDocument subDocument = new WordDocument("Source.docx", FormatType.Docx);
// Gets the content from another Word document to replace
TextBodyPart replacePart = new TextBodyPart(subDocument);
for (Object bodyItem_tempObj :
subDocument.getLastSection().getBody().getChildEntities())
{
    TextBodyItem bodyItem = (TextBodyItem) bodyItem_tempObj;
    replacePart.getBodyItems().add(bodyItem.clone());
}
String placeholderText = "Suppliers/Vendors of Northwind" + "Customers of
Northwind" + "Employee details of Northwind traders" + "The product
information" + "The inventory details" + "The shippers" + "Purchase Order
transactions" + "Sales Order transaction" + "Inventory transactions" +
"Invoices" + "[end replace]";
// Finds the text that extends to several paragraphs and replaces it with
desired content.
document.replaceSingleLine(placeholderText, replacePart, false, false);
subDocument.close();
// Saves the Word document
document.save("Sample.docx");
// Closes the document
document.close();

```

### Find text in a Word document and format

You can find text in a Word document and format or highlight it. You can find the first occurrence of text using the `find` method. Find the next occurrences of the text using the `findNext` method.



The following code example illustrates how to find all occurrences of a length of text and highlight it.

#### **JAVA**

```
// Opens the input Word document
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
// Finds all occurrence of the text in the Word document
TextSelection[] textSelections = document.findAll("Adventure", true, true);
for (int i = 0; i < textSelections.length; i++)
{
    // Sets the highlight color for the searched text as Yellow
    textSelections[i].getAsOneRange().getCharacterFormat().setHighlightColor(ColorSupport.getYellow());
}
// Saves the Word document
document.save("Sample.docx");
// Closes the document
document.close();
```

### Working with Shapes in Word document

Shapes are drawing objects that include lines, curves, circles, rectangles, etc. It can be preset or custom geometry. You can create and manipulate the pre-defined shape in DOCX and WordML format documents.

#### Adding shapes

The following code example shows how to add pre-defined shape to the document.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
WParagraph paragraph = (WParagraph) section.addParagraph();
//Add new shape to the document.
Shape rectangle = paragraph.appendShape(AutoShapeType.RoundedRectangle, 150, 100);
//Set position for shape.
rectangle.setVerticalPosition(72);
rectangle.setHorizontalPosition(72);
paragraph = (WParagraph) section.addParagraph();
//Add textbody contents to the shape.
paragraph = (WParagraph) rectangle.getTextBody().addParagraph();
IWTextRange text = paragraph.appendText("This text is in rounded rectangle shape");
text.getCharacterFormat().setTextColor(ColorSupport.getGreen());
text.getCharacterFormat().setBold(true);
//Add another shape to the document.
paragraph = (WParagraph) section.addParagraph();
paragraph.appendBreak(BreakType.LineBreak);
Shape pentagon = paragraph.appendShape(AutoShapeType.Pentagon, 100, 100);
paragraph = (WParagraph) pentagon.getTextBody().addParagraph();
paragraph.appendText("This text is in pentagon shape");
//Set position for shape.
```



```

pentagon.setHorizontalPosition(72);
pentagon.setVerticalPosition(200);
//Save and close the Word document instance.
document.save("Result.docx", FormatType.Docx);
document.close();

```

### Format shapes

Shape can have formatting such as line color, fill color, positioning, wrap formats, etc. The following code example illustrates how to apply formatting options for shape.

### JAVA

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
IWParagraph paragraph = (WParagraph)section.addParagraph();
//Append shape to the paragraph.
Shape rectangle = paragraph.appendShape(AutoShapeType.RoundedRectangle, 150,
100);
rectangle.setVerticalPosition(72);
rectangle.setHorizontalPosition(72);
paragraph = (WParagraph)section.addParagraph();
paragraph = (WParagraph)rectangle.getTextBody().addParagraph();
IWTextRange text = paragraph.appendText("This text is in rounded rectangle
shape");
//Apply format to the text.
text.getCharacterFormat().setTextColor(ColorSupport.getGreen());
text.getCharacterFormat().setBold(true);
//Apply fill color for shape.
rectangle.getFillFormat().setFill(true);
rectangle.getFillFormat().setColor(ColorSupport.getLightGray());
//Apply wrap formats.
rectangle.getWrapFormat().setTextWrappingStyle(TextWrappingStyle.Square);
rectangle.getWrapFormat().setTextWrappingType(TextWrappingType.Right);
//Set horizontal and vertical origin.
rectangle.setHorizontalOrigin(HorizontalOrigin.Margin);
rectangle.setVerticalOrigin(VerticalOrigin.Page);
//Set line format.
rectangle.getLineFormat().setDashStyle(LineDashing.Dot);
rectangle.getLineFormat().setColor(ColorSupport.getDarkGray());
//Save and close the Word document instance.
document.save("Result.docx", FormatType.Docx);
document.close();

```

### Rotate shapes

You can rotate the shape and also apply flipping (horizontal and vertical) to it. The following code example explains how to rotate and flip the shape.

### JAVA

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.

```

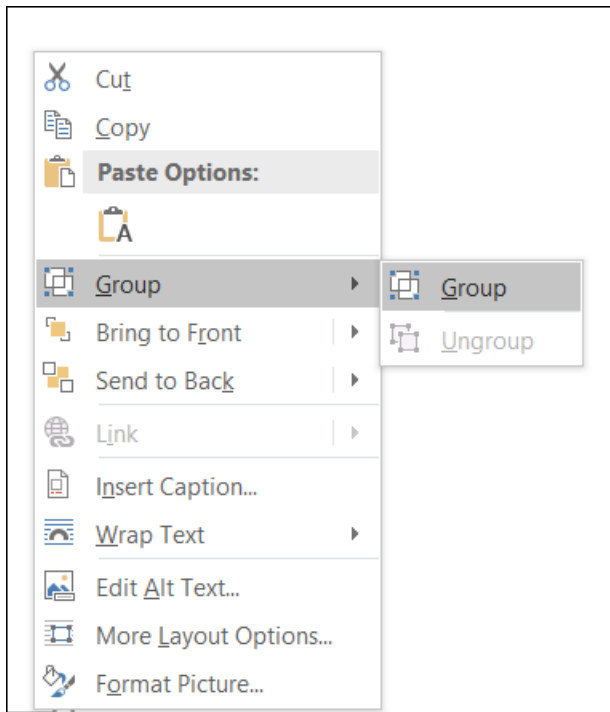


```
IWSection section = document.addSection();  
//Add new paragraph to the section.  
WParagraph paragraph = (WParagraph)section.addParagraph();  
Shape rectangle = paragraph.appendShape(AutoShapeType.RoundedRectangle, 150,  
100);  
//Set position for shape.  
rectangle.setVerticalPosition(72);  
rectangle.setHorizontalPosition(72);  
//Set 90 degree rotation.  
rectangle.setRotation(90);  
//Set horizontal flip.  
rectangle.setFlipHorizontal(true);  
paragraph = (WParagraph)section.addParagraph();  
paragraph = (WParagraph)rectangle.getTextBody().addParagraph();  
IWTextRange text = paragraph.appendText("This text is in rounded rectangle  
shape");  
//Save the Word document.  
document.save("Result.docx", FormatType.Docx);  
//Close the document.  
document.close();
```

### Grouping shapes

Word library now allows you to create or group multiple shapes, pictures, text boxes as a group shape in Word document (DOCX) and preserve it as in DOCX and WordML format conversions.

You can create a document with group shapes by using Microsoft Word. It provides an option to group a set of shapes and images as a single shape and a group shape as individual item.



### Key Features:

1. You can easily manage group of shapes, pictures, text boxes as a group shape.



2. You can move several shapes or images simultaneously and apply the same formatting properties for children of group shapes.

**Note:** 1. While grouping the shapes or other objects, the shapes should be positioned relative to the "Page".

2. While grouping the shapes or other objects, the wrapping style should not be "In Line with Text".

The following code example shows how to create group shape in Word document.

#### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
WParagraph paragraph = (WParagraph)section.addParagraph();
//Create new group shape.
GroupShape groupShape = new GroupShape(document);
//Add group shape to the paragraph.
paragraph.getChildEntities().add(groupShape);
//Create new shape.
Shape shape = new Shape(document, AutoShapeType.RoundedRectangle);
//Set height and width for shape.
shape.setHeight(100);
shape.setWidth(150);
//Set horizontal and vertical position.
shape.setHorizontalPosition(72);
shape.setVerticalPosition(72);
//Set wrapping style for shape.
shape.getWrapFormat().setTextWrappingStyle(TextWrappingStyle.InFrontOfText);
//Set horizontal and vertical origin.
shape.setHorizontalOrigin(HorizontalOrigin.Page);
shape.setVerticalOrigin(VerticalOrigin.Page);
//Add the specified shape to group shape.
groupShape.add(shape);
//Create new picture.
WPicture picture = new WPicture(document);
FileStreamSupport imageStream = new FileStreamSupport("Image.png",
    FileMode.Open, FileAccess.ReadWrite);
picture.loadImage(imageStream.toArray());
//Set wrapping style for picture.
picture.setTextWrappingStyle(TextWrappingStyle.InFrontOfText);
//Set height and width for the image.
picture.setHeight(100);
picture.setWidth(100);
//Set horizontal and vertical position.
picture.setHorizontalPosition(400);
picture.setVerticalPosition(150);
//Set horizontal and vertical origin.
picture.setHorizontalOrigin(HorizontalOrigin.Page);
picture.setVerticalOrigin(VerticalOrigin.Page);
//Add the specified picture to group shape.
groupShape.add(picture);
//Create new textbox.
WTextBox textbox = new WTextBox(document);
```



```

textbox.getTextBoxFormat().setWidth(150);
textbox.getTextBoxFormat().setHeight(75);
//Add new text to the textbox body.
IWParagraph textboxParagraph = textbox.getTextBoxBody().addParagraph();
textboxParagraph.appendText("Text inside text box");
//Set wrapping style for textbox.
textbox.getTextBoxFormat().setTextWrappingStyle(TextWrappingStyle.Behind);
//Set horizontal and vertical position.
textbox.getTextBoxFormat().setHorizontalPosition(200);
textbox.getTextBoxFormat().setVerticalPosition(200);
//Set horizontal and vertical origin.
textbox.getTextBoxFormat().setVerticalOrigin(VerticalOrigin.Page);
textbox.getTextBoxFormat().setHorizontalOrigin(HorizontalOrigin.Page);
//Add the specified textbox to group shape.
groupShape.add(textbox);
//Save and close the Word document instance.
document.save("Result.docx", FormatType.Docx);
document.close();

```

### Nested group shapes

The following code example illustrates how to group the nested group shapes as a group shape in Word document.

### JAVA

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
WParagraph paragraph = (WParagraph)section.addParagraph();
//Create new group shape.
GroupShape groupShape = new GroupShape(document);
//Add group shape to the paragraph.
paragraph.getChildEntities().add(groupShape);
//Append new shape to the document.
Shape shape = new Shape(document, AutoShapeType.RoundedRectangle);
//Set height and width for shape.
shape.setHeight(100);
shape.setWidth(150);
//Set Wrapping style for shape.
shape.getWrapFormat().setTextWrappingStyle(TextWrappingStyle.InFrontOfText);
//Set horizontal and vertical position for shape.
shape.setHorizontalPosition(72);
shape.setVerticalPosition(72);
//Set horizontal and vertical origin for shape.
shape.setHorizontalOrigin(HorizontalOrigin.Page);
shape.setVerticalOrigin(VerticalOrigin.Page);
//Add the specified shape to group shape.
groupShape.add(shape);
//Append new picture to the document.
WPicture picture = new WPicture(document);
//Load image from the file.
FileStreamSupport imageStream = new FileStreamSupport("Image.png",
FileMode.Open, FileAccess.ReadWrite);
picture.loadImage(imageStream.toArray());

```



```
//Set wrapping style for picture.
picture.setTextWrappingStyle(TextWrappingStyle.InFrontOfText);
//Set height and width for the picture.
picture.setHeight(100);
picture.setWidth(100);
//Set horizontal and vertical position for the picture.
picture.setHorizontalPosition(400);
picture.setVerticalPosition(150);
//Set horizontal and vertical origin for the picture.
picture.setHorizontalOrigin(HorizontalOrigin.Page);
picture.setVerticalOrigin(VerticalOrigin.Page);
//Add specified picture to the group shape.
groupShape.add(picture);
//Create new nested group shape.
GroupShape nestedGroupShape = new GroupShape(document);
//Append new textbox to the document.
WTextBox textbox = new WTextBox(document);
//Set width and height for the textbox.
textbox.getTextBoxFormat().setWidth(150);
textbox.getTextBoxFormat().setHeight(75);
//Add new text to the textbox body.
IWParagraph textboxParagraph = textbox.getTextBoxBody().addParagraph();
//Add new text to the textbox paragraph.
textboxParagraph.appendText("Text inside text box");
//Set wrapping style for the textbox.
textbox.getTextBoxFormat().setTextWrappingStyle(TextWrappingStyle.Behind);
//Set horizontal and vertical position for the textbox.
textbox.getTextBoxFormat().setHorizontalPosition(200);
textbox.getTextBoxFormat().setVerticalPosition(200);
//Set horizontal and vertical origin for the textbox.
textbox.getTextBoxFormat().setVerticalOrigin(VerticalOrigin.Page);
textbox.getTextBoxFormat().setHorizontalOrigin(HorizontalOrigin.Page);
//Add specified textbox to the nested group shape.
nestedGroupShape.add(textbox);
//Append new shape to the document.
shape = new Shape(document, AutoShapeType.Oval);
//Set height and width for the new shape.
shape.setHeight(100);
shape.setWidth(150);
//Set horizontal and vertical position for the shape.
shape.setHorizontalPosition(200);
shape.setVerticalPosition(72);
//Set horizontal and vertical origin for the shape.
shape.setHorizontalOrigin(HorizontalOrigin.Page);
shape.setVerticalOrigin(VerticalOrigin.Page);
//Set horizontal and vertical position for the nested group shape.
nestedGroupShape.setHorizontalPosition(72);
nestedGroupShape.setVerticalPosition(72);
//Add specified shape to the nested group shape.
nestedGroupShape.add(shape);
//Add nested group shape to the group shape of the paragraph.
groupShape.add(nestedGroupShape);
//Save and close the Word document instance.
document.save("Result.docx", FormatType.Docx);
document.close();
```



### Ungrouping shapes

You can ungroup the group shapes in the Word document to preserve each shape as individual item.

The following code example shows how to ungroup the group shape in Word document.

#### JAVA

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx",
FormatType.Automatic);
//Get the last paragraph.
WParagraph lastParagraph = document.getLastParagraph();
//Iterate through the paragraph items to get the group shape.
for (int i = 0; i < lastParagraph.getChildEntities().getCount(); i++)
{
if (lastParagraph.getChildEntities().get(i) instanceof GroupShape)
{
GroupShape groupShape = (GroupShape)lastParagraph.getChildEntities().get(i);
//Ungroup the child shapes in the group shape.
groupShape.ungroup();
break;
}
}
//Save and closes the Word document instance.
document.save("Result.docx", FormatType.Docx);
document.close();
```

### Working with Mail merge

Mail merge is a process of merging data from data source to a Word template document. The **WMergeField** class provides support to bind template document and data source. The **WMergeField** instance is replaced with the actual data retrieved from data source for the given merge field name in a template document.

The following data sources are supported by Essential DocIO for performing Mail merge:

- String Arrays
- Java objects

### Mail merge process

The mail merge process involves three documents:

1. **Template Word document:** This document contains the static or templated text and graphics along with the merge fields (that are placeholders) for replacing dynamic data.
2. **Data source:** This represents file containing data to replace the merge fields in template Word document.
3. **Final merged document:** This resultant document is a combination of the template Word document and the data from data source.

**Tips:** 1. You can use conditional fields ([IF](#), [Formula](#)) combined with merge fields, when you require intelligent decisions in addition to simple mail merge (replace merge fields with result text). To use conditional fields, execute mail merge and then update fields in the Word document using **updateDocumentFields** API.

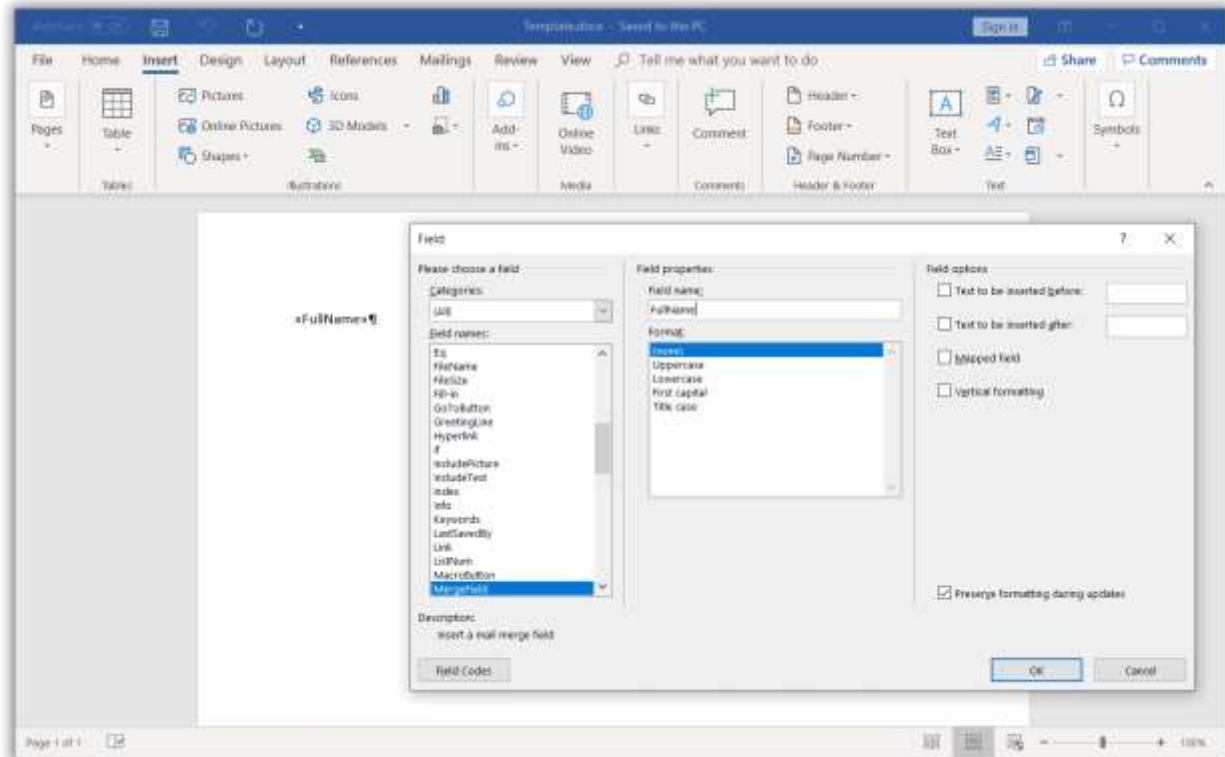


**Tips:** 2. You can replace the fields ([IF](#), [Formula](#)) combined with merge fields, with its most recent result and **generates the plain Word document** by unlinking the fields. Refer to this [link](#) for more information.

### Create Word document template

You can create a template document with merge fields by using any Word editor application, like Microsoft Word. By using Word editor application, you can take the advantage of the visual interface to design unique layout, formatting, and more for your Word document template interactively.

The following screenshot shows how to insert a merge field in the Word document by **using the Microsoft Word**.



You need to add a prefix ("Image:") to the merge field name for merging an image in the place of a merge field.

**For example:** The merge field name should be like "Image:Photo" (<<Image:MergeFieldName>>)

You can **create Word document template programmatically** by adding merge fields to the Word document using Essential DocIO.

The following code example shows how to create a merge field in the Word document.

### JAVA

```
//Creates an instance of a WordDocument.
WordDocument document = new WordDocument();
//Adds a section and a paragraph in the document.
document.ensureMinimal();
//Appends merge field to the last paragraph.
document.getLastParagraph().appendField("FullName",
FieldType.FieldMergeField);
//Saves the Word document.
```



```
document.save("Template.docx", FormatType.Docx);  
//Closes the Word document.  
document.close();
```

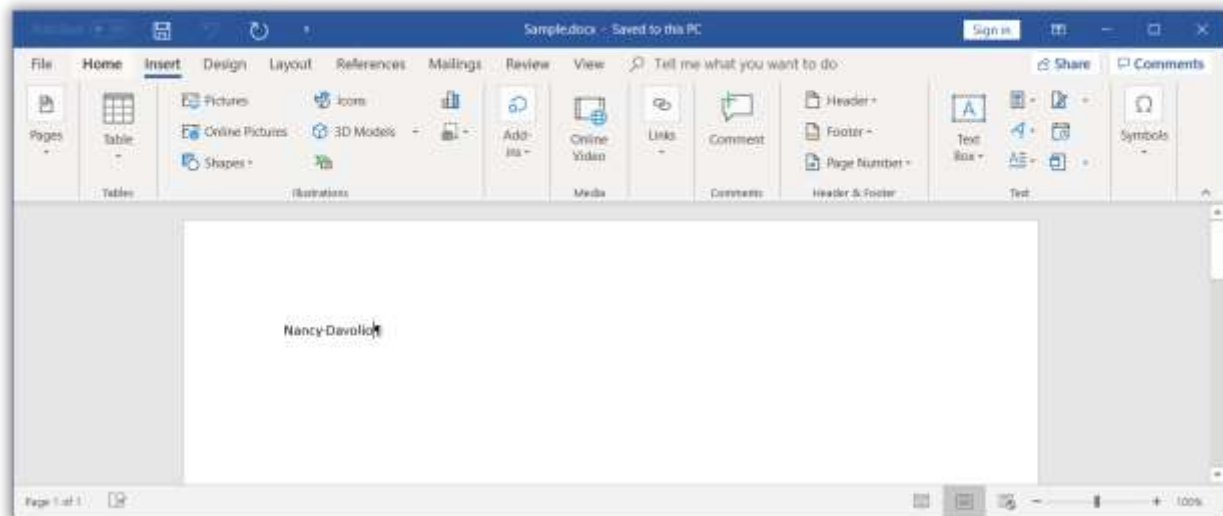
### Execute mail merge

The following code example shows how to perform mail merge in above Word document template using string arrays as data source.

### JAVA

```
//Opens the template document.  
FileInputStream fileStreamPath = new FileInputStream("Template.docx");  
WordDocument document = new WordDocument(fileStreamPath, FormatType.Docx);  
String[] fieldNames = new String[] { "FullName" };  
String[] fieldValues = new String[] { "Nancy Davolio" };  
//Performs the mail merge.  
document.getMailMerge().execute(fieldNames, fieldValues);  
//Saves the Word document.  
document.save("Output.docx", FormatType.Docx);  
//Closes the Word document.  
document.close();
```

By executing the previous code example, it generates the resultant Word document as follows.



### Simple Mail merge

The **MailMerge** class provides various overloads for the **Execute** method to perform Mail merge from various data sources. For further information, click [here](#).

### Performing Mail merge for a group

You can perform Mail merge and append multiple records from data source within a specified region to a template document. For further information, click [here](#).

### Performing Nested Mail merge for group

You can perform nested Mail merge with relational or hierarchical data source and independent data tables in a template document. For further information, click [here](#).



### Performing Mail merge with business objects

You can perform Mail merge with business objects in a template document. For further information, click [here](#).

### Performing Nested Mail merge with relational data objects

Essential DocIO supports performing nested Mail merge with implicit relational data objects without any explicit relational commands by using the `ExecuteNestedGroup` overload method. For further information, click [here](#).

### Event support for mail merge

The `MailMerge` class provides event support to customize the document contents and merging image data during the Mail merge process. The following events are supported by Essential DocIO in Mail merge process:

- `MergeField`: Occurs when a **Mail merge field** except image Mail merge field is encountered.
- `MergeImageField`: Occurs when an **image Mail merge field** is encountered.
- `BeforeClearGroupField`: Occurs when an **unmerged group field** is encountered.

#### *MergeField event*

You can customize the merging text during Mail merge process by using the `MergeField` event. For further information, click [here](#).

#### *MergeImageField event*

You can customize the merging image during Mail merge process by using the `MergeImageField` event. For further information, click [here](#).

#### *BeforeClearGroupField event*

You can get the unmerged groups during Mail merge process by using the `BeforeClearGroupField` event. For further information, click [here](#).

### Mail merge options

The `MailMerge` class allows you to customize the Mail merge process with the following options:

#### *Field mapping*

You can automatically map the merge field names with data source column names during Mail merge process. For further information, click [here](#).

#### *Retrieving the merge field names*

You can retrieve the merge field names and also merge field group names in the Word document. For further information, click [here](#).

#### *Removing empty paragraphs*

You can remove the empty paragraphs when the paragraph has a merge field item without any data during Mail merge process. For further information, click [here](#).

#### *Removing empty merge fields*

You can remove or keep the unmerged merge fields in the output document based on the `ClearFields` property on each mail merge execution. For further information, click [here](#).

#### *Restart numbering in lists*

You can restart the list numbering in a Word document during Mail merge. For further information, click [here](#).



## Working with Watermark

Watermarks are text or pictures that appear behind the document text. You can access the watermark in the document by using the `Watermark` property of `WordDocument` class.

There are two types of watermarks: Text and Picture.

### Text watermark

You can add or modify text watermark in the Word document. The `TextWatermark` class represents the text watermark in the Word document.

The following code example shows how to add a text watermark to the Word document.

#### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a section and a paragraph in the document.
document.ensureMinimal();
IWParagraph paragraph = document.getLastParagraph();
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.");
//Create a new text watermark.
TextWatermark textWatermark = new TextWatermark("TextWatermark", "", 250,
100);
//Set the created watermark to the document.
document.setWatermark(textWatermark);
//Set the text watermark font size.
textWatermark.setSize(72);
//Set the text watermark layout to horizontal.
textWatermark.setLayout(WatermarkLayout.Horizontal);
textWatermark.setSemitransparent(false);
//Set the text watermark text color.
textWatermark.setColor(ColorSupport.getBlack());
//Save the Word document.
document.save("Result_watermark1.docx", FormatType.Docx);
//Closes the document.
document.close();
```

### Picture Watermark

You can add or modify the picture watermark in the Word document. The `PictureWatermark` class represents the picture watermark in the Word document.

The following code example shows how to add a picture watermark to the Word document.

#### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add a section and a paragraph in the document.
document.ensureMinimal();
IWParagraph paragraph = document.getLastParagraph();
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.");
//Create a new picture watermark.
```



```
PictureWatermark picWatermark = new PictureWatermark();
//Set the scaling to picture.
picWatermark.setScaling(120f);
picWatermark.setWashout(true);
//Set the picture watermark to document.
document.setWatermark(picWatermark);
//Set the image to the picture watermark.
Path path = Paths.get("David.png");
byte[] data = Files.readAllBytes(path);
picWatermark.loadPicture(data);
//Save and close the document.
document.save("PictureWatermark.docx", FormatType.Docx);
document.close();
```

## Working with Comments

A comment is a note or annotation that an author or reviewer can add to a document. DocIO represents comment with `WComment` instance.

**Note:** The comment start and end ranges and dates can be preserved only on processing an existing document that already contains these information for each comment.

### Adding a Comment

You can add a new comment to the Word document by using `appendComment` method of `WParagraph` class.

The following code shows how to add a new comment to the document:

#### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds a section and a paragraph in the document.
document.ensureMinimal();
IWParagraph paragraph = document.getLastParagraph();
//Appends text to the paragraph.
paragraph.appendText("AdventureWorks Cycles, the fictitious company on which
the AdventureWorks sample databases are based, is a large, multinational
manufacturing company.");
//Adds comment to a paragraph.
WComment comment = paragraph.appendComment("comment test");
//Specifies the author of the comment.
comment.getFormat().setUser("Peter");
//Specifies the initial of the author.
comment.getFormat().setUserInitials("St");
//Set the date and time for comment.
comment.getFormat().setDateTime(LocalDateTime.now());
//Saves the Word document.
document.save("Comment.docx", FormatType.Docx);
//Closes the document.
document.close();
```

### Modifying a Comment

The following code illustrates how to modify the text of an existing comment in the Word document:

#### JAVA



```
//Opens the template document.
WordDocument document = new WordDocument("Comment.docx", FormatType.Docx);
//Iterates the comments in the Word document.
for (Object comments : document.getComments())
{
    WComment comment = (WComment)comments;
    //Modifies the last paragraph text of an existing comment when it is added
    by "Peter".
    if ((comment.getFormat().getUser()).equals("Peter"))
        comment.getTextBody().getLastParagraph().setText("Modified Comment
        Content");
}
//Saves the Word document.
document.save("ModifiedComment.docx", FormatType.Docx);
//Closes the document.
document.close();
```

### Removing Comments

You can either remove all the comments or a particular comment from the Word document.

The following code shows how to remove all the comments in Word document.

#### **JAVA**

```
//Opens the template document.
WordDocument document = new WordDocument("Comment.docx", FormatType.Docx);
//Removes all the comments in a Word document.
document.getComments().clear();
//Saves the Word document.
document.save("Result.docx", FormatType.Docx);
//Closes the document
document.close();
```

The following code shows how to remove a particular comment from Word document.

#### **JAVA**

```
//Opens the template document.
WordDocument document = new WordDocument("Comments.docx", FormatType.Docx);
//Removes second comments from a document.
document.getComments().removeAt(1);
//Saves the Word document.
document.save("Result.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Accessing parent comment

You can access the parent comment of a particular comment (reply) in a Word document using `getAncestor()` API. The ancestor for parent comment returns `null` as default.

The following code examples show how to access the parent comment of a particular comment in a Word document.

#### **JAVA**



```
//Create an empty WordDocument instance.
WordDocument document = new WordDocument();
//open an existing Word document using the Open method of WordDocument
class.
document.open("Comments.docx");
//Get the Ancestor comment.
WComment ancestorComment = document.getComments().get(1).getAncestor();
//Save and Close the Word document.
document.save("Result.docx", FormatType.Docx);
document.close();
```

## Working with Form Fields

You can create template document with form fields such as Text, Checkbox and Drop-Down. You can also open an existing template document and fill the form fields with the specified data.

The following are the types of form field in the Word document

- Checkbox – represented by an instance of WCheckBox
- Drop-down – represented by an instance of WDropDownFormField
- Text input – represented by an instance of WTextFormField

### Check Box

You can add new Checkbox form field to a Word document by using `appendCheckBox` method of `WParagraph` class.

The following code illustrates how to add new checkbox form field.

### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds new section to the document.
IWSection section = document.addSection();
//Adds new paragraph to the section.
WParagraph paragraph = (WParagraph)section.addParagraph();
paragraph.appendText("Gender\t");
//Appends new Checkbox.
WCheckBox checkbox = paragraph.appendCheckBox();
checkbox.setChecked(false);
//Sets Checkbox size.
checkbox.setCheckBoxSize(10);
checkbox.setCalculateOnExit(true);
//Sets help text.
checkbox.setHelp("Help text");
paragraph.appendText("Male\t");
checkbox = paragraph.appendCheckBox();
checkbox.setChecked(false);
checkbox.setCheckBoxSize(10);
checkbox.setCalculateOnExit(true);
paragraph.appendText("Female");
//Saves the Word document.
document.save("Checkbox.docx", FormatType.Docx);
//Closes the document
document.close();
```



You can modify the checkbox properties such as checked state, size, help text in a Word document. The following code illustrates how to modify the checkbox form field properties.

#### JAVA

```
//Loads the template document.
WordDocument document = new WordDocument("Checkbox.docx");
//Iterates through paragraph items.
for (Object item_tempObj : document.getLastParagraph().getChildEntities())
{
    ParagraphItem item = (ParagraphItem) item_tempObj;
    if (item instanceof WCheckBox)
    {
        WCheckBox checkbox = (WCheckBox) item;
        //Modifies check box properties.
        if (checkbox.getChecked())
            checkbox.setChecked(false);
        checkbox.setSizeType(CheckBoxSizeType.Exactly);
    }
}
//Saves the Word document.
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

#### Drop-Down

You can add new Dropdown form field to a Word document by using `appendDropDownFormField` method of `WParagraph` class.

The following code illustrates how to add a new dropdown field.

#### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds new section to the document.
IWSection section = document.addSection();
//Adds new paragraph to the section.
WParagraph paragraph = (WParagraph)section.addParagraph();
paragraph.appendText("Educational Qualification\t");
//Appends Dropdown field.
WDropDownFormField dropDownField = paragraph.appendDropDownFormField();
//Adds items to the Dropdown items collection.
dropDownField.getDropDownItems().add("Higher");
dropDownField.getDropDownItems().add("Vocational");
dropDownField.getDropDownItems().add("Universal");
dropDownField.setEnabled(true);
//Sets the item index for default value.
dropDownField.setDropDownSelectedIndex(1);
dropDownField.setCalculateOnExit(true);
//Saves the Word document.
document.save("Dropdown.docx", FormatType.Docx);
//Closes the document.
document.close();
```



You can add or modify list of items of a Dropdown form field in a Word document. The following code illustrates how to modify the dropdown list of a Dropdown form field.

#### JAVA

```
//Loads the template document.
WordDocument document = new WordDocument("Dropdown.docx");
//Iterates through paragraph items.
for (Object item_tempObj : document.getLastParagraph().getChildEntities())
{
    ParagraphItem item = (ParagraphItem) item_tempObj;
    if (item instanceof WDropDownFormField)
    {
        WDropDownFormField dropdown = (WDropDownFormField)item;
        //Modifies the dropdown items.
        dropdown.getDropDownItems().remove(1);
        dropdown.setDropDownSelectedIndex(0);
        dropdown.getCharacterFormat().setFontName("Arial");
    }
}
//Saves the Word document.
document.save("Sample.docx", FormatType.Docx);
//Closes the document.
document.close();
```

#### Text Form field

You can add new text form field to a Word document by using `appendTextFormField` method of `WParagraph` class.

The following code illustrates how to add new text form field.

#### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds new section to the document.
IWSection section = document.addSection();
//Adds new paragraph to the section.
WParagraph paragraph = (WParagraph)section.addParagraph();
paragraph.appendText("General Information");
section.addParagraph();
paragraph = (WParagraph)section.addParagraph();
IWTextRange text = paragraph.appendText("Name\t");
text.getCharacterFormat().setBold(true);
//Appends Text form field.
WTextFormField textField = paragraph.appendTextFormField(null);
//Sets type of Text form field.
textField.setType(TextFormFieldType.RegularText);
textField.getCharacterFormat().setFontName("Calibri");
textField.setCalculateOnExit(true);
section.addParagraph();
paragraph = (WParagraph)section.addParagraph();
text = paragraph.appendText("Date of Birth\t");
text.getCharacterFormat().setBold(true);
//Appends Text form field.
```



```

textField = paragraph.appendTextFormField("Date field",
DateTimeSupport.toString(LocalDateTime.now(), "MM/DD/YY"));
textField.setStringFormat("MM/DD/YY");
textField.setType(TextFormFieldType.DateText);
textField.setCalculateOnExit(true);
//Saves the Word document.
document.save("TextForm.docx", FormatType.Docx);
//Closes the document
document.close();

```

You can add or modify text form field properties such as default text, type in a Word document. The following code illustrates how to modify the text form field

### JAVA

```

//Loads the template document.
WordDocument document = new WordDocument("TextForm.docx");
//Iterates through section.
for (Object section_tempObj : document.getSections())
{
WSection section = (WSection) section_tempObj;
//Iterates through section child elements.
for (Object textBody_tempObj : section.getChildEntities())
{
WTextBody textBody = (WTextBody) textBody_tempObj;
//Iterates through form fields.
for (Object formField_tempObj : textBody.getFormFields())
{
WFormField formField = (WFormField) formField_tempObj;
switch (formField.getFormFieldType().toString())
{
case "TextInput":
WTextFormField textField = (WTextFormField) formField;
if (textField.getType().getEnumValue() ==
TextFormFieldType.DateText.getEnumValue())
{
//Modifies the text form field.
textField.setType(TextFormFieldType.RegularText);
textField.setStringFormat("");
textField.setDefaultText("Default text");
textField.setText("Default text");
textField.setCalculateOnExit(false);
}
break;
}
}
}
}
//Saves the Word document.
document.save("Sample.docx", FormatType.Docx);
//Closes the document.
document.close();

```



## Working with Security

You can encrypt a Word document with password to restrict unauthorized access. You can also control the types of changes you make to this document.

### Encrypting with password

The following code example shows how to encrypt the Word document with password.

#### JAVA

```
//Open an input Word document.
WordDocument document = new WordDocument("Template.docx");
//Encrypt the Word document with a password.
document.encryptDocument("password");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Opening the encrypted Word document

The following code example shows how to open the encrypted Word document.

#### JAVA

```
//Open an input Word document.
WordDocument document = new WordDocument("Template.docx", "password");
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Remove encryption

You can open the encrypted Word document and remove the encryption from the document. The following code example shows how to remove the encryption from encrypted Word document.

#### JAVA

```
//Open an encrypted Word document.
WordDocument document = new WordDocument ("Template.docx", "password");
//Remove encryption in Word document.
document.removeEncryption();
//Save and close the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Protecting Word document from editing

You can restrict a Word document from editing either by providing a password or without password.

The following are the types of protection:

1. **AllowOnlyComments**: You can add/modify only the comments in the Word document.
2. **AllowOnlyFormFields**: You can modify the form field values in the Word document.
3. **AllowOnlyRevisions**: You can accept or reject the revisions in the Word document.
4. **AllowOnlyReading**: You can only view the content in the Word document.
5. **NoProtection**: You can access/edit the Word document contents as normally.



The following code example shows how to restrict editing to modify only form fields in a Word document.

#### JAVA

```
//Open a Word document.
WordDocument document = new WordDocument("Template.docx");
//Set the protection with password and it allows only to modify the form
fields type.
document.protect(ProtectionType.AllowOnlyFormFields, "password");
//Save the Word document.
document.save("Protection.docx", FormatType.Docx);
document.close();
```

## Working with Content Controls

### What is Content Control?

Content controls are individual controls that you can add and customize to use in templates, forms, and documents. For example, many online forms are designed with a drop-down list control that provides a restricted set of choices.

**Note:** You can use content controls only in documents that are saved in the Open XML Format.

Content controls can be categorized based on its occurrence in a document as follows,

- **InlineContentControl:** Among inline content inside, as a child of a paragraph.
- **BlockContentControl:** Among paragraphs and tables, as a child of a Body, HeaderFooter, Comment, Footnote, or a Shape node.

### Block Content Control

You can add content control to a text body of the Word document using block content control. You can add text, tables, pictures, or other items into the block content control. Refer to the following code.

#### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds new section to the document.
IWSection section = document.addSection();
WTextBody textBody = section.getBody();
//Adds block content control into Word document.
BlockContentControl blockContentControl =
(BlockContentControl)textBody.addBlockContentControl(ContentControlType.Rich
Text);
//Adds new paragraph in the block content control.
WParagraph paragraph =
(WParagraph)blockContentControl.getTextBody().addParagraph();
//Adds new text to the paragraph.
paragraph.appendText("Block content control");
//Adds new table to the block content control.
WTable table = (WTable)blockContentControl.getTextBody().addTable();
//Specifies the total number of rows and columns.
table.resetCells(2,3);
paragraph = (WParagraph)blockContentControl.getTextBody().addParagraph();
//Adds image to the paragraph.
```



```
paragraph.appendPicture(new FileInputStream("Image.png"));  
//Saves and closes the Word document instance.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

### Inline Content Control

You can add content control as a child to a paragraph using the inline content control. You can add text, pictures, fields or other paragraph items into the inline content control. Refer to the following code.

### JAVA

```
//Creates a new Word document.  
WordDocument document = new WordDocument();  
//Adds one section and one paragraph to the document.  
document.ensureMinimal();  
//Gets the last paragraph.  
WParagraph paragraph = document.getLastParagraph();  
//Adds text to the paragraph.  
paragraph.appendText("A new text is added to the paragraph. ");  
//Appends inline content control to the paragraph.  
InlineContentControl inlineContentControl =  
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.  
e.RichText);  
WTextRange textRange = new WTextRange(document);  
//Adds new text to the inline content control.  
textRange.setText("Inline content control ");  
inlineContentControl.getParagraphItems().add(textRange);  
//Saves and closes the Word document instance.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

**Note:** Currently, DocIO does not support RowContentControl and CellContentControl.

### Common properties of Content Control

You can set formatting options for the content control in the Word document. The following are the common properties of a content control.

#### Title

The title of the content control.

#### Tag

The tag value to identify the content control.

#### Appearance

This property allows you to define the appearance of the content controls. The appearance can be any one of the following:

- BoundingBox: Displays the contents of content control within a box.
- Tags: Displays the contents of content control within tags.
- Hidden: Displays the contents of content control without any box or tags.

#### Color

Defines the color of the content control.



*Temporary*

This property defines whether to remove a content control from the Word document when you edit the contents of the content control.

*Lock Contents*

Locking the contents of the content control. It restricts to modify the contents of the content control.

*Lock Content Control*

It restricts to remove or delete the content control.

*Example – Content Control Common properties*

The following code sample illustrates the content control properties usage.

**JAVA**

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document.
document.ensureMinimal();
//Gets the last paragraph.
WParagraph paragraph = document.getLastParagraph();
//Adds text to the paragraph.
paragraph.appendText("A new text is added to the paragraph. ");
//Appends rich text content control to the paragraph.
IInlineContentControl contentControl =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.
e.RichText);
WTextRange textRange = new WTextRange(document);
textRange.setText("Rich text content control.");
//Adds new text to the rich text content control.
contentControl.getParagraphItems().add(textRange);
//Sets tag appearance for the content control.
contentControl.getContentControlProperties().setAppearance(ContentControlApp
pearance.Tags);
//Sets a tag property to identify the content control.
contentControl.getContentControlProperties().setTag("Rich Text");
//Sets a title for the content control.
contentControl.getContentControlProperties().setTitle("Text");
//Sets the color for the content control.
contentControl.getContentControlProperties().setColor(ColorSupport.getMagenta());
//Gets the type of content control.
ContentControlType controlType =
contentControl.getContentControlProperties().getType();
//Enables content control lock.
contentControl.getContentControlProperties().setLockContentControl(true);
//Protects the contents of content control.
contentControl.getContentControlProperties().setLockContents(true);
//Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

*Why Content Control?*

The content controls have the following three major use cases:



- Protection
- Form Filling
- Data Binding with Content Controls (XML Mapping)

### Protection

Content controls provides options to prevent users from editing or deleting parts of a Word document contents. This is useful if you have information in a Word document or template that you should be able to read but not edit, or if you want to be able to edit content controls but not delete them.

To protect contents inside a content control, you can use properties of the content control to prevent editing or deleting the content control:

- The **LockContents** property prevents from editing the contents of the content control.
- The **LockContentControl** property prevents from deleting the content control.

The following code sample shows how to protect the content control and its contents.

### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document.
document.ensureMinimal();
//Gets the last paragraph.
WParagraph paragraph = document.getLastParagraph();
//Adds text to the paragraph.
paragraph.appendText("A new text is added to the paragraph. ");
//Appends rich text content control to the paragraph.
IInlineContentControl contentControl =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.
RichText);
WTextRange textRange = new WTextRange(document);
textRange.setText("Rich text content control.");
//Adds new text to the rich text content control.
contentControl.getParagraphItems().add(textRange);
//Sets tag appearance for the content control.
contentControl.getContentControlProperties().setAppearance(ContentControlApp
earance.Tags);
//Sets a tag property to identify the content control.
contentControl.getContentControlProperties().setTag("Rich Text Protected");
//Sets a title for the content control.
contentControl.getContentControlProperties().setTitle("Text Protected");
//Enables content control lock.
contentControl.getContentControlProperties().setLockContentControl(true);
//Protects the contents of content control.
contentControl.getContentControlProperties().setLockContents(true);
//Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Form Filling

Another major use case is to create the forms. You can design your own forms for various stages using the text box, check box, and list box. Refer to the following code example.



Form creation:

### JAVA

```
//Creates a new Word document .
WordDocument document = new WordDocument();
//Adding a new section to the document.
IWSection section = document.addSection();
//Adding a new paragraph to the section.
//Document formatting.
//Sets background color for document.
IWParagraph paragraph = section.addParagraph();
document.getBackground().getGradient().setColor1(ColorSupport.fromArgb(232,232,232));
document.getBackground().getGradient().setColor2(ColorSupport.fromArgb(255,255,255));
document.getBackground().setType(BackgroundType.Gradient);
document.getBackground().getGradient().setShadingStyle(GradientShadingStyle.Horizontal);
document.getBackground().getGradient().setShadingVariant(GradientShadingVariant.ShadingDown);
//Sets page size for document.
section.getPageSetup().getMargins().setAll(30f);
section.getPageSetup().setPageSize(new SizeFSupport(600,600f));
//Title Section.
//Adds a new table to the section.
IWTable table = section.getBody().addTable();
table.resetCells(1,2);
//Gets the table first row.
WTableRow row = table.getRows().get(0);
row.setHeight(25f);
//Adds a new paragraph to the cell.
IWParagraph cellPara = row.getCells().get(0).addParagraph();
//Appends a new picture.
IWPicture pic = cellPara.appendPicture(new FileInputStream("Image.jpg"));
pic.setHeight((float)80);
pic.setWidth((float)180);
//Adds a new paragraph to the next cell.
cellPara = row.getCells().get(1).addParagraph();
row.getCells().get(1).getCellFormat().setVerticalAlignment(VerticalAlignment.Middle);
row.getCells().get(1).getCellFormat().setBackColor(ColorSupport.fromArgb(173,215,255));
//Appends the text.
IWTextRange txt = cellPara.appendText("Job Application Form");
cellPara.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Center);
//Sets the formats.
txt.getCharacterFormat().setBold(true);
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(18f);
//Sets the width and border type.
row.getCells().get(0).setWidth((float)200);
row.getCells().get(1).setWidth((float)300);
row.getCells().get(1).getCellFormat().getBorders().setBorderType(BorderStyle.Hairline);
//Adds a new paragraph.
```



```

section.addParagraph();
//General Information.
//Adds a new table.
table=section.getBody().addTable();
table.resetCells(2,1);
row = table.getRows().get(0);
row.setHeight((float)20);
row.getCells().get(0).setWidth((float)500);
//Adds a new paragraph.
cellPara = row.getCells().get(0).addParagraph();
//Sets a border type, color, background, and vertical alignment for cell.
row.getCells().get(0).getCellFormat().getBorders().setBorderType(BorderStyle
.Thick);
row.getCells().get(0).getCellFormat().getBorders().setColor(ColorSupport.from
mArgb(155,205,255));
row.getCells().get(0).getCellFormat().setBackColor(ColorSupport.fromArgb(198
,227,255));
row.getCells().get(0).getCellFormat().setVerticalAlignment(VerticalAlignment
.Middle);
txt=cellPara.appendText(" General Information");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setBold(true);
txt.getCharacterFormat().setFontSize(11f);
row = table.getRows().get(1);
cellPara=row.getCells().get(0).addParagraph();
//Sets a width, border type, color and background for cell
row.getCells().get(0).setWidth((float)500);
row.getCells().get(0).getCellFormat().getBorders().setBorderType(BorderStyle
.Hairline);
row.getCells().get(0).getCellFormat().getBorders().setColor(ColorSupport.from
mArgb(155,205,255));
row.getCells().get(0).getCellFormat().setBackColor(ColorSupport.fromArgb(222
,239,255));
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n Full Name:\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a new inline content control to enter the value.
InlineContentControl txtField =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.Text);
txtField.getContentControlProperties().setTitle("Text");
//Sets formatting options for text present inside a content control.
txtField.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue
());
txtField.getBreakCharacterFormat().setFontName("Arial");
txtField.getBreakCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n\n Birth Date:\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a new inline content control to enter the value.
txtField =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.Date);
txtField.getContentControlProperties().setTitle("Date");
//Sets the date display format.

```



```

txtField.getContentControlProperties().setDateDisplayFormat("M/d/yyyy");
//Sets formatting options for text present inside a content control.
txtField.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue());
txtField.getBreakCharacterFormat().setFontName("Arial");
txtField.getBreakCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt=cellPara.appendText("\n\n Address:\t\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a new inline content control to enter the value.
txtField =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.Text);
txtField.getContentControlProperties().setTitle("Text");
//Sets multiline property to true to get the multiple line input of Address.
txtField.getContentControlProperties().setMultiline(true);
//Sets formatting options for text present inside a content control.
txtField.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue());
txtField.getBreakCharacterFormat().setFontName("Arial");
txtField.getBreakCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n\n Phone:\t\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a new inline content control to enter the value.
txtField =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.Text);
txtField.getContentControlProperties().setTitle("Text");
//Sets formatting options for text present inside a content control.
txtField.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue());
txtField.getBreakCharacterFormat().setFontName("Arial");
txtField.getBreakCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n\n Email:\t\t\t\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a new inline content control to enter the value.
txtField =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.Text);
txtField.getContentControlProperties().setTitle("Text");
//Sets formatting options for text present inside a content control.
txtField.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue());
txtField.getBreakCharacterFormat().setFontName("Arial");
txtField.getBreakCharacterFormat().setFontSize(11f);
cellPara.appendText("\n");
section.addParagraph();
//Educational Qualification.
//Adds a new table to the section.
table = section.getBody().addTable();
table.resetCells(2,1);
row = table.getRows().get(0);

```



```

row.setHeight((float)20);
//Sets width, border type, color, background and vertical alignment for
cell.
row.getCells().get(0).setWidth((float)500);
row.getCells().get(0).getCellFormat().getBorders().setBorderType(BorderStyle
.Thick);
row.getCells().get(0).getCellFormat().getBorders().setColor(ColorSupport.fro
mArgb(155,205,255));
row.getCells().get(0).getCellFormat().setBackColor(ColorSupport.fromArgb(198
,227,255));
row.getCells().get(0).getCellFormat().setVerticalAlignment(VerticalAlignment
.Middle);
cellPara=row.getCells().get(0).addParagraph();
//Appends a text to paragraph of cell.
txt = cellPara.appendText(" Educational Qualification");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setBold(true);
txt.getCharacterFormat().setFontSize(11f);
row = table.getRows().get(1);
//Sets width, border type, color, and background for cell.
row.getCells().get(0).setWidth((float)500);
row.getCells().get(0).getCellFormat().getBorders().setBorderType(BorderStyle
.Hairline);
row.getCells().get(0).getCellFormat().getBorders().setColor(ColorSupport.fro
mArgb(155,205,255));
row.getCells().get(0).getCellFormat().setBackColor(ColorSupport.fromArgb(222
,239,255));
cellPara = row.getCells().get(0).addParagraph();
//Appends a text to paragraph of cell.
txt=cellPara.appendText("\n Type:\t\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a new inline content control to enter the value.
InlineContentControl dropdown =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.DropDownList);
WTextRange textRange = new WTextRange(document);
textRange.setText("Choose an item from drop down list");
dropdown.getParagraphItems().add(textRange);
//Creates an item for dropdown list.
ContentControlListItem item = new ContentControlListItem();
//Sets the text to be displayed as list item.
item.setDisplayText("Higher");
//Sets the value to the list item.
item.setValue("1");
//Adds item to the dropdown list.
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
item = new ContentControlListItem();
item.setDisplayText("Vocational");
item.setValue("2");
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
item = new ContentControlListItem();
item.setDisplayText("Universal");
item.setValue("3");

```



```

dropdown.getContentControlProperties().getContentControlListItems().add(item
);
dropdown.getContentControlProperties().setTitle("Drop-Down");
//Sets formatting options for text present inside a content control.
dropdown.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue
());
dropdown.getBreakCharacterFormat().setFontName("Arial");
dropdown.getBreakCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n\n Institution:\t\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a new inline content control to enter the value.
txtField =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.Text);
//Sets formatting options for text present inside a content control.
txtField.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue
());
txtField.getBreakCharacterFormat().setFontName("Arial");
txtField.getBreakCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n\n Programming Languages:");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n\n\t C#:\t\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(9f);
//Appends a new inline content control to enter the value.
dropdown =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.DropDownList);
textRange = new WTextRange(document);
textRange.setText("Choose an item from drop down list");
dropdown.getParagraphItems().add(textRange);
//Creates an item for dropdown list.
item = new ContentControlListItem();
item.setDisplayText("Perfect");
//Sets the value to the list item.
item.setValue("1");
//Adds item to the dropdown list.
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
item = new ContentControlListItem();
item.setDisplayText("Good");
item.setValue("2");
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
item = new ContentControlListItem();
item.setDisplayText("Excellent");
item.setValue("3");
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
//Sets formatting options for text present inside a content control.
dropdown.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue
());

```



```

dropdown.getBreakCharacterFormat().setFontName("Arial");
dropdown.getBreakCharacterFormat().setFontSize(11f);
//Appends a text to paragraph of cell.
txt = cellPara.appendText("\n\n\t VB:\t\t\t\t");
txt.getCharacterFormat().setFontName("Arial");
txt.getCharacterFormat().setFontSize(9f);
//Appends a new inline content control to enter the value.
dropdown =
(InlineContentControl)cellPara.appendInlineContentControl(ContentControlType
.DropDownList);
textRange = new WTextRange(document);
textRange.setText("Choose an item from drop down list");
dropdown.getParagraphItems().add(textRange);
//Creates an item for dropdown list.
item = new ContentControlListItem();
//Sets the text to be displayed as list item.
item.setDisplayText("Perfect");
//Sets the value to the list item.
item.setValue("1");
//Adds item to the dropdown list.
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
item = new ContentControlListItem();
item.setDisplayText("Good");
item.setValue("2");
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
item = new ContentControlListItem();
item.setDisplayText("Excellent");
item.setValue("3");
dropdown.getContentControlProperties().getContentControlListItems().add(item
);
dropdown.getContentControlProperties().setTitle("Drop-Down");
//Sets formatting options for text present inside a content control.
dropdown.getBreakCharacterFormat().setTextColor(ColorSupport.getMidnightBlue
());
dropdown.getBreakCharacterFormat().setFontName("Arial");
dropdown.getBreakCharacterFormat().setFontSize(11f);
//Saves and closes the Word document instance.
document.save("Form_Template.docx");
document.close();

```

You can also fill the forms using the DocIO. Refer to the following code example.

Form filling:

#### **JAVA**

```

//Open the created form document.
WordDocument document1 = new WordDocument("Form_Template.docx");
IWSection sec = document1.getLastSection();
InlineContentControl inlineCC;
InlineContentControl dropDownCC;
WTable table1 = (WTable)sec.getTables().get(1);
WTableRow row1 = table1.getRows().get(1);
//General Information.
//Fill the name

```



```

WParagraph cellPara1 =
(WParagraph)row1.getCells().get(0).getChildEntities().get(1);
inlineCC = (InlineContentControl)cellPara1.getChildEntities().getLastItem();
WTextRange text = new WTextRange(document1);
text.applyCharacterFormat(inlineCC.getBreakCharacterFormat());
text.setText("Steve Jobs");
inlineCC.getParagraphItems().add(text);
//Fill the date of birth.
cellPara1 = (WParagraph)row1.getCells().get(0).getChildEntities().get(3);
inlineCC = (InlineContentControl)cellPara1.getChildEntities().getLastItem();
text=new WTextRange(document1);
text.applyCharacterFormat(inlineCC.getBreakCharacterFormat());
text.setText("06/01/1994");
inlineCC.getParagraphItems().add(text);
//Fill the address.
cellPara1 = (WParagraph)row1.getCells().get(0).getChildEntities().get(5);
inlineCC=(InlineContentControl)cellPara1.getChildEntities().getLastItem();
text = new WTextRange(document1);
text.applyCharacterFormat(inlineCC.getBreakCharacterFormat());
text.setText("2501 Aerial Center Parkway.");
inlineCC.getParagraphItems().add(text);
text = new WTextRange(document1);
text.applyCharacterFormat(inlineCC.getBreakCharacterFormat());
text.setText("Morrisville, NC 27560.");
inlineCC.getParagraphItems().add(text);
text = new WTextRange(document1);
text.applyCharacterFormat(inlineCC.getBreakCharacterFormat());
text.setText("USA.");
inlineCC.getParagraphItems().add(text);
//Fill the phone no.
cellPara1 = (WParagraph)row1.getCells().get(0).getChildEntities().get(7);
inlineCC = (InlineContentControl)cellPara1.getChildEntities().getLastItem();
text = new WTextRange(document1);
text.applyCharacterFormat(inlineCC.getBreakCharacterFormat());
text.setText("+1 919.481.1974");
inlineCC.getParagraphItems().add(text);
//Fill the email id.
cellPara1 = (WParagraph)row1.getCells().get(0).getChildEntities().get(9);
inlineCC=(InlineContentControl)cellPara1.getChildEntities().getLastItem();
text = new WTextRange(document1);
text.applyCharacterFormat(inlineCC.getBreakCharacterFormat());
text.setText("steve123@email.com");
inlineCC.getParagraphItems().add(text);
//Educational Information.
table1=(WTable)sec.getTables().get(2);
row1 = table1.getRows().get(1);
//Fill the education type.
cellPara1 = (WParagraph)row1.getCells().get(0).getChildEntities().get(1);
dropDownCC =
(InlineContentControl)cellPara1.getChildEntities().getLastItem();
text = new WTextRange(document1);
text.applyCharacterFormat(dropDownCC.getBreakCharacterFormat());
text.setText(dropDownCC.getContentControlProperties().getContentControlListI
tems().get(1).getDisplayText());
dropDownCC.getParagraphItems().add(text);
//Fill the university.
cellPara1 = (WParagraph)row1.getCells().get(0).getChildEntities().get(3);

```



```

inlineCC = (InlineContentControl)cellPara1.getChildEntities().getLastItem();
text = new WTextRange(document1);
text.applyCharacterFormat(dropDownCC.getBreakCharacterFormat());
text.setText("Michigan University");
inlineCC.getParagraphItems().add(text);
//Fill the C# experience level.
cellPara1 = (WParagraph)row1.getCells().get(0).getChildEntities().get(7);
dropDownCC=(InlineContentControl)cellPara1.getChildEntities().getLastItem();
text = new WTextRange(document1);
text.applyCharacterFormat(dropDownCC.getBreakCharacterFormat());
text.setText(dropDownCC.getContentControlProperties().getContentControlListI
tems().get(2).getDisplayText());
dropDownCC.getParagraphItems().add(text);
//Fill the VB experience level.
cellPara1=(WParagraph)row1.getCells().get(0).getChildEntities().get(9);
dropDownCC=(InlineContentControl)cellPara1.getChildEntities().getLastItem();
text = new WTextRange(document1);
text.applyCharacterFormat(dropDownCC.getBreakCharacterFormat());
text.setText(dropDownCC.getContentControlProperties().getContentControlListI
tems().get(1).getDisplayText());
dropDownCC.getParagraphItems().add(text);
//Saves and closes the Word document instance.
document1.save("Form_Filled.docx");
document1.close();

```

## Types of Content Controls

The following types of content controls can be created by using the Essential DocIO.

- Rich Text
- Plain Text
- Check Box
- Drop-Down List and Combo Box
- Picture

### Rich Text

A rich text content control contains text or other items, such as tables, pictures, or other content controls. The following code illustrates how to add new rich text content control.

### JAVA

```

//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document.
document.ensureMinimal();
//Gets the last paragraph.
WParagraph paragraph = document.getLastParagraph();
//Adds text to the paragraph.
paragraph.appendText("A new text is added to the paragraph. ");
InlineContentControl richTextControl =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlTyp
e.RichText);
WTextRange textRange = new WTextRange(document);
textRange.setText("Rich text content control.");
//Adds new text to the rich text content control.

```



```
richTextControl.getParagraphItems().add(textRange);
WPicture picture = new WPicture(document);
picture.loadImage(new FileInputStream("Image.png"));
picture.setHeight((float)100);
picture.setWidth((float)100);
//Adds new picture to the rich text content control.
richTextControl.getParagraphItems().add(picture);
//Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Plain Text

A plain text content control contains text and cannot contain other items, such as tables, pictures, or other content controls. Refer to the following code to add plain text content control.

#### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document.
document.ensureMinimal();
//Gets the last paragraph.
WParagraph paragraph = document.getLastParagraph();
//Adds text to the paragraph.
paragraph.appendText("A new text is added to the paragraph. ");
//Appends plain text content control to the paragraph.
InlineContentControl plainTextControl =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.
Text);
WTextRange textRange = new WTextRange(document);
textRange.setText("Plain text content control.");
//Adds new text to the plain text content control.
plainTextControl.getParagraphItems().add(textRange);
//Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Check Box

A check box content control provides a UI that represents a binary state: checked or unchecked. Default state for check box is unchecked. Refer to the following code to add check box content control.

#### JAVA

```
//Creates a new Word document .
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document.
document.ensureMinimal();
//Gets the last paragraph.
WParagraph paragraph = document.getLastParagraph();
//Adds text to the paragraph.
paragraph.appendText("A new text is added to the paragraph. ");
//Appends checkbox content control to the paragraph.
InlineContentControl checkBox =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.
CheckBox);
```



```
checkboxBox.getContentControlProperties().setIsChecked(true);
//Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Drop-Down List and Combo Box

A drop-down list content control and combo box content control displays a list of items you can select. Unlike a drop-down list, the combo box allows to add your own items. Refer to the following code to add drop-down list and combo box content controls.

### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds new section to the document.
IWSection section = document.addSection();
//Adds new paragraph to the section.
WParagraph paragraph = (WParagraph)section.addParagraph();
InlineContentControl dropdown =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.
DropDownList);
WTextRange textRange = new WTextRange(document);
//Sets default option to display.
textRange.setText("Choose an item from drop down list");
dropdown.getParagraphItems().add(textRange);
//Creates an item for dropdown list.
ContentControlListItem item = new ContentControlListItem();
//Sets the text to be displayed as list item.
item.setDisplayText("ASP.NET MVC");
//Sets the value to the list item.
item.setValue("1");
//Adds item to the dropdown list.
dropdown.getContentControlProperties().getContentControlListItems().add(item);
item = new ContentControlListItem();
item.setDisplayText("Windows Forms");
item.setValue("2");
dropdown.getContentControlProperties().getContentControlListItems().add(item);
item = new ContentControlListItem();
item.setDisplayText("WPF");
item.setValue("3");
dropdown.getContentControlProperties().getContentControlListItems().add(item);
//Adds new paragraph to the section.
paragraph = (WParagraph)section.addParagraph();
//Appends combo box content control to the paragraph.
InlineContentControl comboBox =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.
ComboBox);
textRange=new WTextRange(document);
//Sets default option to display.
textRange.setText("Choose an item from combo box");
comboBox.getParagraphItems().add(textRange);
//Creates an item for combo box.
item = new ContentControlListItem();
```



```
//Sets the text to be displayed as list item.
item.setDisplayText("Word to HTML");
//Sets the value to the list item.
item.setValue("1");
comboBox.getContentControlProperties().getContentControlListItems().add(item);
item = new ContentControlListItem();
item.setDisplayText("Word to Image");
item.setValue("2");
comboBox.getContentControlProperties().getContentControlListItems().add(item);
item = new ContentControlListItem();
item.setDisplayText("Word to PDF");
item.setValue("3");
comboBox.getContentControlProperties().getContentControlListItems().add(item);
//Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Picture

A picture content control displays an image. Refer to the following code to add new picture content control.

### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document.
document.ensureMinimal();
//Gets the last paragraph.
WParagraph paragraph = document.getLastParagraph();
//Adds text to the paragraph.
paragraph.appendText("A new text is added to the paragraph. ");
//Appends picture content control to the paragraph.
InlineContentControl pictureContentControl =
(InlineContentControl)paragraph.appendInlineContentControl(ContentControlType.
Picture);
//Creates a new image instance and load image.
WPicture picture = new WPicture(document);
picture.loadImage(new FileInputStream("Image.png"));
//Adds picture to the picture content control.
pictureContentControl.getParagraphItems().add(picture);
//Saves and closes the Word document instance.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### Working with Mathematical Equation

Equations in Word document are combination of mathematical symbols or text. For example, you can create a Fourier series equation in Word document.

![Mathematical equation in Microsoft Word document](WorkingwithMathematicalEquation\_images/Mathematical Equation.png)



## Types of equation

The following different structures of equation can be created by using the Essential DocIO.



- Accent
- Bar
- Box
- Border box
- Delimiter
- Equation array
- Fraction
- Function
- Group character
- Limit
- Matrix
- N-Array
- Radical
- Phantom
- SubSuperscript
- Left SubSuperscript
- Right SubSuperscript

### Accent

You can add an accent mark to the equation. The following code example shows how to add an accent mark to the equation.

### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds an accent equation
IOfficeMathAccent mathAccent = (IOfficeMathAccent)
officeMath.getFunctions().add(MathFunctionType.Accent);
//Sets the accent character
mathAccent.setAccentCharacter("^");
//Adds the run element for accent
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
mathAccent.getEquation().getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
WTextRange textRange = (WTextRange) officeMathRunElement.getItem();
//Sets text for accent equation
textRange.setText("a");
```



```
//Applies character formatting for text range
textRange.getCharacterFormat().setBold(true);
textRange.getCharacterFormat().setItalic(true);
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Bar

You can add a bar (which adds horizontal line on top or bottom) to the equation. The following code example shows how to add a bar to the equation.

### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Add a section and a paragraph in the empty document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds a bar equation
IOfficeMathBar mathBar = (IOfficeMathBar) officeMath.getFunctions().add(0,
MathFunctionType.Bar);
//Sets the position of bar
mathBar.setBarTop(true);
//Adds the run element for bar
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
mathBar.getEquation().getFunctions().add(0, MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for bar equation
((WTextRange)officeMathRunElement.getItem()).setText("a");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Box

You can add a box to the equation. The following code example shows how to add a box to the equation.

### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds a box equation
IOfficeMathBox mathBox = (IOfficeMathBox) officeMath.getFunctions().add(0,
MathFunctionType.Box);
```



```
//Adds the run element for box
IOfficeMathRunElement officeMathRunElement =(IOfficeMathRunElement)
officeMath.getFunctions().add(0, MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for math
((WTextRange)officeMathRunElement.getItem()).setText("a+b");
//Enables the flag, to behave the box and its contents as a single operator
mathBox.setOperatorEmulator(true);
//Enables the flag, to act box as the mathematical differential
mathBox.setEnableDifferential(true);
//Adds a break in box equation
mathBox.setBreak(officeMath.getBreaks().add(0));
//Adds the run element for box
officeMathRunElement =(IOfficeMathRunElement)
mathBox.getEquation().getFunctions().add(0, MathFunctionType.RunElement);
officeMathRunElement.setItem( new WTextRange(document));
//Sets text for box equation
((WTextRange)officeMathRunElement.getItem()).setText( "==");
//Adds the run element for box
officeMathRunElement =(IOfficeMathRunElement)
mathBox.getEquation().getFunctions().add(1, MathFunctionType.RunElement);
officeMathRunElement.setItem( new WTextRange(document));
//Sets text for box equation
((WTextRange)officeMathRunElement.getItem()).setText("adx");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Border box

You can add a box with the borders on four sides and strikethrough on horizontal, vertical, and diagonal directions to the equation. The following code example shows how to add a border box to the equation.

### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds a border box equation
IOfficeMathBorderBox mathBorderBox =(IOfficeMathBorderBox)
officeMath.getFunctions().add(0, MathFunctionType.BorderBox);
//Sets the diagonal strikethrough from lower left to upper right
mathBorderBox.setStrikeDiagonalUp(true);
//Sets the diagonal strikethrough from upper left to lower right
mathBorderBox.setStrikeDiagonalDown(true);
//Sets the horizontal strikethrough
mathBorderBox.setStrikeHorizontal(true);
//Sets the vertical strikethrough
mathBorderBox.setStrikeVertical(true);
//Enables the flag, to hide the bottom border of an equation
mathBorderBox.setHideBottom(true);
```



```
//Enables the flag, to hide the left border of an equation
mathBorderBox.setHideLeft(true);
//Sets false to show the right border of an equation
mathBorderBox.setHideRight(false);
//Sets false to show the top border of an equation
mathBorderBox.setHideTop(false);
//Adds the run element for border box
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
mathBorderBox.getEquation().getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for border box equation
((WTextRange)officeMathRunElement.getItem()).setText("a+b-c");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Delimiter

You can add a delimiter (parenthesis, square brackets and other characters) to the equation. The following code example shows how to add a delimiter to the equation.

### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds a delimiter equation
IOfficeMathDelimiter mathDelimiter = (IOfficeMathDelimiter)
officeMath.getFunctions().add(0, MathFunctionType.Delimiter);
//Sets the begin character
mathDelimiter.setBeginCharacter("[");
//Sets the end character
mathDelimiter.setEndCharacter("]");
//Enables the flag, to grow delimiter characters to full height of the
arguments
mathDelimiter.setIsGrow(true);
//Sets the appearance of delimiters
mathDelimiter.setDelimiterShape(MathDelimiterShapeType.Match);
//Adds the run element for delimiter
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
mathDelimiter.getEquation().add(0).getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for delimiter equation
((WTextRange)officeMathRunElement.getItem()).setText("a+b");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```



### Equation array

You can create a one dimensional array of equations in Word document. The following code example shows how to create an array of equations.

#### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds an equation array
IOfficeMathEquationArray mathEquationArray = (IOfficeMathEquationArray)
officeMath.getFunctions().add(0, MathFunctionType.EquationArray);
//Sets the vertical alignment for equation array
mathEquationArray.setVerticalAlignment(MathVerticalAlignment.Center);
//Enables the flag, to distribute the equation array equally within the
container
mathEquationArray.setExpandEquationContainer(true);
//Enables the flag, to expand the equations in an equation array to the
maximum width
mathEquationArray.setExpandEquationContainer(true);
//Sets the row spacing rule
mathEquationArray.setRowSpacingRule(SpacingRule.Multiple);
//Adds the run element for equation array
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
mathEquationArray.getEquation().add(0).getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for equation) array
((WTextRange)officeMathRunElement.getItem()).setText("x+y+z=0");
//Adds the run element for equation array
officeMathRunElement = (IOfficeMathRunElement)
mathEquationArray.getEquation().add(1).getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for equation array
((WTextRange)officeMathRunElement.getItem()).setText("x+y-z=1");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Fraction

You can create a fraction equation with a numerator and denominator in Word document. The following code example shows how to create a fraction equation.

#### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
```



```

//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds a fraction getEquation()
IOfficeMathFraction mathFraction = (IOfficeMathFraction)
officeMath.getFunctions().add(0, MathFunctionType.Fraction);
//Sets the denominator for fraction
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
mathFraction.getNumerator().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("a");
//Sets the Numerator for fraction
officeMathRunElement = (IOfficeMathRunElement)
mathFraction.getDenominator().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("b");
//Sets the fraction type
mathFraction.setFractionType(MathFractionType.NormalFractionBar);
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();

```

### Function

You can create trigonometric functions in a Word document. The following code example shows how to create a function.

### JAVA

```

//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds a function
IOfficeMathFunction mathFunction = (IOfficeMathFunction)
officeMath.getFunctions().add(0, MathFunctionType.Function);
//Sets the function name
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
mathFunction.getFunctionName().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("sin");
//Adds the run element for function
officeMathRunElement = (IOfficeMathRunElement)
mathFunction.getEquation().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for function
((WTextRange)officeMathRunElement.getItem()).setText("90");

```



```
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Group character

You can group mathematical equations by adding a grouping character at above or below to the corresponding equations. The following code example shows how to create an equation with grouping character.

#### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath math = document.getLastParagraph().appendMath();
//Adds a new math
IOfficeMath officeMath = math.getMathParagraph().getMaths().add();
//Adds a group character equation
IOfficeMathGroupCharacter officeMathGroupCharacter
=(IOfficeMathGroupCharacter) officeMath.getFunctions().add(0,
MathFunctionType.GroupCharacter);
//Sets the group character
officeMathGroupCharacter.setGroupCharacter("⏟");
//Enables the flag to align group character at top
officeMathGroupCharacter.setHasAlignTop(true);
//Enables the flag to align the text and group character
officeMathGroupCharacter.setHasCharacterTop(true);
//Adds the run element for group character
IOfficeMathRunElement officeMathRunElement =(IOfficeMathRunElement)
officeMathGroupCharacter.getEquation().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for group character equation
((WTextRange)officeMathRunElement.getItem()).setText("a-b");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Limit

You can add upper limit or lower limit to the mathematical equation. The following code example shows how to create limit equation.

#### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath wMath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wMath.getMathParagraph().getMaths().add();
```



```

//Adds function to the math
IOfficeMathFunction officeMathFunction =(IOfficeMathFunction)
officeMath.getFunctions().add(0, MathFunctionType.Function);
//Adds a mathematical limit equation
IOfficeMathLimit officeMathLimit =(IOfficeMathLimit)
officeMathFunction.getFunctionName().getFunctions().add(0,
MathFunctionType.Limit);
IOfficeMathRunElement officeMathRunElement =(IOfficeMathRunElement)
officeMathLimit.getEquation().getFunctions().add(MathFunctionType.RunElement
);
officeMathRunElement.setItem( new WTextRange(document));
//Sets text for limit equation.
((WTextRange)officeMathRunElement.getItem()).setText("lim");
//Sets the type of the limit.
officeMathLimit.setLimitType(MathLimitType.LowerLimit);
IOfficeMathRunElement officeMathRunElement_limit =(IOfficeMathRunElement)
officeMathLimit.getLimit().getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement_limit.setItem(new WTextRange(document));
//Sets the Limit value.
((WTextRange)officeMathRunElement_limit.getItem()).setText("n=0");
officeMathLimit.setLimitType( MathLimitType.LowerLimit);
officeMathRunElement =(IOfficeMathRunElement)
officeMathFunction.getEquation().getFunctions().add(MathFunctionType.RunElem
ent);
officeMathRunElement.setItem( new WTextRange(document));
//Sets text for base of the specified equation
((WTextRange)officeMathRunElement.getItem()).setText("x");
//Saves the Word document
document.save("Sample.docx");
//Closes the document
document.close();

```

### Matrix

You can create a matrix equation in a Word document. The following code example shows how to create a matrix equation.

### JAVA

```

//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath wmath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wmath.getMathParagraph().getMaths().add();
//Adds matrix getEquation()
IOfficeMathMatrix mathMatrix = (IOfficeMathMatrix)
officeMath.getFunctions().add(MathFunctionType.Matrix);
//Sets vertical alignment for matrix
mathMatrix.setVerticalAlignment(MathVerticalAlignment.Center);
//Sets width for matrix columns
mathMatrix.setColumnWidth(1);
//Sets column spacing rule
mathMatrix.setColumnSpacingRule(SpacingRule.OneAndHalf);
//Sets column spacing value
mathMatrix.setColumnSpacing(3);

```



```

//Enables the flag to hide place holders
mathMatrix.setHidePlaceHolders(true);
//Sets row spacing rule.
mathMatrix.setRowSpacingRule(SpacingRule.Double);
//Sets row spacing value.
mathMatrix.setRowSpacing(2);
//Adds a new column
mathMatrix.getColumns().add();
//Adds a new row
mathMatrix.getRows().add();
//Sets horizontal alignment for column
mathMatrix.getColumns().get(0).setHorizontalAlignment(MathHorizontalAlignmen
t.Left);
//Gets an argument in first cell in first row
officeMath = mathMatrix.getRows().get(0).getArguments().get(0);
//Sets text for argument in first cell in first row
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
officeMath.getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("1");
//Adds a new column
mathMatrix.getColumns().add();
//Adds a new row
mathMatrix.getRows().add();
//Gets an argument in second cell in first row
officeMath = mathMatrix.getRows().get(0).getArguments().get(1);
//Sets text for argument in second cell in first row
officeMathRunElement = (IOfficeMathRunElement)
officeMath.getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("2");
//Gets an argument in first cell in second row
officeMath = mathMatrix.getRows().get(1).getArguments().get(0);
//Sets text for argument in first cell in second row
officeMathRunElement = (IOfficeMathRunElement)
officeMath.getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("3");
//Gets an argument in second cell in second row
officeMath = mathMatrix.getRows().get(1).getArguments().get(1);
//Sets text for argument in second cell in second row
officeMathRunElement = (IOfficeMathRunElement)
officeMath.getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("4");
//Saves the Word document.
document.save("Sample.docx");
//Closes the document
document.close();

```

### N-Array

You can create an equation with common large operators such as summation, integrals, union, intersection, logical OR, logical AND, products and co-products. The following code example shows how to create a summation with limits.



**JAVA**

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath wMath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wMath.getMathParagraph().getMaths().add();
//Adds a N-Array equation
IOfficeMathNArray officeMathNArray = (IOfficeMathNArray)
officeMath.getFunctions().add(0, MathFunctionType.NArray);
//Sets N-Array character.
officeMathNArray.setNArrayCharacter("Σ");
//Enables the flag, to grow N-array character to full height of the
arguments
officeMathNArray.setHasGrow(false);
//Enables the flag to hide lower limit
officeMathNArray.setHideLowerLimit(false);
//Enables the flag to hide upper limit
officeMathNArray.setHideUpperLimit(false);
//Sets false to set limit position on above the summation
officeMathNArray.setSubSuperscriptLimit(false);
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
officeMathNArray.getSubscript().getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for superscript property of NArray equation
((WTextRange)officeMathRunElement.getItem()).setText("n=1");
officeMathRunElement = (IOfficeMathRunElement)
officeMathNArray.getSuperscript().getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("10");
officeMathRunElement = (IOfficeMathRunElement)
officeMathNArray.getEquation().getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for NArray equation
((WTextRange)officeMathRunElement.getItem()).setText("x");
//Saves the Word document.
document.save("Sample.docx");
//Closes the document
document.close();
```

*Radical*

You can create a radical equation in Word document. The following example shows how to create a radical equation.

**JAVA**

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
```



```

WMath wmath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wmath.getMathParagraph().getMaths().add();
//Adds a radical equation
IOfficeMathRadical officeMathRadical = (IOfficeMathRadical)
officeMath.getFunctions().add(0, MathFunctionType.Radical);
//Sets false to show degree in radical
officeMathRadical.setHideDegree(false);
//Adds a degree for radical equation
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
officeMathRadical.getDegree().getFunctions().add(MathFunctionType.RunElement
);
officeMathRunElement.setItem( new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("2");
//Adds an run element for radical
officeMathRunElement = (IOfficeMathRunElement)
officeMathRadical.getEquation().getFunctions().add(MathFunctionType.RunElement
);
officeMathRunElement.setItem(new WTextRange(document));
//Sets the radicand text for radical equation
((WTextRange)officeMathRunElement.getItem()).setText("x");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();

```

### Phantom

You can create a phantom equation to add the spacing of the phantom

without displaying that base and suppressing part of the glyph from spacing considerations. The following code example shows how to create a phantom equation.

### JAVA

```

//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath wmath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wmath.getMathParagraph().getMaths().add();
//Adds a radical equation
IOfficeMathRadical officeMathRadical = (IOfficeMathRadical)
officeMath.getFunctions().add(0, MathFunctionType.Radical);
//Adds a degree for radical
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
officeMathRadical.getDegree().getFunctions().add(MathFunctionType.RunElement
);
officeMathRunElement.setItem( new WTextRange(document));
((WTextRange)officeMathRunElement.getItem()).setText("2");
//Adds a phantom equation in radical.
IOfficeMathPhantom officeMathPhantom = (IOfficeMathPhantom)
officeMathRadical.getEquation().getFunctions().add(0,
MathFunctionType.Phantom);
//Enables the flag, to show the contents of phantom
officeMathPhantom.setShow(true);
//Enables the flag, to transparent the phantom

```



```

officeMathPhantom.setTransparent(true);
//Enables the flag, to ignore the ascent of the phantom contents in spacing
officeMathPhantom.setZeroAscent(true);
//Enables the flag, to ignore the descent of the phantom contents in spacing
officeMathPhantom.setZeroDescent(true);
//Enables the flag, to ignore the width of a phantom contents in spacing
officeMathPhantom.setZeroWidth(true);
//Adds a run element for phantom
officeMathRunElement = (IOfficeMathRunElement)
officeMathPhantom.getEquation().getFunctions().add(MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for phantom equation
((WTextRange)officeMathRunElement.getItem()).setText("a-b");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();

```

### SubSuperscript

You can add a superscript or subscript equation in a Word document. The following code shows how to create a superscript equation.

### JAVA

```

//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath wmath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wmath.getMathParagraph().getMaths().add();
//Adds a superscript equation
IOfficeMathScript officeMathScript = (IOfficeMathScript)
officeMath.getFunctions().add(0, MathFunctionType.SubSuperscript);
//Sets the type of the script as superscript.
officeMathScript.setScriptType(MathScriptType.Superscript);
//Adds a run element for superscript.
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
officeMathScript.getScript().getFunctions().add(MathFunctionType.RunElement);
;
officeMathRunElement.setItem(new WTextRange(document));
WTextRange textRange = (WTextRange) officeMathRunElement.getItem();
//Sets text for superscript.
textRange.setText("2");
//Adds run element for equation
officeMathRunElement = (IOfficeMathRunElement)
officeMathScript.getEquation().getFunctions().add(MathFunctionType.RunElement);
//Sets text for equation
((WTextRange)officeMathRunElement.getItem()).setText("x");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();

```



### Left SubSuperscript

You can add superscript and subscript on the left side of mathematical equation. The following code example shows how to add superscript and subscript on the left side of the equation.

#### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath wmath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wmath.getMathParagraph().getMaths().add();
//Adds a left subsuperscript equation
IOfficeMathLeftScript officeMathLeftSubScript = (IOfficeMathLeftScript)
officeMath.getFunctions().add(0, MathFunctionType.LeftSubSuperscript);
//Adds run element for left subscript
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
officeMathLeftSubScript.getSubscript().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for subscript
((WTextRange)officeMathRunElement.getItem()).setText("1");
//Adds a run element for left superscript
officeMathRunElement = (IOfficeMathRunElement)
officeMathLeftSubScript.getSuperscript().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for left superscript.
((WTextRange)officeMathRunElement.getItem()).setText("n");
officeMathRunElement = (IOfficeMathRunElement)
officeMathLeftSubScript.getEquation().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for equation
((WTextRange)officeMathRunElement.getItem()).setText("Y");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();
```

### Right SubSuperscript

You can add superscript and subscript on the right side of mathematical equation. The following code example shows how to add superscript and subscript on the right side of the equation.

#### JAVA

```
//Creates a new Word document
WordDocument document = new WordDocument();
//Adds one section and one paragraph to the document
document.ensureMinimal();
//Appends a new mathematical equation to the paragraph
WMath wmath = document.getLastParagraph().appendMath();
IOfficeMath officeMath = wmath.getMathParagraph().getMaths().add();
//Adds a right subsuperscript equation
```



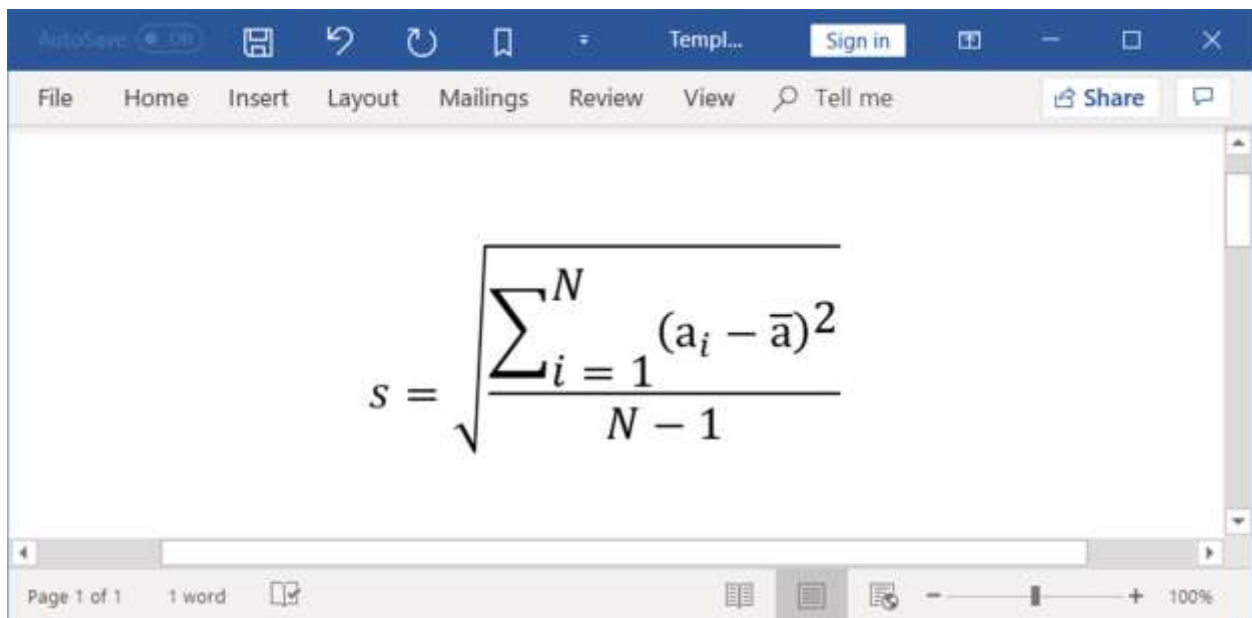
```

IOfficeMathRightScript officeMathRightScript = (IOfficeMathRightScript)
officeMath.getFunctions().add(0, MathFunctionType.RightSubSuperscript);
//Sets false to align subscript and superscript horizontally
officeMathRightScript.setIsSkipAlign(true);
//Adds run element for right subscript.
IOfficeMathRunElement officeMathRunElement = (IOfficeMathRunElement)
officeMathRightScript.getSubscript().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for right subscript
((WTextRange)officeMathRunElement.getItem()).setText("1");
//Adds run element for right superscript
officeMathRunElement = (IOfficeMathRunElement)
officeMathRightScript.getSuperscript().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for right superscript.
((WTextRange)officeMathRunElement.getItem()).setText("n");
officeMathRunElement = (IOfficeMathRunElement)
officeMathRightScript.getEquation().getFunctions().add(0,
MathFunctionType.RunElement);
officeMathRunElement.setItem(new WTextRange(document));
//Sets text for equation
((WTextRange)officeMathRunElement.getItem()).setText("Y");
//Saves the Word document
document.save("Sample.docx", FormatType.Docx);
//Closes the document
document.close();

```

### Modify existing equation

You can add or modify the text and formatting of existing mathematical equation in Word document. The following screenshots shows an existing mathematical equation in the input Word document.





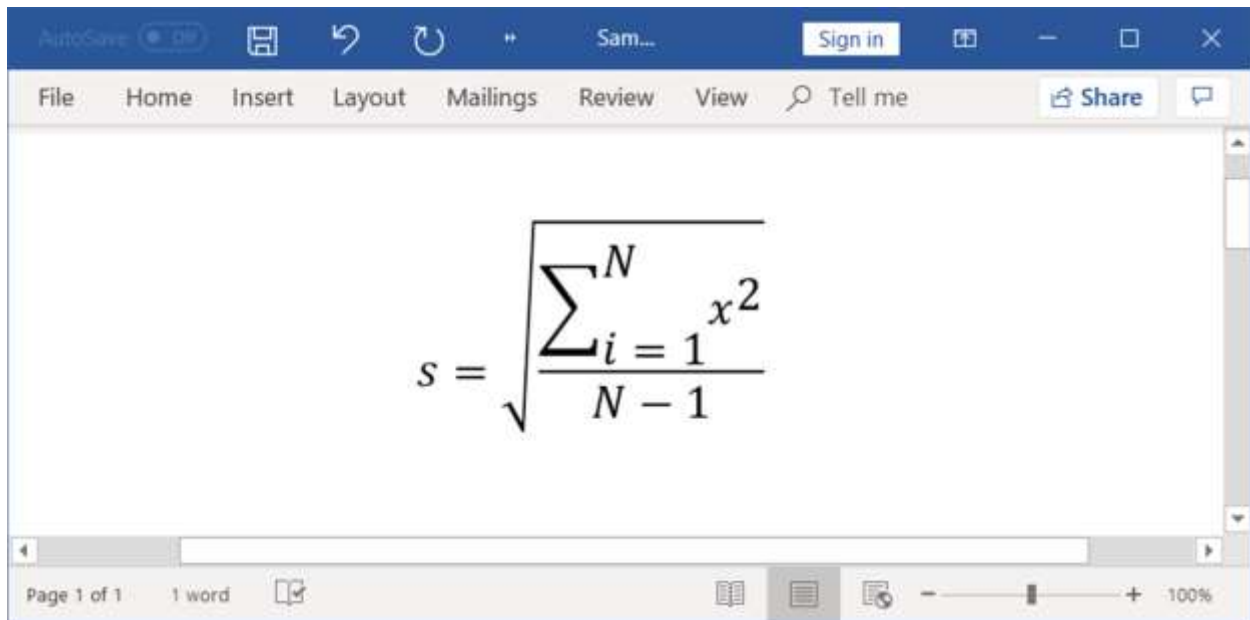
The following code example shows how to modify an existing mathematical equation in the Word document.

#### JAVA

```
//Opens an existing Word document
WordDocument document = new WordDocument("Template.docx");
//Access the paragraph from Word document
WParagraph paragraph = (WParagraph)
document.getLastSection().getBody().getChildEntities().get(0);
//Access the mathematical equation from the paragraph
WMath math = (WMath) paragraph.getChildEntities().get(0);
//Access the radical equation
IOfficeMathRadical mathRadical = (IOfficeMathRadical)
math.getMathParagraph().getMaths().get(0).getFunctions().get(1);
//Access the fraction equation in radical
IOfficeMathFraction mathFraction = (IOfficeMathFraction)
mathRadical.getEquation().getFunctions().get(0);
//Access the n-array equation in fraction
IOfficeMathNArray mathNArray = (IOfficeMathNArray)
mathFraction.getNumerator().getFunctions().get(0);
//Access the math script in n-array
IOfficeMathScript mathScript = (IOfficeMathScript)
mathNArray.getEquation().getFunctions().get(0);
//Access the delimiter in math script
IOfficeMathDelimiter mathDelimiter = (IOfficeMathDelimiter)
mathScript.getEquation().getFunctions().get(0);
//Removes the delimiter
mathScript.getEquation().getFunctions().remove(mathDelimiter);
//Modifies the run element in math script
IOfficeMathRunElement MathParagraphItem = (IOfficeMathRunElement)
mathScript.getEquation().getFunctions().add(MathFunctionType.RunElement);
MathParagraphItem.setItem( new WTextRange(document));
//Sets the text value
((WTextRange)MathParagraphItem.getItem()).setText("x");
//Applies character format to the text
((WTextRange)MathParagraphItem.getItem()).getCharacterFormat().setItalic(true);
((WTextRange)MathParagraphItem.getItem()).getCharacterFormat().setFontSize(20);
//Applies math format to the text
MathParagraphItem.getMathFormat().setStyle(MathStyleType.Italic);
//Saves the word document
document.save("Sample.docx");
//close the word document
document.close();
```

By executing the above code example, it generates output Word document as follows.





### Accepting or Rejecting Track Changes

It is used to keep track of the changes made to a Word document. It helps to maintain the record of author, name and time for every insertion, deletion, or modification in a document. This can be enabled by using the TrackChanges property of the Word document.

#### Note:

With this support, the changes made in the Word document by DocIO library cannot be tracked.

The following code example illustrates how to enable track changes of the document.

#### JAVA

```
//Creates a new Word document.
WordDocument document = new WordDocument();
//Adds new section to the document.
IWSection section = document.addSection();
//Adds new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Appends text to the paragraph.
IWTextRange text = paragraph.appendText("This sample illustrates how to
track the changes made to the word document. ");
//Sets font name and size for text.
text.getCharacterFormat().setFontName("Times New Roman");
text.getCharacterFormat().setFontSize((float)14);
text=paragraph.appendText("This track changes is useful in shared
environment.");
text.getCharacterFormat().setFontSize((float)12);
//Turns on the track changes option.
document.setTrackChanges(true);
//Saves and closes the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```



### Accept all changes

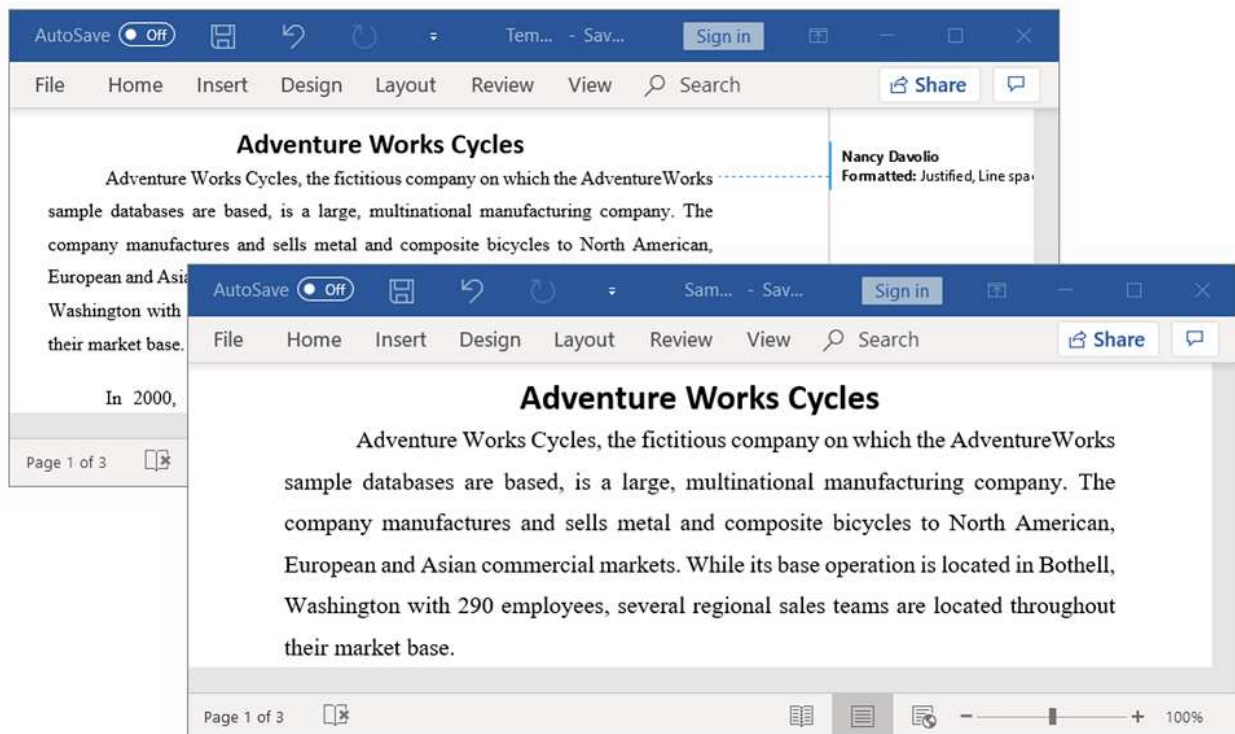
You can **accept all track changes in Word document** using `acceptAll` method.

The following code example shows how to accept all the tracked changes.

#### JAVA

```
//Opens an existing Word document.  
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);  
//Accepts all the tracked changes revisions.  
if(document.getHasChanges())  
document.getRevisions().acceptAll();  
//Saves and closes the document.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

By executing the above code example, it generates output Word document as follows.



### Reject all changes

You can **reject all track changes in Word document** using `rejectAll` method.

The following code example shows how to reject all the tracked changes.

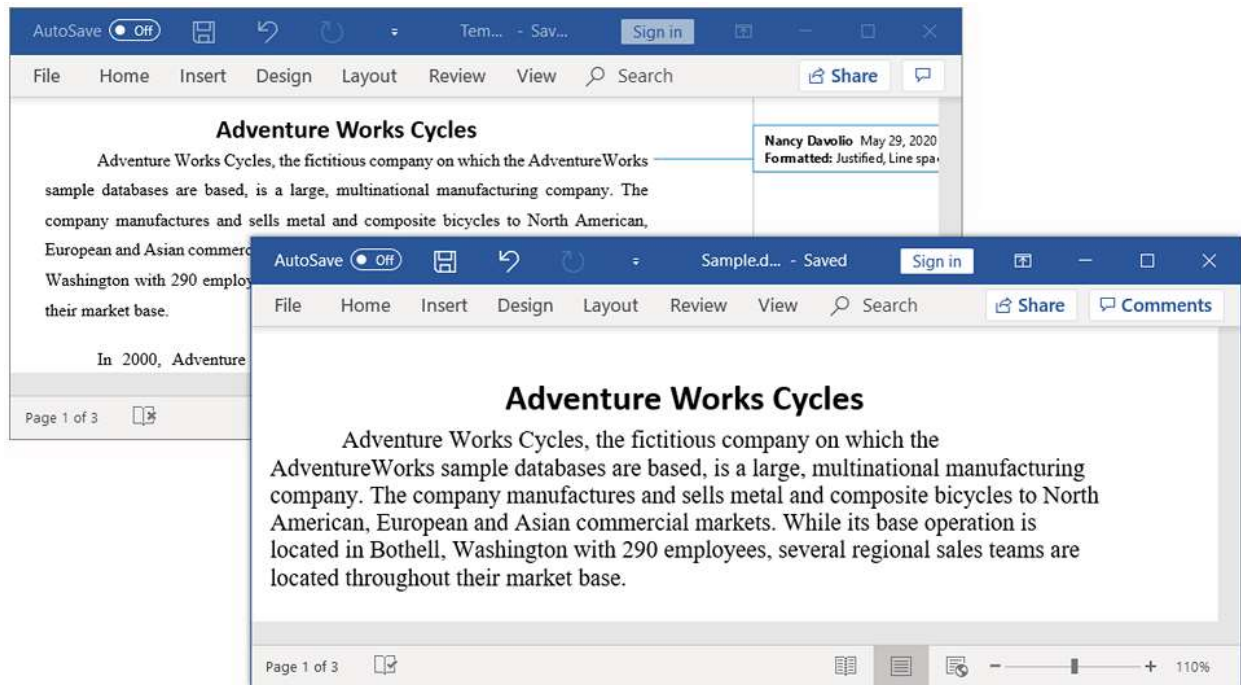
#### JAVA

```
//Opens an existing Word document.  
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);  
//Rejects all the tracked changes revisions.  
if(document.getHasChanges())  
document.getRevisions().rejectAll();  
//Saves and closes the document.  
document.save("Sample.docx", FormatType.Docx);
```



```
document.close();
```

By executing the above code example, it generates output Word document as follows.



Accept all changes by a particular reviewer

You can **accept all changes made by the author** in the Word document using `accept` method.

The following code example shows how to accept the tracked changes made by the author.

### JAVA

```
//Opens an existing Word document.
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Iterates into all the revisions in Word document.
for (int i = document.getRevisions().getCount() - 1; i >= 0; i--)
{
    //Checks the author of current revision and accepts it.
    if (document.getRevisions().get(i).getAuthor().equals("Nancy Davolio"))
        document.getRevisions().get(i).accept();
    //Resets to last item when accept the moving related revisions.
    if (i > document.getRevisions().getCount() - 1)
        i = document.getRevisions().getCount();
}
//Saves and closes the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

Reject all changes by particular reviewer

You can **reject all changes made by the author** in the Word document using `reject` method.

The following code example shows how to reject the tracked changes made by the author.



**JAVA**

```
//Opens an existing Word document.
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Iterates into all the revisions in Word document.
for (int i = document.getRevisions().getCount() - 1; i >= 0; i--)
{
    //Checks the author of current revision and rejects it.
    if(document.getRevisions().get(i).getAuthor().equals("Nancy Davolio"))
        document.getRevisions().get(i).reject();
    //Resets to last item when reject the moving related revisions.
    if (i > document.getRevisions().getCount() - 1)
        i = document.getRevisions().getCount();
}
//Saves and closes the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

## Revision information

You can get the **revision information of track changes** in the Word document like author name, date, and type of revision.

The following code example shows how to get the details about the revision information of track changes.

**JAVA**

```
//Opens an existing Word document.
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Accesses the first revision in the word document.
Revision revision = document.getRevisions().get(0);
//Gets the name of the user who made the specified tracked change.
String author = revision.getAuthor();
// Gets the date and time that the tracked change was made.
LocalDateTime dateTime = revision.getDate();
// Gets the type of the track changes revision.
RevisionType revisionType = revision.getRevisionType();
//Closes the document.
document.close();
```

## Frequently Asked Questions

- [How to check whether a Word document contains tracked changes or not?](#)
- [How to accept or reject track changes of specific type in the Word document?](#)
- [How to enable track changes for Word document?](#)

## Word Document Conversion

## Working with Document Conversions

The Essential DocIO converts documents from one format to another format. Each file format document can be categorized as flow layout document.

**Flow layout document**



- A flow document is designed to "reflow content" depending on the application.
- Does not contain any information about the position of its content.
- Dynamically renders the content by application at run time.
- Example: DOCX, HTML and TEXT file formats.

Essential DocIO can convert various flow document as fixed document by using our layout engine. Following conversions are supported by Essential DocIO.

- Microsoft Word file format Conversions.
- Text Conversions.
- HTML Conversions.

### HTML conversion

Essential DocIO supports converting the HTML file into Word document and vice versa. It supports only the HTML files that meet the validation either against XHTML 1.0 strict or XHTML 1.0 Transitional schema.

For further information kindly refer this [link](#).

### Supported Document elements

Kindly refer to this [link](#) for the document elements and attributes that are supported by DocIO in the Word to HTML and HTML to Word conversions.

### Text file

Essential DocIO supports to convert the Word document into a Text file and vice versa. For further information, kindly refer to this [link](#).

## Supported and Unsupported Features

This section describes the support and unsupported elements in the DocIO.

### Word document features

| Element                      | DOCX, DOTX, DOTM, DOCM, WordML |       | RTF  |       |
|------------------------------|--------------------------------|-------|------|-------|
|                              | Read                           | Write | Read | Write |
| Built-in document properties | Yes                            | Yes   | Yes  | No    |
| Custom document properties   | Yes                            | Yes   | No   | No    |
| Mail merge                   | Yes                            | Yes   | Yes  | Yes   |
| Encryption and Decryption    | Yes                            | Yes   | N/A  | N/A   |
| View setup                   | Yes                            | Yes   | Yes  | No    |



| Element            | DOCX, DOTX, DOTM,<br>DOCM, WordML |   | RTF  |       |
|--------------------|-----------------------------------|---|------|-------|
|                    | Read                              | Write   | Read | Write |
| Watermark          | Yes                               | Yes   | No   | Yes   |
| Track changes      | Yes                               | Yes â€”<br>limited [can<br>only<br>accept/reject] | No   | No    |
| Themes             | Yes                               | Yes   | No   | No    |
| Document variables | Yes                               | Yes   | No   | No    |
| Macros             | Yes                               | Yes   | No   | No    |

### Section features

| Element                  | DOCX, DOTX, DOTM,<br>DOCM, WordML |       | RTF  |       |
|--------------------------|-----------------------------------|-------|------|-------|
|                          | Read                              | Write | Read | Write |
| Headers & footers        | Yes                               | Yes   | Yes  | Yes   |
| Section break            | Yes                               | Yes   | Yes  | Yes   |
| Text columns             | Yes                               | Yes   | Yes  | Yes   |
| Page margin              | Yes                               | Yes   | Yes  | Yes   |
| Page border              | Yes                               | Yes   | No   | Yes   |
| Page orientation         | Yes                               | Yes   | Yes  | Yes   |
| Page size                | Yes                               | Yes   | Yes  | Yes   |
| Header & footer distance | Yes                               | Yes   | No   | No    |
| Vertical alignment       | Yes                               | Yes   | Yes  | Yes   |



*Paragraph elements features*

| Element  | DOCX, DOTX, DOTM, DOCM, WordML |       | RTF  |       |
|--|--------------------------------|-------|------|-------|
|  | Read                           | Write | Read | Write |
| Break – Page break, column break and Line break  | Yes                            | Yes   | Yes  | Yes   |
| Symbols  | Yes                            | Yes   | No   | Yes   |
| Comments   | Yes                            | Yes   | Yes  | Yes   |
| Footnote and endnote                             | No                             | No    | No   | No    |
| Pictures (.bmp, .jpg, .png, .emf, .tif and .gif) | Yes                            | Yes   | Yes  | Yes   |
| Text boxes                                       | Yes                            | Yes   | Yes  | Yes   |
| OLE objects                                      | No                             | No    | No   | No    |
| Predefined shapes                                | Yes                            | Yes   | Yes  | Yes   |
| Custom shapes                                    | No                             | No    | No   | No    |
| Grouped shapes                                   | Yes                            | Yes   | No   | No    |
| Equation   | No                             | No    | No   | No    |
| SmartArt   | No                             | No    | No   | No    |
| WordArt  | No                             | No    | No   | No    |
| Bookmark   | Yes                            | Yes   | Yes  | Yes   |
| Hyperlink  | Yes                            | Yes   | Yes  | Yes   |
| Chart  | No                             | No    | No   | No    |
| Fields   | Yes                            | Yes   | Yes  | Yes   |
| Form Fields-TextInput, CheckBox & DropDown       | Yes                            | Yes   | Yes  | Yes   |



*Paragraph features*

| Element                                     | DOCX, DOTX, DOTM,<br>DOCM, WordML |       | RTF  |       |
|---|-----------------------------------|-------|------|-------|
|   | Read                              | Write | Read | Write |
| Paragraph style                             | Yes                               | Yes   | Yes  | Yes   |
| Alignment “ left, right, center and justify | Yes                               | Yes   | Yes  | Yes   |
| Right to left paragraph                     | Yes                               | Yes   | Yes  | Yes   |
| Lists - bullets and numbers                 | Yes                               | Yes   | Yes  | Yes   |
| Run properties for the paragraph mark       | Yes                               | Yes   | Yes  | Yes   |
| Suppress hyphenation                        | No                                | No    | No   | No    |
| Indents “ left, right, first line & hanging | Yes                               | Yes   | Yes  | Yes   |
| Spacing “ before, after & auto              | Yes                               | Yes   | Yes  | Yes   |
| Keeps and breaks                            | Yes                               | Yes   | Yes  | Yes   |
| Text frames                                 | Yes                               | Yes   | Yes  | Yes   |
| Tabs  | Yes                               | Yes   | Yes  | Yes   |
| Borders                                     | Yes                               | Yes   | Yes  | Yes   |
| Shading                                     | Yes                               | Yes   | Yes  | Yes   |

*Style features*

| Element | DOCX, DOTX, DOTM,<br>DOCM, WordML |       | RTF  |       |
|---------|-----------------------------------|-------|------|-------|
|         | Read                              | Write | Read | Write |



| Element                  | DOCX, DOTX, DOTM,<br>DOCM, WordML |     | RTF |     |
|--------------------------|-----------------------------------|-----|-----|-----|
|                          |                                   |     |     |     |
| Built-in paragraph style | Yes                               | Yes | Yes | Yes |
| Custom paragraph style   | Yes                               | Yes | Yes | Yes |
| Built-in character style | Yes                               | Yes | Yes | Yes |
| Custom character style   | Yes                               | Yes | Yes | Yes |
| Built-in list style      | Yes                               | Yes | Yes | Yes |
| Custom list style        | Yes                               | Yes | Yes | Yes |
| Built-in table style     | Yes                               | Yes | No  | No  |
| Custom table style       | Yes                               | Yes | No  | No  |

*Table features*

| Element                      | DOCX, DOTX, DOTM,<br>DOCM, WordML |       | RTF  |       |
|------------------------------|-----------------------------------|-------|------|-------|
|                              | Read                              | Write | Read | Write |
| Nested tables                | Yes                               | Yes   | Yes  | Yes   |
| Right to left tables         | Yes                               | Yes   | Yes  | Yes   |
| Table style                  | Yes                               | Yes   | No   | No    |
| Conditional formatting style | Yes                               | Yes   | No   | No    |
| Table alignment              | Yes                               | Yes   | Yes  | Yes   |
| Table indent                 | Yes                               | Yes   | Yes  | Yes   |
| Allow AutoFit                | Yes                               | Yes   | Yes  | Yes   |
| Cell margins                 | Yes                               | Yes   | Yes  | Yes   |
| Cell spacing                 | Yes                               | Yes   | Yes  | Yes   |



| Element                            | DOCX, DOTX, DOTM,<br>DOCM, WordML |     | RTF |     |
|------------------------------------|-----------------------------------|-----|-----|-----|
|                                    |                                   |     |     |     |
| Preferred table width              | Yes                               | Yes | Yes | Yes |
| Table shading                      | Yes                               | Yes | Yes | Yes |
| Table borders                      | Yes                               | Yes | Yes | Yes |
| Floating tables                    | Yes                               | Yes | Yes | Yes |
| Allow row to break across pages    | Yes                               | Yes | Yes | Yes |
| Repeat as header row               | Yes                               | Yes | Yes | Yes |
| Height                             | Yes                               | Yes | Yes | Yes |
| Height type                        | Yes                               | Yes | Yes | Yes |
| Cell borders                       | Yes                               | Yes | Yes | Yes |
| Cell shading                       | Yes                               | Yes | Yes | Yes |
| Wrap text                          | Yes                               | Yes | Yes | Yes |
| Fit text                           | Yes                               | Yes | Yes | Yes |
| Horizontal & vertical merged cells | Yes                               | Yes | Yes | Yes |
| Vertical alignment                 | Yes                               | Yes | Yes | Yes |
| Text direction                     | Yes                               | Yes | Yes | Yes |

*Text features*

| Element         | DOCX, DOTX, DOTM,<br>DOCM, WordML |       | RTF  |       |
|-----------------|-----------------------------------|-------|------|-------|
|                 | Read                              | Write | Read | Write |
| Character style | Yes                               | Yes   | Yes  | Yes   |
| Color           | Yes                               | Yes   | Yes  | Yes   |



|                       |     |     |     |     |
|-----------------------|-----|-----|-----|-----|
| Highlight color       | Yes | Yes | Yes | Yes |
| Language              | Yes | Yes | Yes | Yes |
| Border                | Yes | Yes | No  | Yes |
| Shading               | Yes | Yes | Yes | Yes |
| Font                  | Yes | Yes | Yes | Yes |
| Underline             | Yes | Yes | Yes | Yes |
| Strikethrough         | Yes | Yes | Yes | Yes |
| Double strikethrough  | Yes | Yes | Yes | Yes |
| Subscript/Superscript | Yes | Yes | Yes | Yes |
| Small caps            | Yes | Yes | Yes | Yes |
| All caps              | Yes | Yes | Yes | Yes |
| Hidden text           | Yes | Yes | Yes | Yes |
| Scale                 | Yes | Yes | Yes | Yes |
| Expanded / Compressed | Yes | Yes | Yes | No  |
| Vertical Position     | Yes | Yes | No  | Yes |

## Frequently Asked Questions

The frequently asked questions in Essential DocIO are listed below.

### How to modify an existing style?

The following code illustrates how to modify the built-in style while creating new Word document.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Create built-in style and modifies its properties
Style style = (Style)Style.createBuiltinStyle(BuiltinStyle.Heading1,
document);
style.getCharacterFormat().setItalic(true);
style.getCharacterFormat().setTextColor(ColorSupport.getDarkGreen());
//Add it to the styles collection.
```



```
document.getStyles().add(style);
//Add new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
IWTextRange text = paragraph.appendText("A built-in style is modified and is
applied to this paragraph.");
//Apply the new style to paragraph.
paragraph.applyStyle(style.getName());
//Save the Word document.
document.save("Sample.docx", FormatType.Docx);
//Closes the document.
document.close();
```

### How to set OpenType Font Features?

The Open type features provide special effects for the text. This feature is specific to Word 2010 and later version documents. The OpenType features includes the following:

- Ligatures – combination of characters, written as glyph
- Use Contextual Alternates – combination of letters based on surrounding characters
- Number spacing – specifies number width
- Number forms – specifies number height
- Stylistic sets – specifies the look of the text, based on the font used

The following code illustrates how to set ligature types for text.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add new text.
IWTextRange text = paragraph.appendText("Text to describe discretionary
ligatures");
//Set ligature type.
text.getCharacterFormat().setLigatures(LigatureType.Discretionary);
text.getCharacterFormat().setFontName("Arial");
paragraph = section.addParagraph();
text = paragraph.appendText("Text to describe contextual ligatures");
text.getCharacterFormat().setLigatures(LigatureType.Contextual);
text.getCharacterFormat().setFontName("Arial");
paragraph = section.addParagraph();
text = paragraph.appendText("Text to describe historical ligatures");
text.getCharacterFormat().setLigatures(LigatureType.Historical);
text.getCharacterFormat().setFontName("Arial");
//Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

The following code example illustrates how to set contextual alternates.

#### **JAVA**



```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add new text.
IWTextRange text = paragraph.appendText("Text to describe contextual
alternates");
text.getCharacterFormat().setFontName("Segoe Script");
//Set contextual alternates.
text.getCharacterFormat().setUseContextualAlternates(true);
paragraph = section.addParagraph();
//Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

The following code example illustrates how to set number spacing.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add new text.
IWTextRange text = paragraph.appendText("Numbers to describe tabular number
spacing 0123456789");
text.getCharacterFormat().setFontName("Calibri");
//Set number spacing.
text.getCharacterFormat().setNumberSpacing(NumberSpacingType.Tabular);
paragraph = section.addParagraph();
text = paragraph.appendText("Numbers to describe proportional number spacing
0123456789");
text.getCharacterFormat().setFontName("Calibri");
//Set number spacing.
text.getCharacterFormat().setNumberSpacing(NumberSpacingType.Proportional);
//Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

The following code example illustrates how to set number style.

#### **JAVA**

```
//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add new text.
IWTextRange text = paragraph.appendText("Numbers to describe oldstyle number
form 0123456789");
```



```

text.getCharacterFormat().setFontName("Calibri");
//Set number style.
text.getCharacterFormat().setNumberForm(NumberFormType.OldStyle);
paragraph = section.addParagraph();
text = paragraph.appendText("Numbers to describe lining number form
0123456789");
text.getCharacterFormat().setFontName("Calibri");
//Set number style.
text.getCharacterFormat().setNumberForm(NumberFormType.Lining);
//Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

The following code example illustrates how to set different styles for the text.

### **JAVA**

```

//Create a new Word document.
WordDocument document = new WordDocument();
//Add new section to the document.
IWSection section = document.addSection();
//Add new paragraph to the section.
IWParagraph paragraph = section.addParagraph();
//Add new text.
IWTextRange text = paragraph.appendText("Text to describe stylistic sets");
text.getCharacterFormat().setFontName("Gabriola");
//Set stylistic set.
text.getCharacterFormat().setStylisticSet(StylisticSetType.StylisticSet06);
paragraph = section.addParagraph();
//Add new text.
text = paragraph.appendText("Text to describe stylistic sets");
text.getCharacterFormat().setFontName("Gabriola");
//Set stylistic set.
text.getCharacterFormat().setStylisticSet(StylisticSetType.StylisticSet15);
//Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();

```

### How to insert a table from HTML string in Word document?

An HTML string can be inserted to the Word document at text body or paragraph. The following code illustrates how to insert a table to the document from the HTML string.

### **JAVA**

```

//Loads the template document
WordDocument document = new WordDocument("Template.docx");
//Gets the text body
WTextBody textbody = document.getSections().get(0).getBody();
//Html string that represents table with two rows and two columns
String htmlString = " <table border='1'><tr><td><p>First Row First
Cell</p></td><td><p>First Row Second Cell</p></td></tr><tr><td><p>Second Row
First Cell</p></td><td><p>Second Row Second Cell</p></td></tr></table> ";
//Inserts the string to the text body
textbody.insertXHTML(htmlString);
//Saves and closes the document
document.save("Sample.docx");

```



```
document.close();
```

### How to set table cell width?

Each cell in the table can have its own width. The following code illustrates how to set the width of the cell.

#### **JAVA**

```
// Open word document.
WordDocument document = new WordDocument("Template.docx");
// Get the text body of first section.
WTextBody textbody = document.getSections().get(0).getBody();
// Get the table.
IWTable table = textbody.getTables().get(0);
// Iterate through table rows.
for (Object rows_tempObj : table.getRows())
{
    WTableRow row = (WTableRow) rows_tempObj;
    // Set width for cells.
    for (int i = 0; i < row.getCells().getCount(); i++)
    {
        WTableCell cell = row.getCells().get(i);
        if (i % 2 == 0)
            // Set width as 100 for cells in even column.
            cell.setWidth(100);
        else
            // Set width as 150 for cell in odd column.
            cell.setWidth(150);
    }
}
// Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

### How to position a table in a Word document?

You can position a table in a Word document by setting position properties. The following code illustrates how to set position properties for a table.

#### **JAVA**

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx");
//Get the text body of first section.
WTextBody textbody = document.getSections().get(0).getBody();
//Get the table.
IWTable table = textbody.getTables().get(0);
//Set the horizontal and vertical position for table.
table.getTableFormat().getPositioning().setHorizPosition(40);
table.getTableFormat().getPositioning().setVertPosition(100);
//Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```



### How to set the text direction to a table in Word document?

The contents of the table cell can be in vertical or horizontal direction. Each cell content can have different text direction. The following code illustrates how to set the text direction for the text in the table.

#### **JAVA**

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx");
//Get the text body of first section.
WTextBody textbody = document.getSections().get(0).getBody();
//Get the table
IWTable table = textbody.getTables().get(0);
//Iterate through table rows
for(Object row_tempObj : table.getRows())
{
    WTableRow row = (WTableRow)row_tempObj;
    for(Object cell_tempObj : row.getCells())
    {
        WTableCell cell = (WTableCell)cell_tempObj;
        //Set the text direction for the contents.
        cell.getCellFormat().setTextDirection(TextDirection.Vertical);
    }
}
//Save and close the document.
document.save("Sample.docx",FormatType.Docx);
document.close();
```

### How to remove headers and footers from the document?

The following code illustrates how to remove the header contents from the document.

#### **JAVA**

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx",FormatType.Docx);
//Iterate through the sections.
for(Object section_tempObj : document.getSections())
{
    WSection section = (WSection)section_tempObj;
    HeaderFooter header;
    //Get even footer of current section.
    header=section.getHeadersFooters().get(HeaderFooterType.EvenHeader);
    //Remove even footer.
    header.getChildEntities().clear();
    //Get odd footer of current section.
    header=section.getHeadersFooters().get(HeaderFooterType.OddHeader);
    //Remove odd footer.
    header.getChildEntities().clear();
    //Get first page footer.
    header=section.getHeadersFooters().get(HeaderFooterType.FirstPageHeader);
    //Remove first page footer.
    header.getChildEntities().clear();
}
//Save and close the document.
document.save("Sample.docx",FormatType.Docx);
document.close();
```



The following code illustrates how to remove the footer contents from the document.

#### JAVA

```
//Load the template document.
WordDocument document = new WordDocument("Template.docx");
//Iterate through the sections.
for(Object section_tempObj : document.getSections())
{
    WSection section = (WSection)section_tempObj;
    HeaderFooter footer;
    //Get even footer of current section.
    footer=section.getHeadersFooters().get(HeaderFooterType.EvenFooter);
    //Remove even footer.
    footer.getChildEntities().clear();
    //Get odd footer of current section.
    footer=section.getHeadersFooters().get(HeaderFooterType.OddFooter);
    //Remove odd footer.
    footer.getChildEntities().clear();
    //Get first page footer.
    footer=section.getHeadersFooters().get(HeaderFooterType.FirstPageFooter);
    //Remove first page footer.
    footer.getChildEntities().clear();
}
//Save and close the document.
document.save("Sample.docx",FormatType.Docx);
document.close();
```

Which units does Java Word library uses for measurement properties such as size, margins, etc, in a Word document?

Java Word library uses Points for measurement properties in a Word document.

Migration from Microsoft Office Automation to Essential DocIO

#### Bookmarks

Bookmarks identify the location of text in a Word document that you can name and identify for future reference.

The following code example illustrates how to insert the bookmark by using DocIO. Here, the `appendBookmarkStart()` and `appendBookmarkEnd()` methods are used to add the bookmark.

#### JAVA

```
//Create a new Word document.
WordDocument doc = new WordDocument();
//Add new section.
IWSection section = doc.addSection();
//Add new paragraph.
IWParagraph paragraph = section.addParagraph();
paragraph.appendText("Simple Bookmark");
paragraph=section.addParagraph();
paragraph.appendText("Bookmark with one ");
//Insert bookmark.
paragraph.appendBookmarkStart("one_word");
paragraph.appendText("word");
```



```
paragraph.appendBookmarkEnd("one_word");
paragraph.appendText(" selected");
//Save the document.
doc.save("Sample.docx",FormatType.Docx);
//Close the document.
doc.close();
```

### Page Numbers

Page numbers can be added to the Word document in headers or footers.

The following code example illustrates how page numbers are inserted to the footer of the Word document by using DocIO.

#### JAVA

```
//Open the Word document.
WordDocument doc = new WordDocument("Template.docx",FormatType.Docx);
//Iterate through sections.
for(Object sec_tempObj : doc.getSections())
{
    WSection sec = (WSection)sec_tempObj;
    IWParagraph para = sec.addParagraph();
    //Append page field to the paragraph.
    para.appendField("footer",FieldType.FieldPage);
    para.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Center);
    ;
    sec.getPageSetup().setPageNumberStyle(PageNumberStyle.Arabic);
    //Add paragraph to footer.
    sec.getHeadersFooters().getFooter().getParagraphs().add(para);
}
//Save the document.
doc.save("Sample.docx",FormatType.Docx);
//Close the document.
doc.close();
```

### Headers and Footers

The headers and footers can be inserted with text, graphics, and any other information that is contained in the document.

You can set the header and footer by using the HeadersFooters property in the Word document section.

To access a particular header/footer, you can use the following properties of **WHeadersFooters** class:

- FirstPageHeader
- FirstPageFooter
- OddHeader
- OddFooter
- EvenHeader
- EvenFooter

#### JAVA

```
//Open a Word document.
WordDocument doc = new WordDocument("Template.docx");
//Add header and footer to each section in the document.
```



```
for(Object sec_tempObj : doc.getSections())
{
    //Header.
    WSection sec = (WSection)sec_tempObj;
    WParagraph para = new WParagraph(doc);
    para.appendField("page",FieldType.FieldPage);
    para.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Right);
    sec.getHeadersFooters().getHeader().getParagraphs().add(para);
    //Footer.
    WParagraph para1 = new WParagraph(doc);
    para1.appendText("Internal");
    para1.getParagraphFormat().setHorizontalAlignment(HorizontalAlignment.Left);
    sec.getHeadersFooters().getFooter().getParagraphs().add(para1);
}
//Save the document.
doc.save("Sample.docx",FormatType.Docx);
//Close the document.
doc.close();
```

### Tables

Tables are used to organize information and to display the information in rows and columns. You can also add images or even other tables to the table.

The following code example shows how to insert an empty table to a Word document. The `resetCells()` method is used to specify the number of rows and columns in a table.

### JAVA

```
//Create a new Word document.
WordDocument document = new WordDocument();
IWSection section = document.addSection();
//Add a table to the document.
IWTable table = section.addTable();
table.resetCells(3, 2);
//Save the document.
document.save("Sample.docx",FormatType.Docx);
//Close the document.
document.close();
```

**Note:** For more information on creating tables using DocIO, refer to online documentation link:

[Working with Tables](#)

### Comments

Comments are used to include additional information to a paragraph or text in a Word document. Comments can be added or modified whenever needed and deleted when the comment has served its purpose.

You can insert comments to a paragraph or text in a Word document by using DocIO. The following code example shows how to insert comments to a Word document.

### JAVA

```
//Create a new Word document.
WordDocument doc = new WordDocument();
IWSection section = doc.addSection();
```



```
//Add a paragraph to the document.
IWParagraph para = section.addParagraph();
para.appendText("New Text");
//Add comment to the paragraph.
para.appendComment("Comment goes here");
//Save the document.
doc.save("Sample.docx", FormatType.Docx);
```

**Note:** For more information on working with the comments using Java Word library, you can refer to the online documentation link:

### [Working with Comments](#)

How to check whether a Word document contains tracked changes or not?

You can check whether a Word document contains tracked changes by using `HasChanges` property in Essential DocIO.

The following code example shows how to check whether a Word document contains tracked changes.

#### **JAVA**

```
//Open an existing Word document.
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Get a flag which denotes whether the Word document has track changes.
boolean hasChanges = document.getHasChanges();
//When the document has track changes, accepts all changes.
if (hasChanges)
document.getRevisions().acceptAll();
//Save and close the document.
document.save("Sample.docx", FormatType.Docx);
document.close();
```

How to accept or reject track changes of specific type in the Word document?

You can **accept or reject track changes by revision type** in the tracked changes Word document.

For example, if you like to accept or reject changes of specific revision type (insertions, deletions, formatting, move to, or move from), you can iterate into the revisions in Word document and then accept or reject the particular revision type using Essential DocIO.

The following code example shows how to accept or reject track changes of specific type in the Word document.

#### **JAVA**

```
//Open an existing Word document.
WordDocument document = new WordDocument("Template.docx", FormatType.Docx);
//Iterate into all the revisions in Word document
for (int i = document.getRevisions().getCount() - 1; i >= 0; i--)
{
// Get the type of the track changes revision.
RevisionType revisionType =
document.getRevisions().get(i).getRevisionType();
//Accept only insertion and Move from revisions changes.
if (revisionType == RevisionType.Insertions || revisionType ==
RevisionType.MoveFrom)
document.getRevisions().get(i).accept();
}
```



```
//Reset to last item when accept the moving related revisions.  
if (i > document.getRevisions().getCount() - 1)  
i = document.getRevisions().getCount();  
}  
//Save and close the document.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```

### How to enable track changes for Word document?

TrackChanges is used to keep track of the changes made to a Word document. This can be enabled by using the TrackChanges property of the Word document.

The following code example shows how to enable track changes of the document.

#### **JAVA**

```
//Create a new Word document.  
WordDocument document = new WordDocument();  
//Add new section to the document.  
IWSection section = document.addSection();  
//Add new paragraph to the section.  
IWParagraph paragraph = section.addParagraph();  
//Append text to the paragraph.  
IWTextRange text = paragraph.appendText("This sample illustrates how to  
track the changes made to the word document. ");  
//Set font name and size for text.  
text.getCharacterFormat().setFontName("Times New Roman");  
text.getCharacterFormat().setFontSize(14);  
text = paragraph.appendText("This track changes is useful in shared  
environment.");  
text.getCharacterFormat().setFontSize(12);  
//Turn on the track changes option.  
document.setTrackChanges(true);  
//Save and close the document.  
document.save("Sample.docx", FormatType.Docx);  
document.close();
```