Textbox.io for IBM WCM

Welcome to the Textbox.io for IBM WCM documentation.

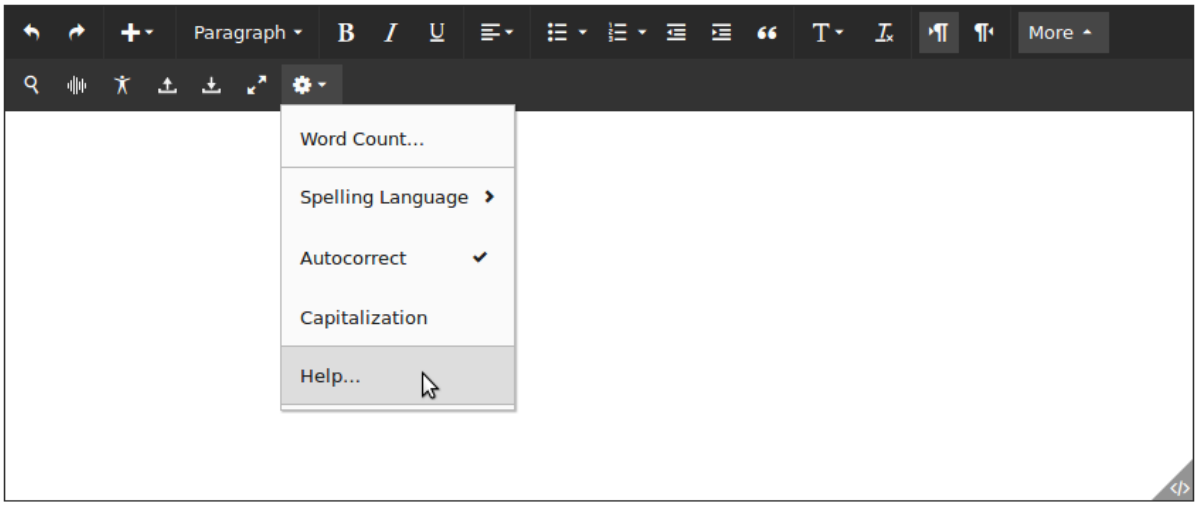
Textbox.io for IBM WCM is the next generation of content authoring for IBM WCM. The integration provides functionality including clean copy and paste with embedded images; smart media embedding with only a URL; spell checking as you type; robust table functionality and more.

This guide provides all the information that you need to get started with the Textbox.io for IBM WCM integration.

General information

For supported versions of IBM WCM and supported browsers please refer to System requirements.

Help for using the editor can be found in Textbox.io's toolbar.



For technical questions on Textbox.io please see http://docs.ephox.com/display/tbio/Textbox.io+Documentation.

Installation

Textbox.io is installed by default in WCM but by default various services and features are turned off.

- Updating Textbox.io for WCM

Customization

Once Textbox.io is installed it can be customized extensively.

- Customising Textbox.io

Tone Analyzer overview and enablement instructions

Please contact IBM if you require support.

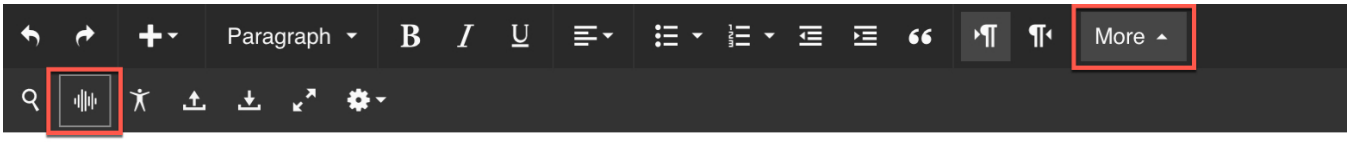
Overview

Tone Analyzer is a new Textbox service that is available to IBM WCM users as of the CF15 release. Tone Analyzer processes text and provides feedback around the emotions that are likely to be experienced by people who read your content.

Please note that Tone Analyzer is disabled by default; enablement instructions can be found further below.

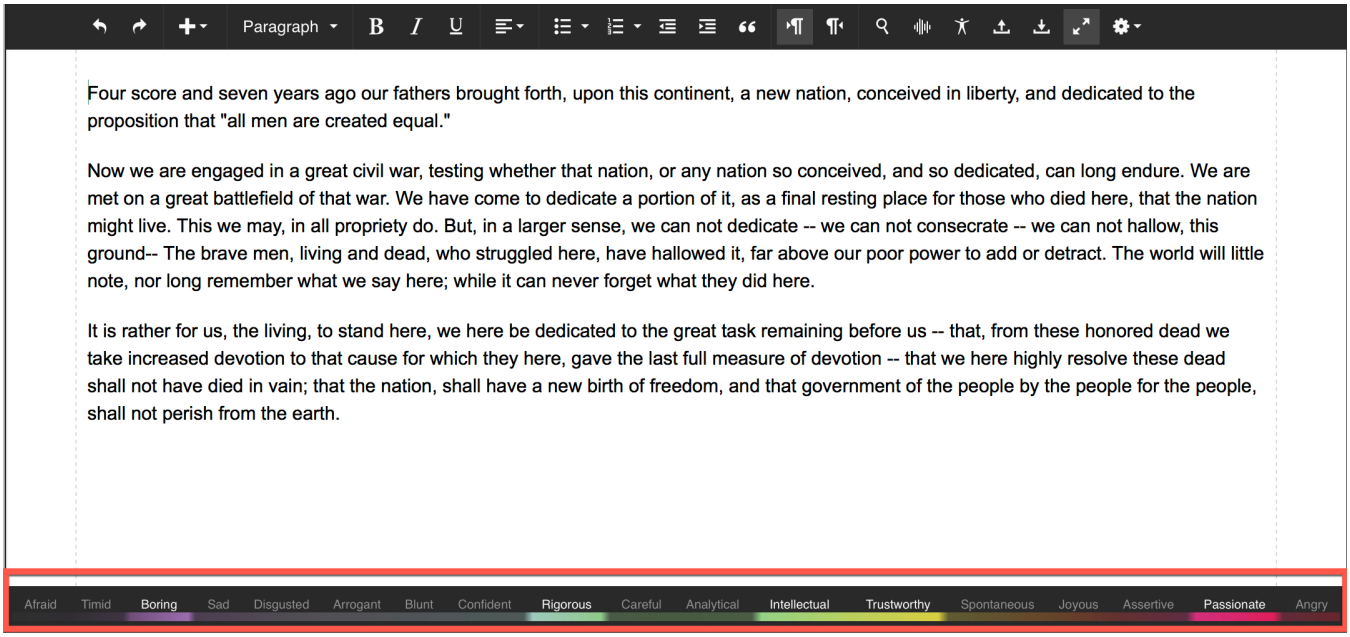
Once enabled, the Tone Analyzer service can be called by clicking on the Tone Analyzer button in the toolbar, under the "More" menu.

(click image to enlarge)



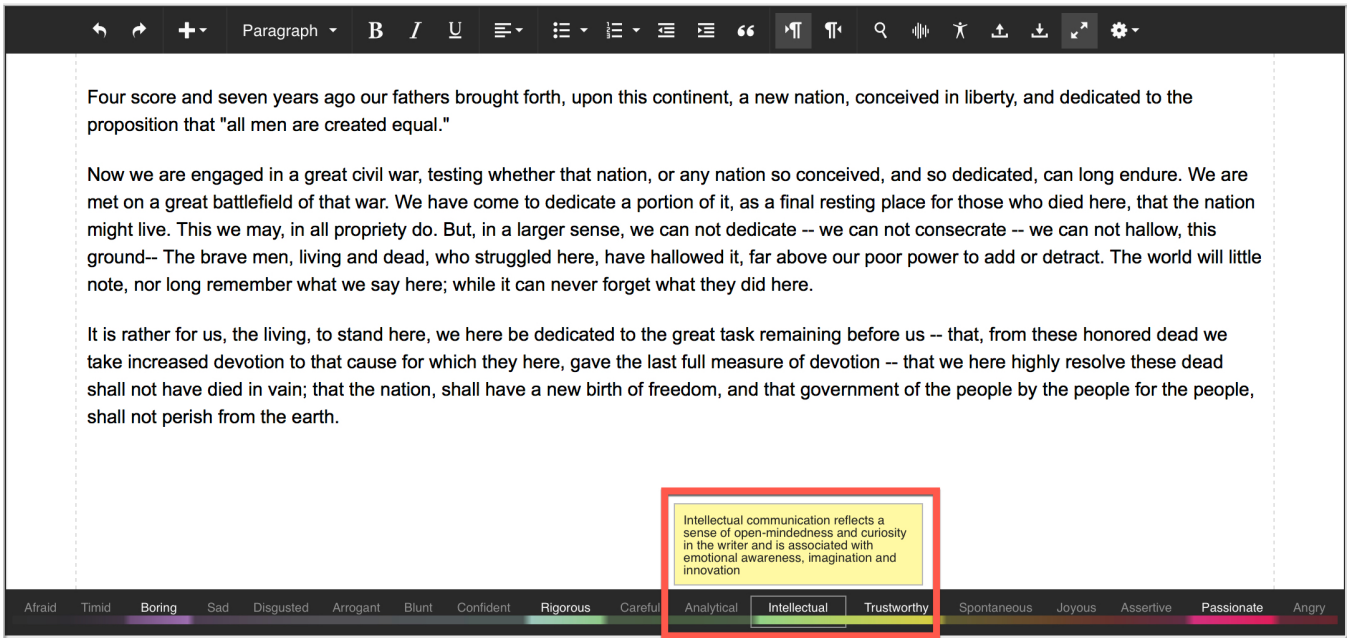
The emotional tones that are present in your content will be highlighted on the tone bar, which will appear at the bottom of your editor.

(click image to enlarge)



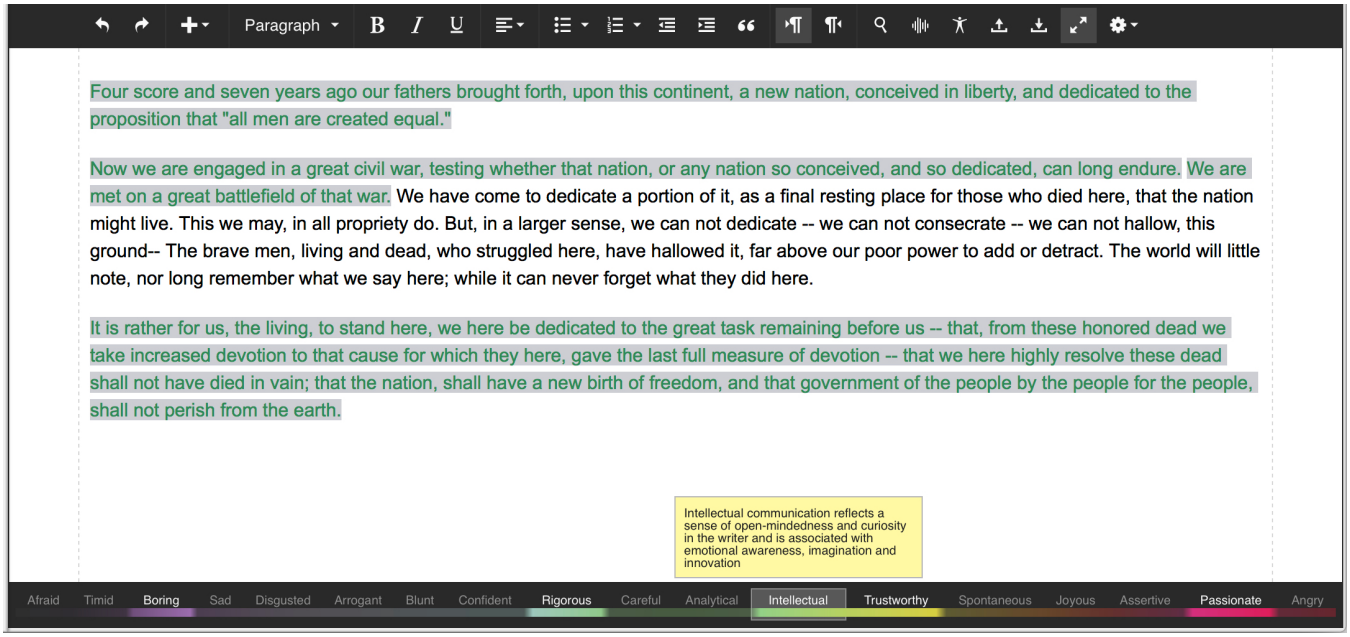
Hovering over a highlighted tone provides more information about that specific tone.

(click image to enlarge)



Clicking on a highlighted tone will highlight all the text in your content that helps to produce that particular tone.

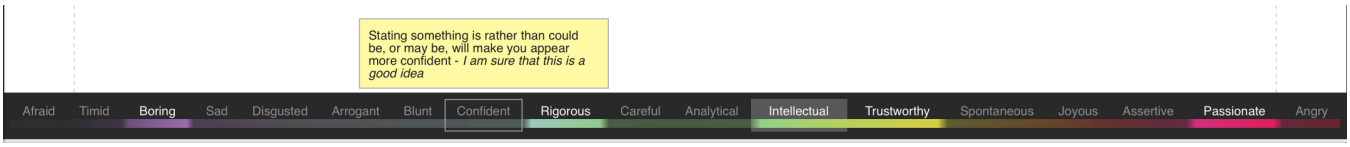
(click image to enlarge)



Clicking on the tone again removes the highlighting.

You can also click on a non-highlighted tone and receive guidance on how your content could produce that tone.

(click image to enlarge)



Once Tone Analyzer has been activated, the service will periodically rescan your text and update the tone bar as needed.

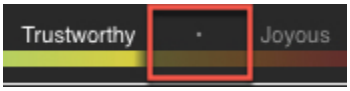
While the Tone Analyzer will work with any amount of text, ten sentences is the minimim recommended amount, and the more text you provide the better.

A note on editor width and the tone bar:

Depending on the width of your editor, the Tone Bar may not have enough room to display all of the tones.

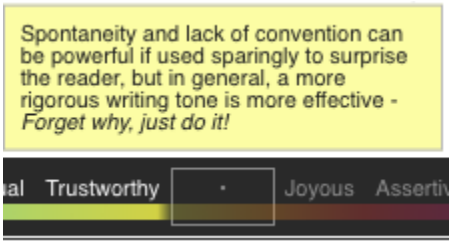
If this is the case, the Tone Bar will start with non-highlighted tones and replace them with a dot, until the tone bar can fit into the width of the editor. Here's an example of what that might look like:

(click image to enlarge)



However, even if the tone itself is replaced by a dot, you can still hover over that dot to see the tone, and its associated information.

(click image to enlarge)



Enablement Instructions

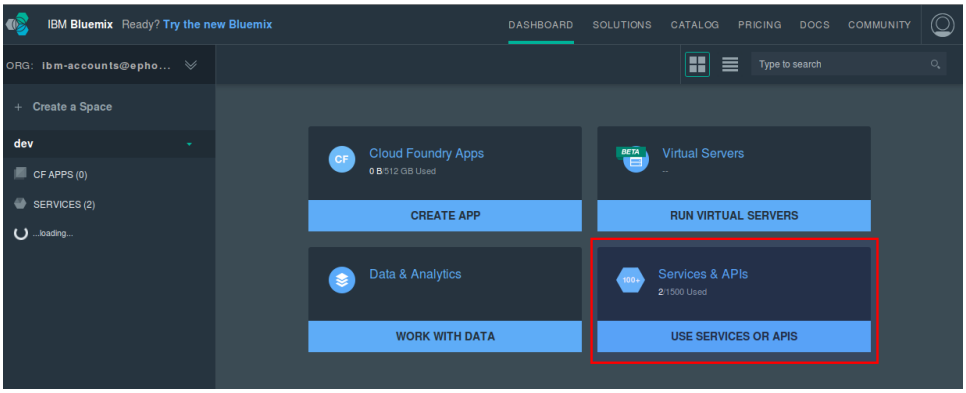
Please note:

- The below steps are specific to users only needing to set up the Tone Analyzer service
  - For all other information around installing, updating, customizing and/or configuring Textbox services within the WCM integration, start here.
- The Tone Analyzer service can be used within:
  - The authoring portlet which provides a form type UI.
  - The draft page, when choosing the option to display the entire authoring form.
- The Tone Analyzer service can NOT be used:
  - When the editor appears "in place" within the content area on a draft page

Obtain credentials for the Tone Analyzer service

In order to create credentials for the Tone Analyzer service, you must have an IBM Bluemix account.

1. Once logged in into your Bluemix account, click on the "USE SERVICES OR APIS" box within the Bluemix dashboard.



2. Inside the "Services" menu, select "Tone Analyzer" to generate the credentials.

Configure the Tone Analyzer service

The service needs additional information to run correctly, where applicable, including:

- Which service should run
- The URLs that they will be accessed from
- Proxy settings to get around firewalls
- API keys for external services
- Configuration of service behavior

Note: Allowed Origins Service

In previous versions of the Textbox.io services, you could use the `ephox.allowed.origins.url` setting to point to the Allowed Origins Service. This service has been discontinued.

Allowed Origins are now configured in the configuration file as below.
If you edit these settings after the "EphoxTbioServices" have been installed you must stop and restart the service for changes to take effect.

You can specify configuration for the Textbox.io server-side features using one of two options.

File based configuration

You can create an *application.conf* file and specify the settings as documented in the Configuring Textbox.io Services article with the following variations for Connections and WebSphere:

1. Create application.conf

On the drive/partition where WebSphere Application Server (WAS) is installed, create the text file

Operating System	Path	Notes
Linux/Unix	/opt/ephox/application.conf	
Windows	***DRIVE***:\opt\ephox\application.conf	***DRIVE*** is the drive where Websphere Application Server is installed, e.g. "C", "D" or "E" etc

2. Edit application.conf

Edit this file as shown in section "Create a configuration file" on the installing the server-side components page and Textbox.io documentation.

If you have an API key for IBM Tone Analyzer you can specify the key as follows:

Example application.conf with tone service configuration only

```
ephox {
  # Tone service configuration
  cognitive {
    tone-analyzer {
      user-name = "<user_name>"
      password = "<password>"
    }
  }
  # Note: other configurations (as in the example above) omitted here for brevity
}
```

Substitute `<user_name>` with the actual user name. Substitute `<password>` with the actual password.

JVM System properties based configuration

If the file-based mechanism is not appropriate, you can set JVM system settings for the configuration of the services.

The following steps describe the process for setting the configuration using JVM system properties.

1. Find JVM settings

Find your JVM settings according to the version of Websphere that you are running. Use this page to do so: http://www-01.ibm.com/support/docview.wss?uid=swg21417365

2. Specify system properties

Specify the domains where the editor is served from, and optionally other settings such as link caching and proxy configuration. For details, please review Server-Side Components - Installation and Setup for Textbox.io.

Example JVM system properties

```
-Dephox.allowed-origins.origins.0=http://connections
-Dhttp.proxyHost=someproxy.internal.corp
-Dhttp.proxyPort=8080
```

When specifying the URL/s for the domains that will serve the Textbox.io editor, you *may* need to specify different combinations of the *protocol*, *hostname* and *port* based on the browsers you use. For more details, please review Server-Side Components - Installation and Setup for Textbox.io.
If you need additional domains where the editor is served from, you need to specify additional allowed origins by repeating **-Dephox.allowed-origins.origins.0=domain**, replacing the “0” with an incrementing number for each domain that is added.

Example JVM system properties

```
-Dephox.allowed-origins.origins.1=http://connections.yourdomain
-Dephox.allowed-origins.origins.2=http://connections:10039
-Dephox.allowed-origins.origins.3=http://connections.yourdomain:10039
```

If you have an API key for IBM Tone Analyzer you can specify the key as follows:

Example JVM system properties for Tone Service

```
-Dephox.cognitive.tone-analyzer.user-name=<user_name>
-Dephox.cognitive.tone-analyzer.password=<password>
```

Substitute `<user_name>` with the actual user name without any quotes. Substitute `<password>` with the actual password without any quotes.

Install the Textbox.io services for IBM WCM

Follow the instructions on this page to install the Textbox.io services for IBM WCM, which include the spelling, hyperlinking, image-proxy and cognitive services.

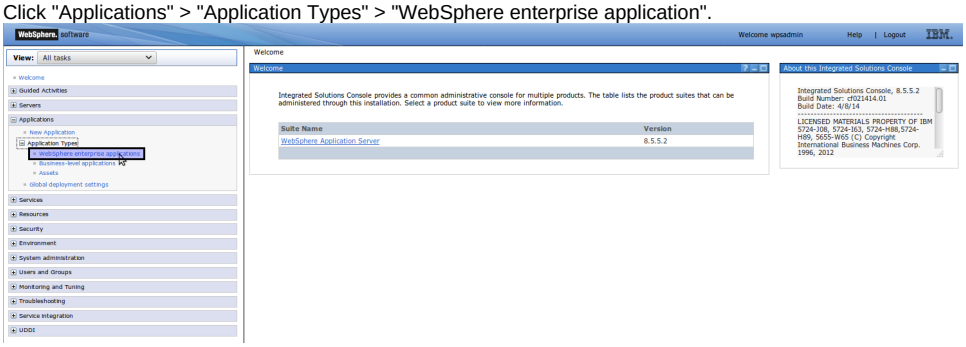
Installing the cognitive service gives you access to the Tone Analyzer.

Important

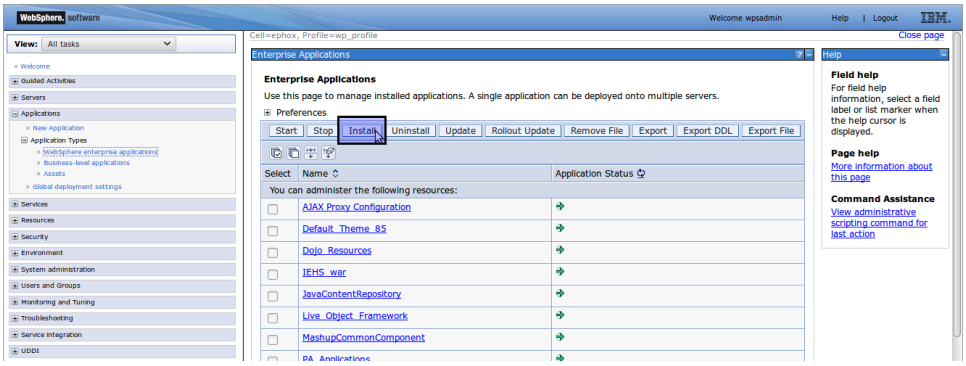
This application must be installed on the same WebSphere Application Server as WCM. Be sure to assign all modules to the correct server in the Manage Modules page.

Installation steps

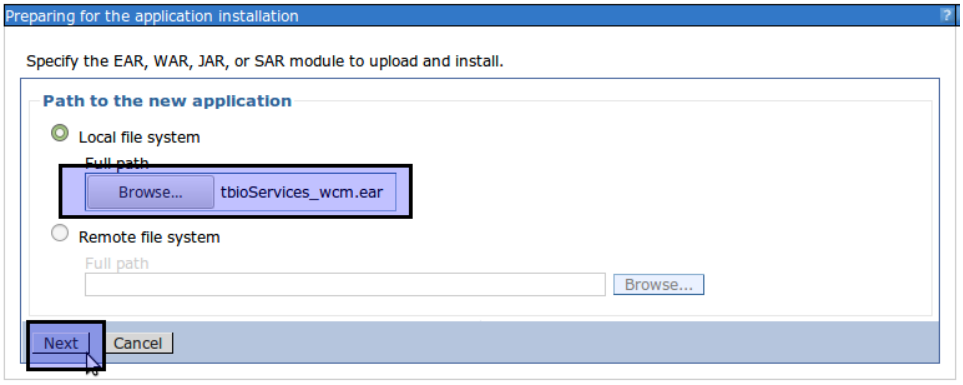
1. Log into your WebSphere Application Server Console (e.g. https://server:port/ibm/console/).
2. Click "Applications" > "Application Types" > "WebSphere enterprise application".



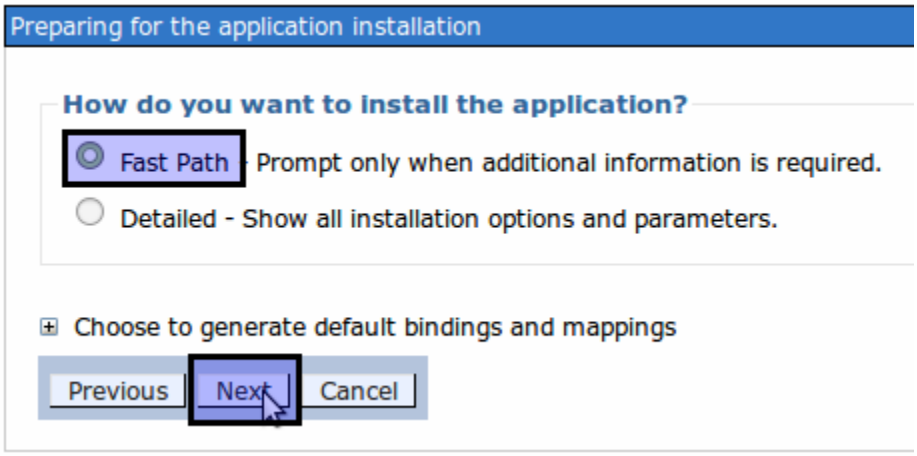
3. Click "Install".



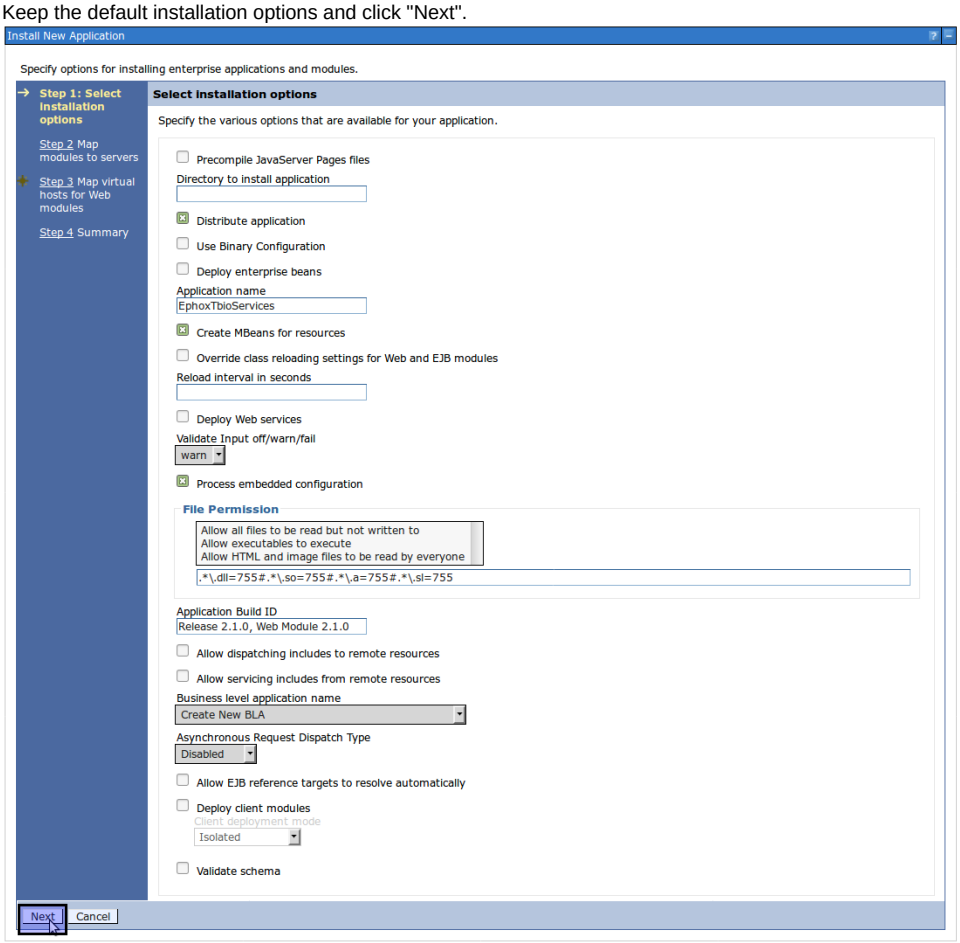
4. Provide path to "tbioServices_wcm.ear" and click "Next".



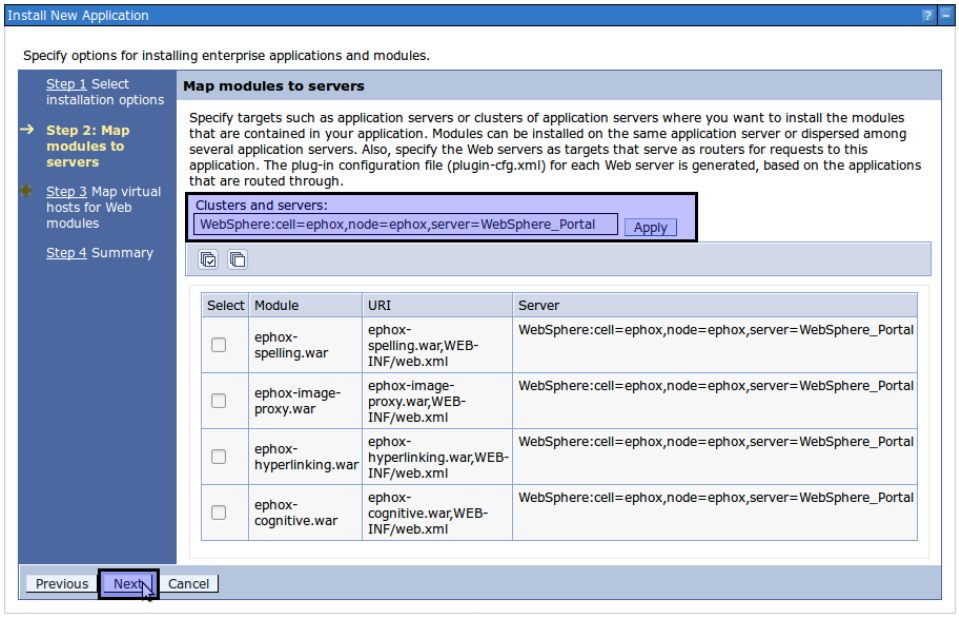
5. Select "Fast Path" and click "Next".



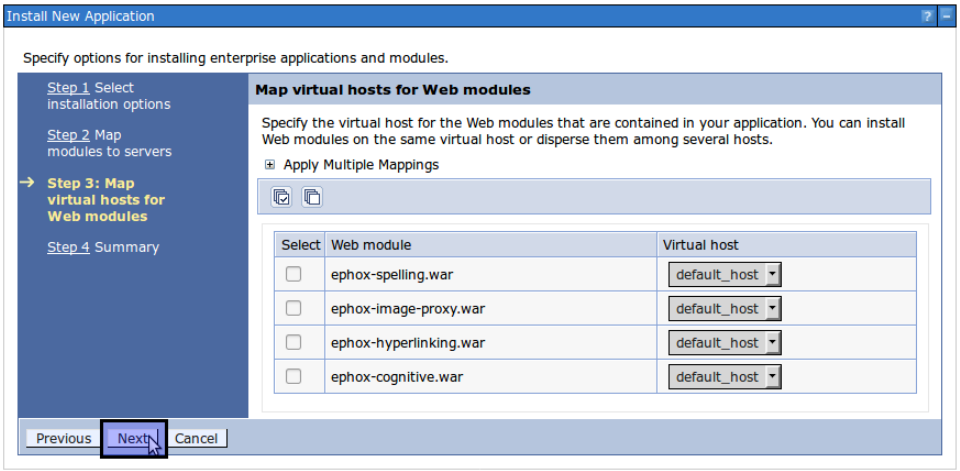
6. Keep the default installation options and click "Next".



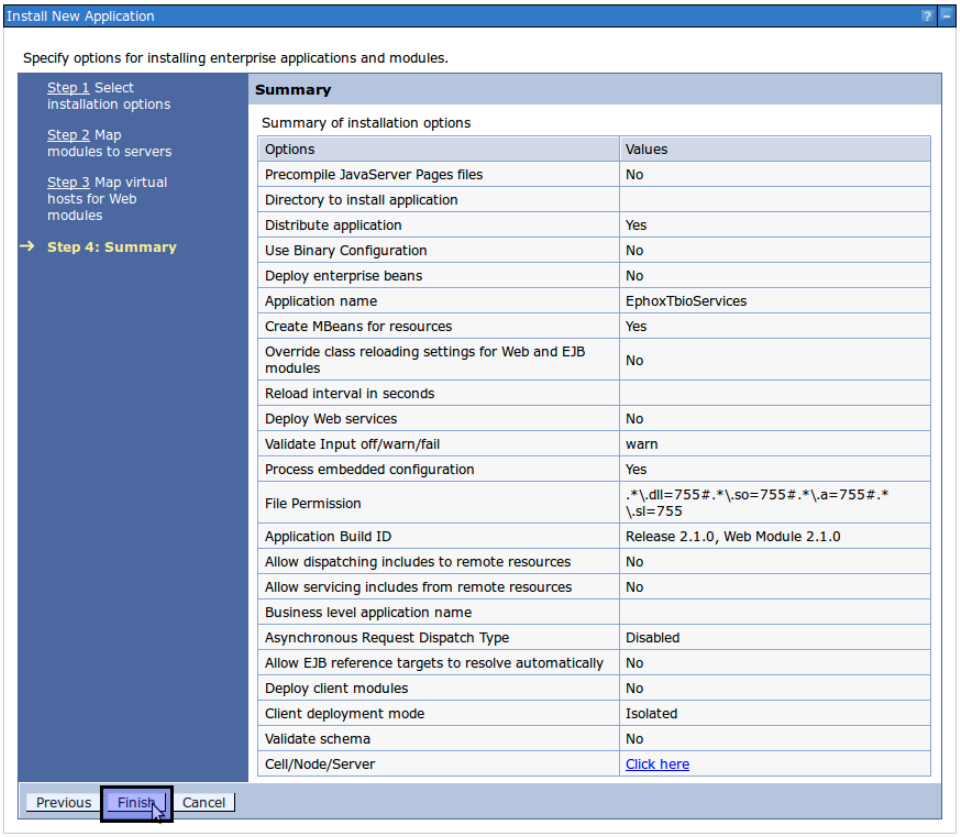
7. Make sure that the modules are deployed to the server running WebSphere Portal. Typically this will be the server called "WebSphere_Portal". If this server does not appear in the list, it may not have been started yet.
   To ensure that the modules are deployed to the right location, you will need to select the server, select the modules, and then click "Apply".
   Once the modules are mapped to the correct server, click "Next".



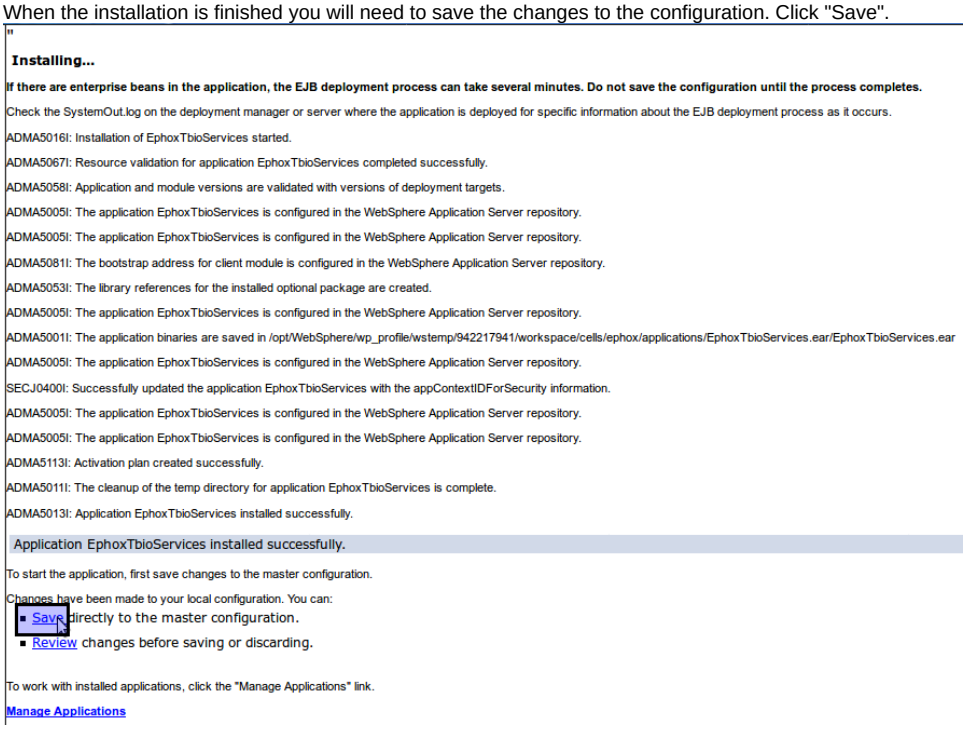
8. No action is required for mapping virtual hosts for Web modules. Click "Next".



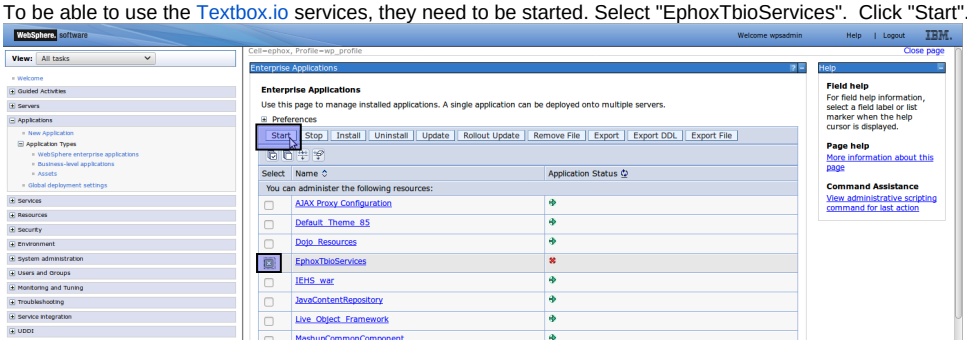
9. You will be presented with a summary of installation options. Click "Finish" to install the application.



10. When the installation is finished you will need to save the changes to the configuration. Click "Save".



11. To be able to use the Textbox.io services, they need to be started. Select "EphoxTbioServices". Click "Start".



Customize Textbox to use Tone Analyzer

Please refer to the following IBM Knowledge Center article:

https://www.ibm.com/support/knowledgecenter/SSYJ99_8.5.0/wcm/wcm_config_ephox_custom.html

Examples

Info:

For more information regarding the customization of Textbox.io, please see:

http://docs.ephox.com/display/tbio/configuration

The sample configuration provided by IBM contains the following base configuration object; the relevant cognitive / isToneEnabled? snippet is towards the bottom:

```
...
  var config = {
    css: {
      stylesheets: styleSheetUrl ? [styleSheetUrl] : []
    },
    images: {
      editing: {
        proxy: isImageProxyEnabled ? urlImageProxyService : undefined
      },
      allowLocal: !isLimitedToLibraryImagePicker
    },
    ui: {
      locale: locale,
      toolbar: {
        items: items
      }
    },
    codeview: {
      enabled: isCodeViewEnabled
    },
    spelling: {
      url: isSpellCheckingEnabled ? urlSpellingService : undefined
    },
    links: {
      validation: {
        url: isLinkValidationEnabled ? urlLinkService : undefined
      },
      embed: {
        url: isMediaEmbedEnabled ? urlLinkService : undefined
      }
    },
    cognitive: {
      tone: {
        url: isToneEnabled ? urlToneService : undefined
      }
    },
```

```
    textboxioExtensions: [
    ]
  };
...
```

This base object will be used in the examples.

Once the customizations have been applied, the configuration must be installed as per IBM's documentation.

Enabling/Disabling Tone Analyzer Service Functionality

Textbox.io makes use of a server-side service to provide its Tone Analyzer service.

This feature can be enabled/disabled in the configuration using the true/false value, which is defined directly above the config object.

Tone Analyzer is disabled by default. To enable it, set isToneEnabled to true.

```
// Disable/Enable services
 var isSpellCheckingEnabled = true;
 var isImageProxyEnabled = true;
 var isLinkValidationEnabled = false;
 var isMediaEmbedEnabled = false;
 var isToneEnabled = false;
```

Changing the value to true/false only enables/disabled the service; it does not turn the actual service on/off.

To turn the desired service on or off, please see the instructions in the service documentation.

System requirements

Please contact IBM if you require support.

Browsers

For a list of supported browsers please refer to http://docs.ephox.com/display/tbio/System+Requirements.

Updating Textbox.io for WCM

Please contact IBM if you require support.

The Textbox.io for WCM integration is installed by default with IBM WCM as the "EphoxTextboxio" and "EphoxTbioServices" applications in WebSphere.

New installs of IBM WCM may want to keep the IBM supplied integration and only do steps 2-4 and 7.

If you have been supplied a new version of the integration then work through all the steps.

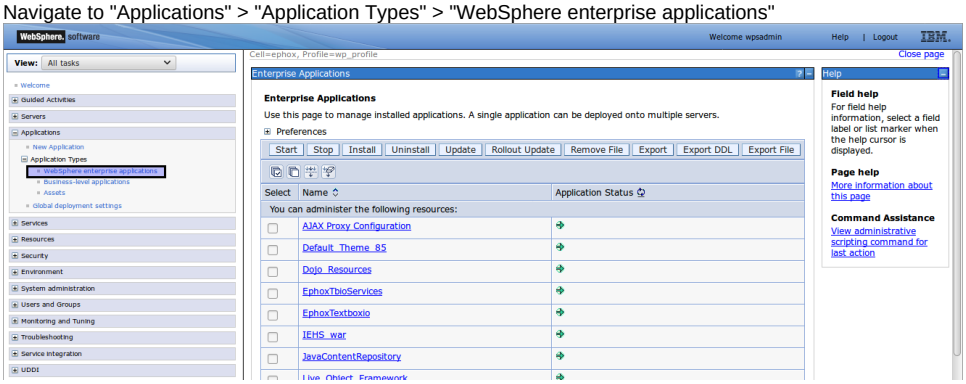
1. Remove old integration

Please contact IBM if you require support.

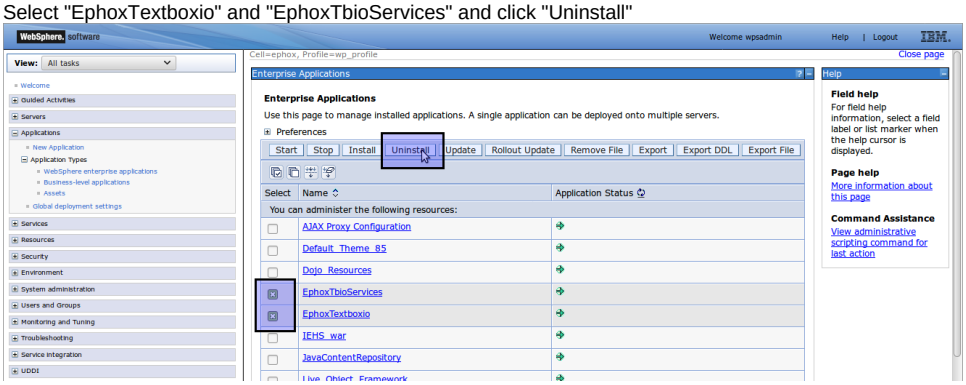
Before you update the integration, please follow these steps to remove the current version.

Uninstallation steps

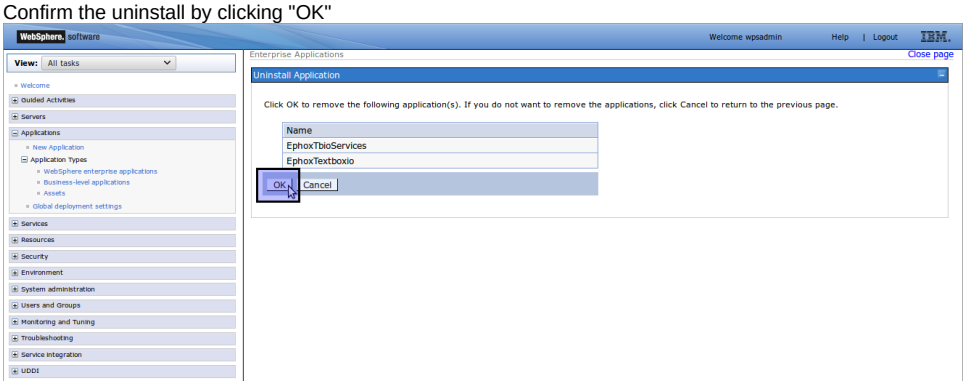
1. Log into WebSphere Integrated Solutions Console
2. Navigate to "Applications" > "Application Types" > "WebSphere enterprise applications"



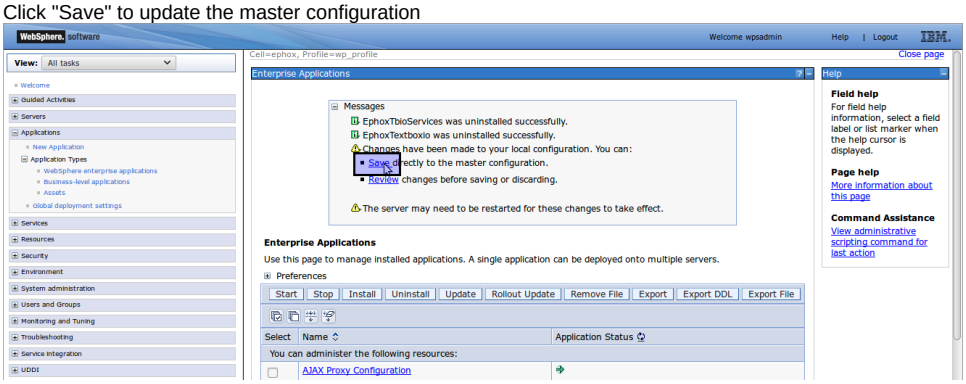
3. Select "EphoxTextboxio" and "EphoxTbioServices" and click "Uninstall"



4. Confirm the uninstall by clicking "OK"



5. Click "Save" to update the master configuration



2. Get third-party API credentials

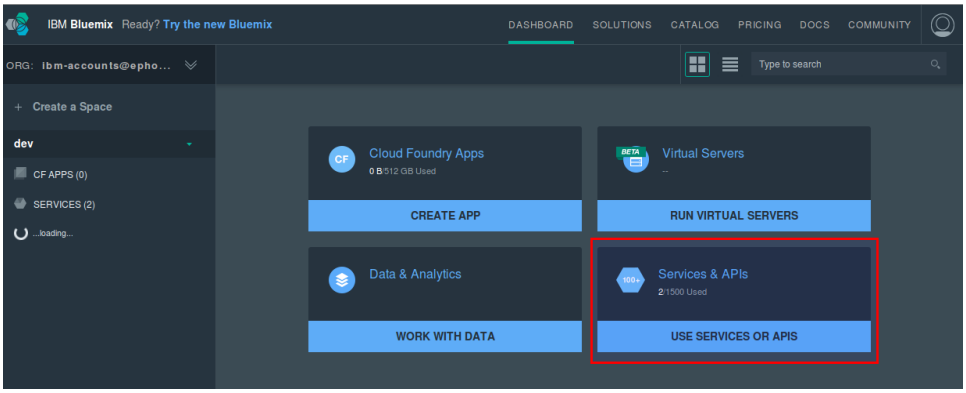
Please contact IBM if you require support.

Some optional features of the integration require API keys from third parties like IBM Watson Tone Analysis.

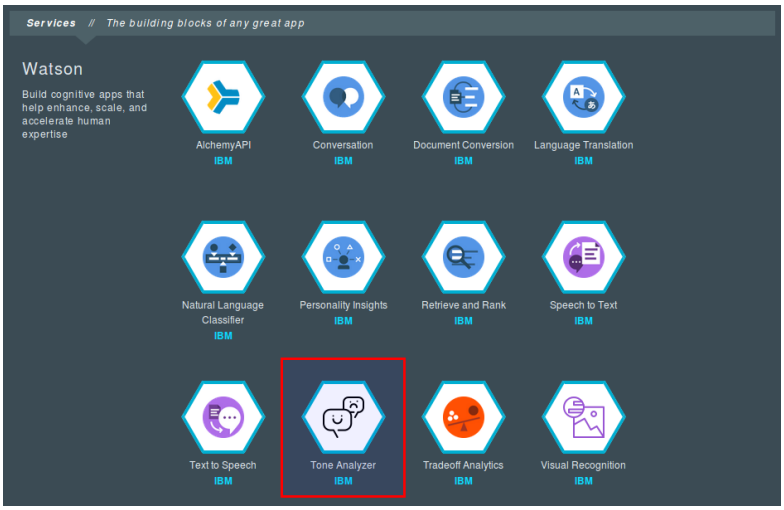
Obtaining credentials for the Tone service

In order to create credentials for the Tone service, you must have an IBM Bluemix account.

1. Once logged in into your Bluemix account, click on the "USE SERVICES OR APIS" box within the Bluemix dashboard.



2. Inside the "Services" menu, select "Tone Analyzer" to generate the credentials.



3. Configuring services

Please contact IBM if you require support.

The services need additional information to run correctly, including:

- Which services should run
- The URLs that they will be accessed from
- Proxy settings to get around firewalls
- API keys for external services
- Configuration of service behavior

Note: Allowed Origins Service

In previous versions of the Textbox.io services, you could use the `ephox.allowed.origins.url` setting to point to the Allowed Origins Service. This service has been discontinued. Allowed Origins are now configured in the configuration file as below.
If you edit these settings after the "EphoxTbioServices" have been installed you must stop and restart the service for changes to take effect.

You can specify configuration for the Textbox.io server-side features using one of two options.

File based configuration

You can create an *application.conf* file and specify the settings as documented in the Configuring Textbox.io Services article with the following variations for Connections and WebSphere:

1. Create application.conf

On the drive/partition where WebSphere Application Server (WAS) is installed, create the text file

Operating System	Path	Notes
Linux/Unix	/opt/ephox/application.conf	
Windows	***DRIVE***:\opt\ephox\application.conf	***DRIVE*** is the drive where Websphere Application Server is installed, e.g. "C", "D" or "E" etc

2. Edit application.conf

Edit this file as shown in section "Create a configuration file" on the installing the server-side components page and the configure the enhanced media embed service page in the Textbox.io documentation. The link-checking and enhanced media embed features are disabled by default in WCM. Enable them explicitly in the *application.conf* file if required. The Installation and Setup page details how to enable link checking and the Configure Enhanced Media Embed Server page details how to enable Enhanced Media Embed.

In the example below we have chosen to enable the link checking and enhanced media embed. The embed service relies on oEmbed endpoints and we have choosen to only include the free Youtube oEmbed endpoint as an example but normally you would list many more or use an Iframely account for the best experience.

Example application.conf

```
ephox {
  allowed-origins {
    origins = [
      "http://connections:9081",
      "https://connections:9444",

      "http://connections",
      "https://connections"
    ]
  }
  proxy {
    http.proxyHost = someproxy.internal.corp
    http.proxyPort = 8080
  }
  link-checking {
    enabled = true
  }
  embed {
    enabled = true
    custom = [
      # youtube
      {
```

```
        endpoint = "http://www.youtube.com/oembed"
        schemes = [
          "http://youtu.be/*",
          "https://youtu.be/*",
          "http://www.youtu.be/*",
          "https://www.youtu.be/*",
          "http://youtube.com/*",
          "https://youtube.com/*",
          "http://www.youtube.com/*",
          "https://www.youtube.com/*",
          "http://m.youtube.com/*",
          "https://m.youtube.com/*"
        ]
      }
          # Note: many other endpoints omitted here for brevity
    ]
  }
}
```

If you have an API key for IBM Tone Analyzer you can specify the key as follows:

Example application.conf with tone service configuration only

```
ephox {
  # Tone service configuration
  cognitive {
    tone-analyzer {
      user-name = "<user_name>"
      password = "<password>"
    }
  }
  # Note: other configurations (as in the example above) omitted here for brevity
}
```

Substitute `<user_name>` with the actual user name. Substitute `<password>` with the actual password.

JVM System properties based configuration

If the file-based mechanism is not appropriate, you can set JVM system settings for the configuration of the services.

Note that while it is possible to configure everything using system properties it is not recommended if you wish to use the Enhanced Media Embed service with free oEmbed endpoints as there are many configuration values required for a good experience.

The following steps describe the process for setting the configuration using JVM system properties.

1. Find JVM settings

Find your JVM settings according to the version of Websphere that you are running. Use this page to do so: http://www-01.ibm.com/support/docview.wss?uid=swg21417365

2. Specify system properties

Specify the domains where the editor is served from, and optionally other settings such as link caching and proxy configuration. For details, please review Server-Side Components - Installation and Setup for Textbox.io.

Example JVM system properties

```
-Dephox.allowed-origins.origins.0=http://connections
-Dhttp.proxyHost=someproxy.internal.corp
-Dhttp.proxyPort=8080
```

When specifying the URL/s for the domains that will serve the Textbox.io editor, you *may* need to specify different combinations of the *protocol*, *hostname* and *port* based on the browsers you use. For more details, please review Server-Side Components - Installation and Setup for Textbox.io.
If you need additional domains where the editor is served from, you need to specify additional allowed origins by repeating **-Dephox.allowed-origins.origins.0=domain**, replacing the “0” with an incrementing number for each domain that is added.

Example JVM system properties

```
-Dephox.allowed-origins.origins.1=http://connections.yourdomain
-Dephox.allowed-origins.origins.2=http://connections:10039
-Dephox.allowed-origins.origins.3=http://connections.yourdomain:10039
```

If you have an API key for IBM Tone Analyzer you can specify the key as follows:

Example JVM system properties for Tone Service

```
-Dephox.cognitive.tone-analyzer.user-name=<user_name>
-Dephox.cognitive.tone-analyzer.password=<password>
```

Substitute `<user_name>` with the actual user name without any quotes. Substitute `<password>` with the actual password without any quotes.

4. Import Certificates for External Servers

Please contact IBM if you require support.

By default, WebSphere only trusts connections to itself. This poses a problem since the link validation service, image proxy service and enhanced media embed service require connections to the external servers with which they must interact.

Therefore, you must ensure WebSphere has the proper SSL configuration - including certificate authority root certificates for verifying SSL connections in its trust store - for all potential targets of secure (SSL) connections. Targets usually include in-house servers potentially secured by an in-house certificate authority as well as servers on the public Internet secured by public certificate authorities.

SSL security can be configured in the WebSphere console under:

> *Security > SSL certificate and key management*

Textbox.io Services will honor dynamic outbound endpoint SSL configurations based on hostname and port number, if configured.

Suggestion

One way of creating a simple, adequate WebSphere SSL configuration would be to create a new keystore entry under "SSL certificate and key management > Key stores and certificates" and point the path to the trust store of WebSphere's JVM, usually something like `/opt/IBM/WebSphere/AppServer/java/8.0/jre/lib/security/cacerts`, with password "changeit", and the type to "JKS".

Root certificates of in-house certificate authorities could then be added to this trust store, and the trust store then selected as the trust store of the default SSL configuration. No extra WebSphere related Textbox.io configuration settings are needed in this case.

Advanced Textbox.io SSL configuration

Textbox.io Services can be configured to use a particular named SSL configuration, instead of using the default WebSphere configuration, by setting **`ephox.http.websphere.ssl-config-name`** to the name of the SSL configuration in WebSphere.

E.g. after creating an SSL configuration named "TbioServices", configure **`ephox.http.websphere.ssl-config-name`**`=TbioServices`

All of WebSphere's SSL security configuration - including trust stores and certificates but also protocol, cipher settings etc. - can optionally be bypassed and the JVM's SSL configuration - including the JVM trust store with its certificates - be used instead by configuring **ephox.http.websphere.use-ssl-config**=false

Simple setup for testing or pre-production environments

For use in evaluation or pre-production environments, all SSL security can be bypassed by both configuring **`ephox.http.websphere.use-ssl-config`**`=false` and **`ephox.http.trust-all-cert`**`=true`.

Bypassing all SSL security is not recommended for production environments.

5. Installing services

Please contact IBM if you require support.

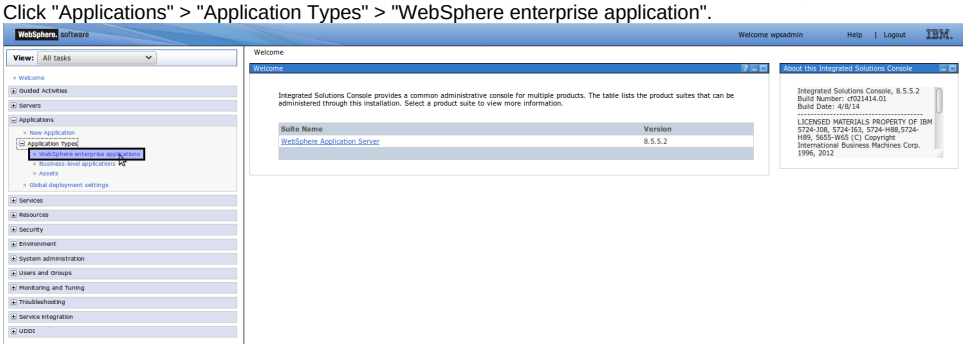
Follow the instructions on this page to install the Textbox.io services for IBM WCM, which include the spelling, hyperlinking, image-proxy and cognitive services, which gives you access to the Tone Analyzer.

Important

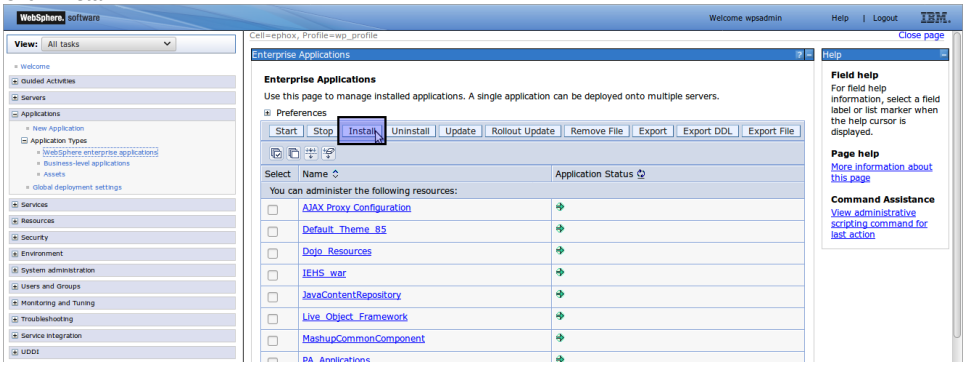
This application must be installed on the same WebSphere Application Server as WCM. Be sure to assign all modules to the correct server in the Manage Modules page.

Installation steps

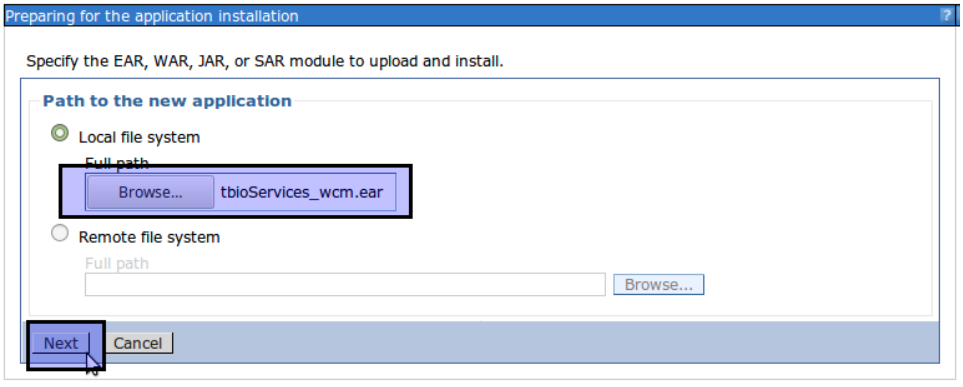
1. Log into your WebSphere Application Server Console (e.g. https://server:port/ibm/console/).
2. Click "Applications" > "Application Types" > "WebSphere enterprise application".



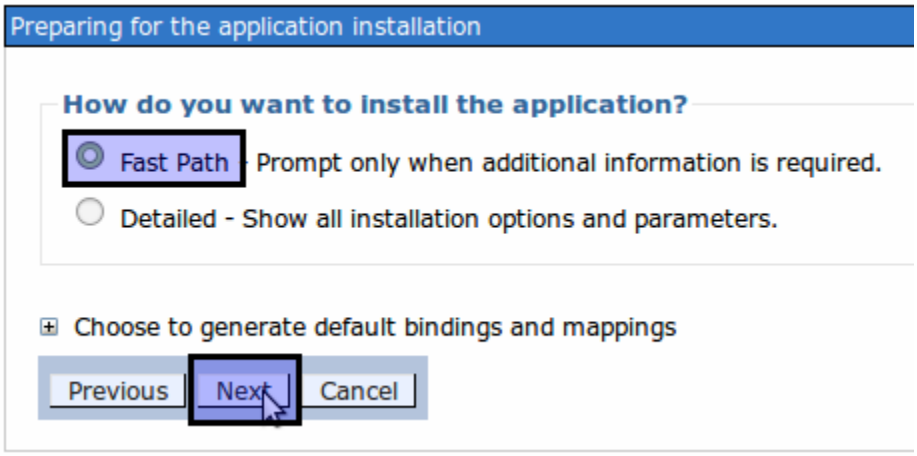
3. Click "Install".



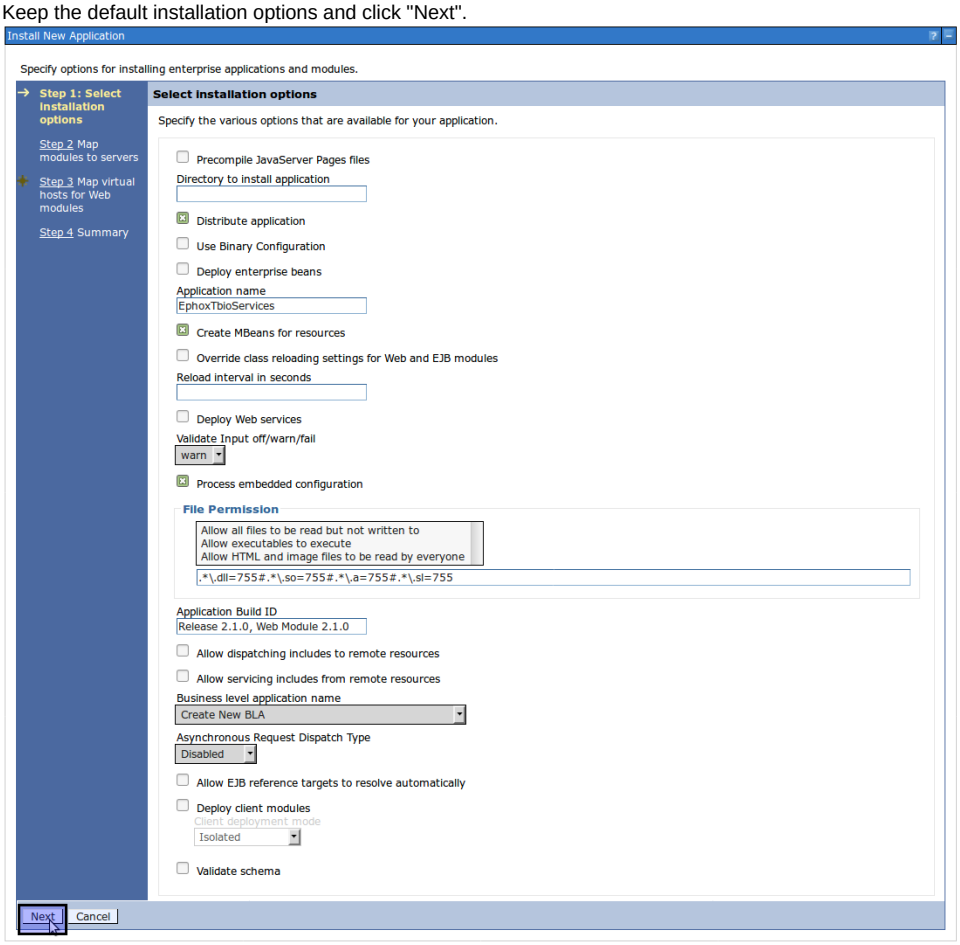
4. Provide path to "tbioServices_wcm.ear" and click "Next".



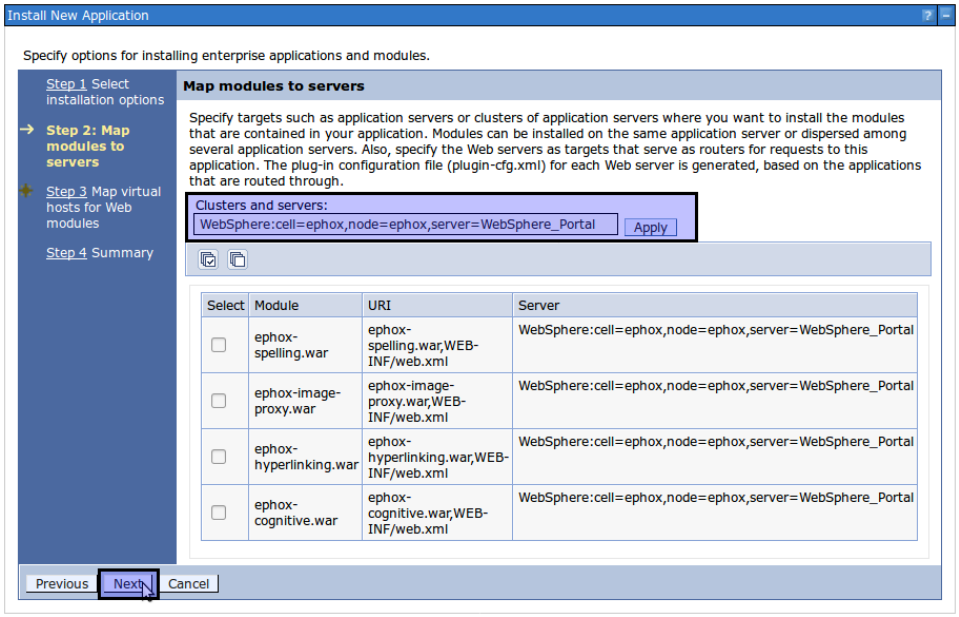
5. Select "Fast Path" and click "Next".



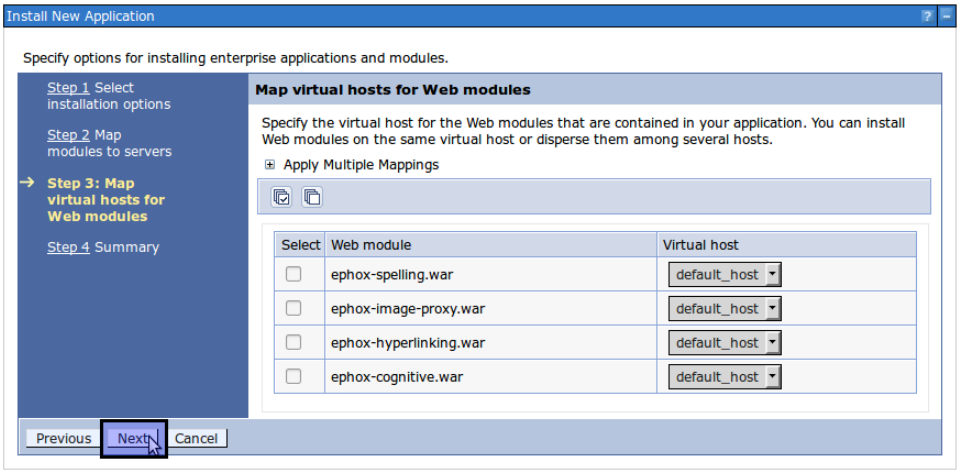
6. Keep the default installation options and click "Next".



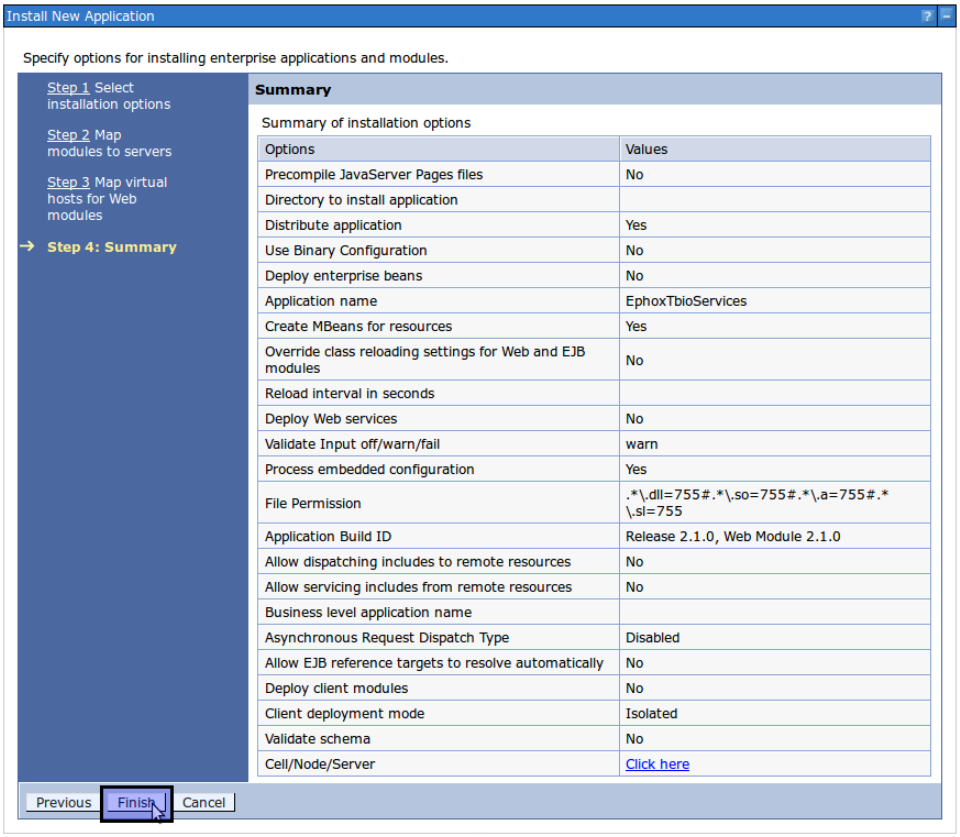
7. Make sure that the modules are deployed to the server running WebSphere Portal. Typically this will be the server called "WebSphere_Portal". If this server does not appear in the list, it may not have been started yet.
   To ensure that the modules are deployed to the right location, you will need to select the server, select the modules, and then click "Apply".
   Once the modules are mapped to the correct server, click "Next".



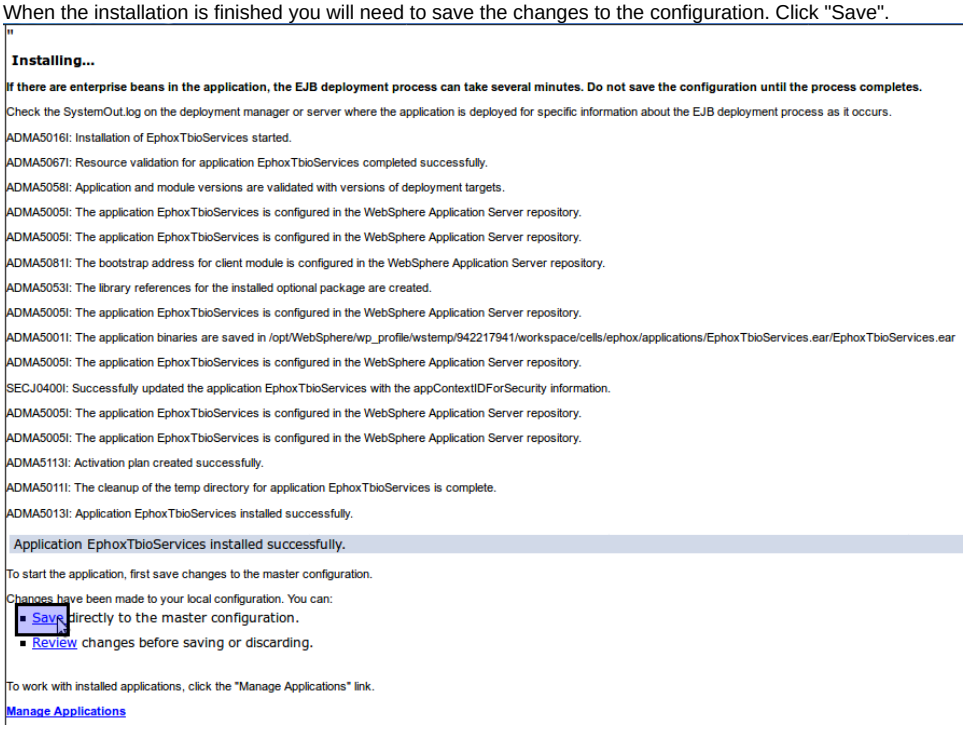
8. No action is required for mapping virtual hosts for Web modules. Click "Next".



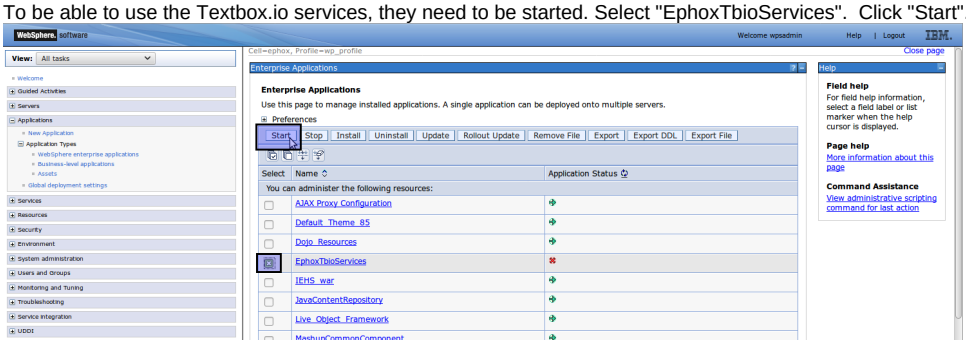
9. You will be presented with a summary of installation options. Click "Finish" to install the application.



10. When the installation is finished you will need to save the changes to the configuration. Click "Save".



11. To be able to use the Textbox.io services, they need to be started. Select "EphoxTbioServices". Click "Start".



6. Installing Textbox.io

Please contact IBM if you require support.

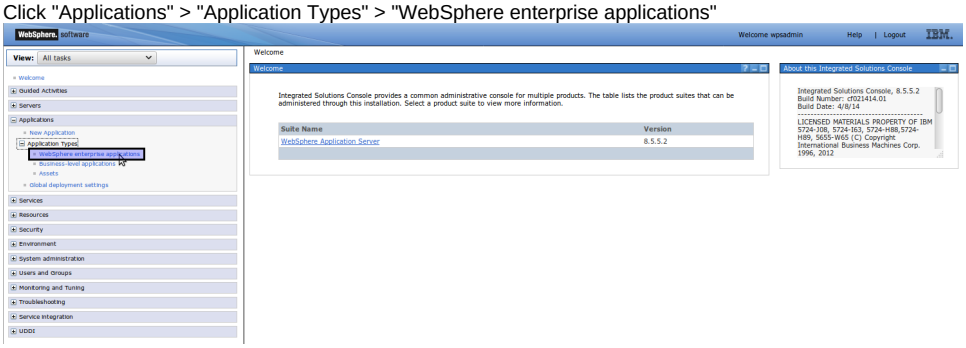
Follow the instructions on this page to install Textbox.io for IBM WCM.

Important

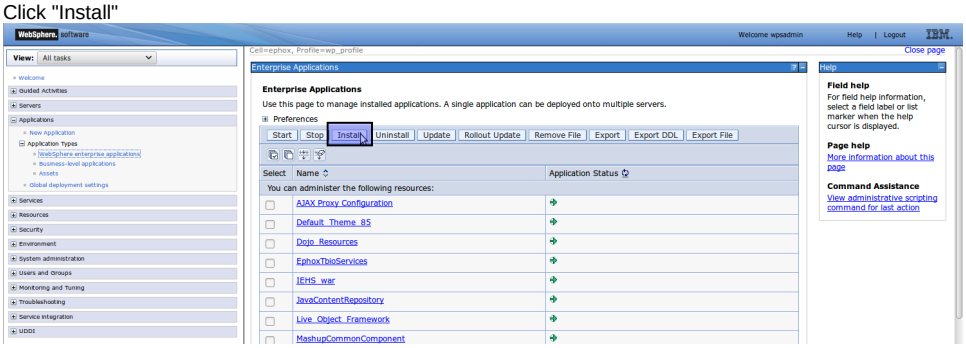
This application must be installed on the same WebSphere Application Server as WCM. Be sure to assign all modules to the correct server in the Manage Modules page.

Installation steps

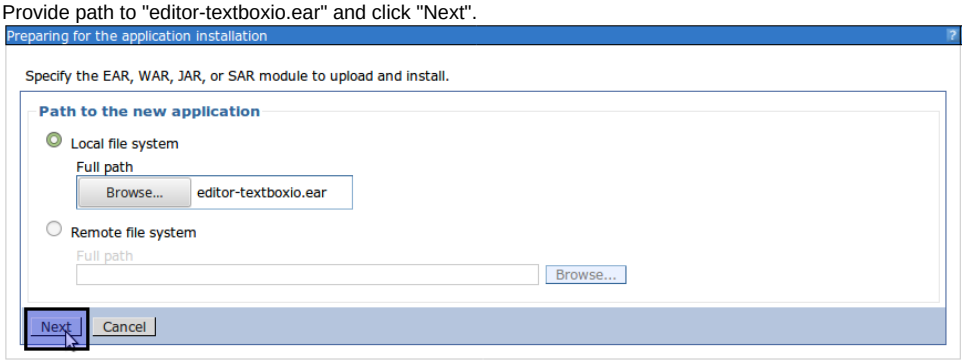
1. Log into your WebSphere Application Server Console (e.g. https://server:port/ibm/console/).
2. Click "Applications" > "Application Types" > "WebSphere enterprise applications"



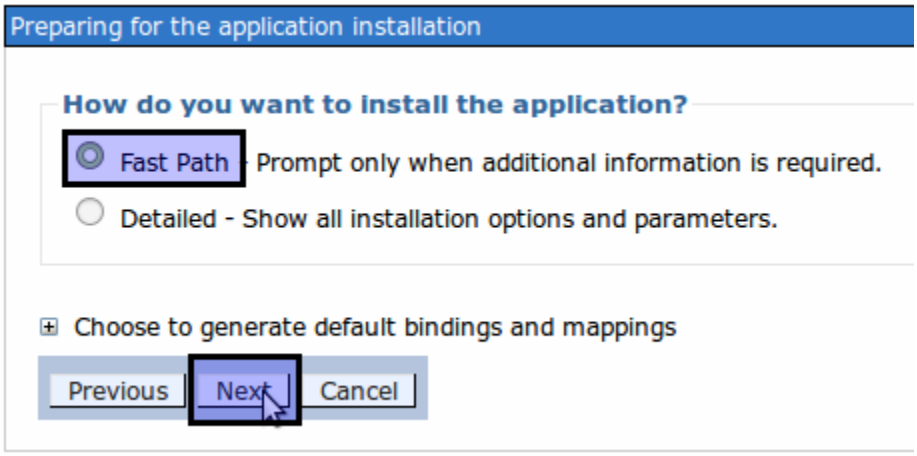
3. Click "Install"



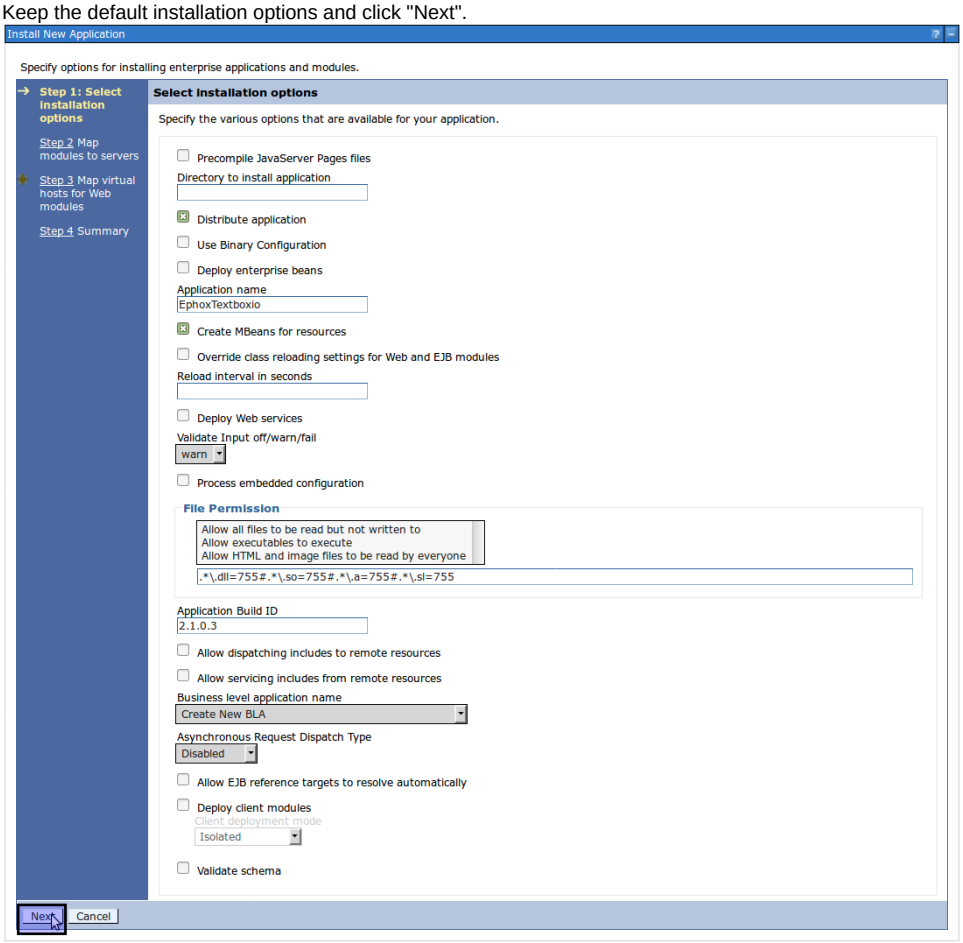
4. Provide path to "editor-textboxio.ear" and click "Next".



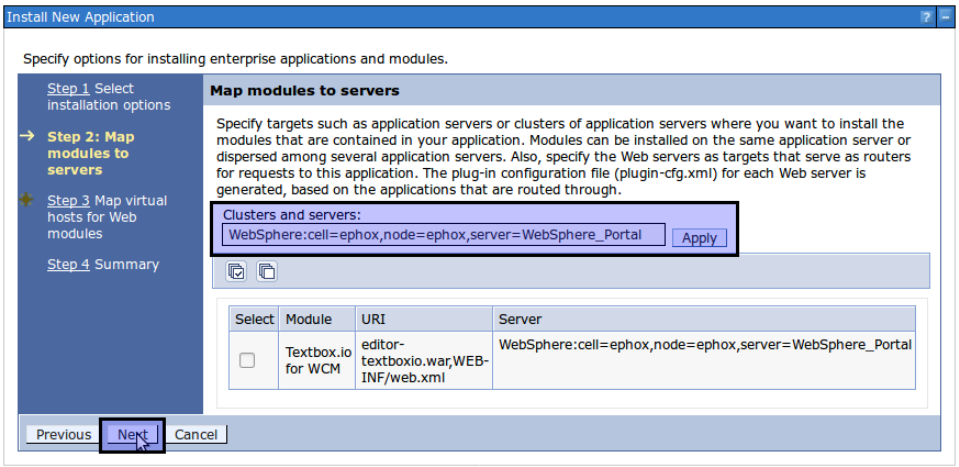
5. Select "Fast Path" and click "Next".



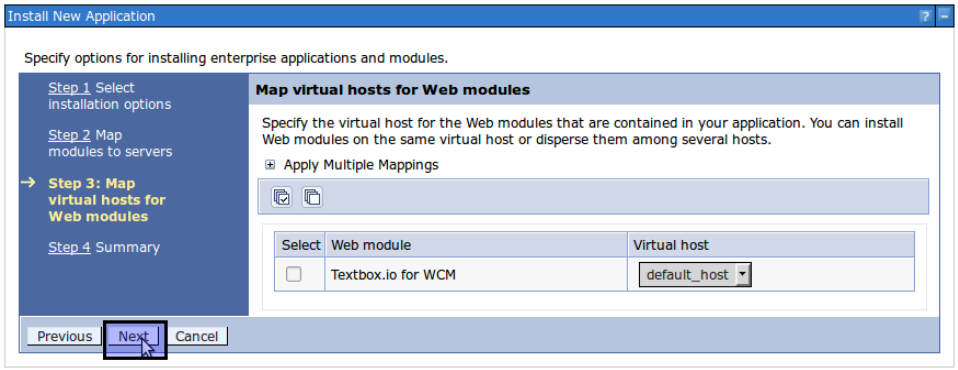
6. Keep the default installation options and click "Next".



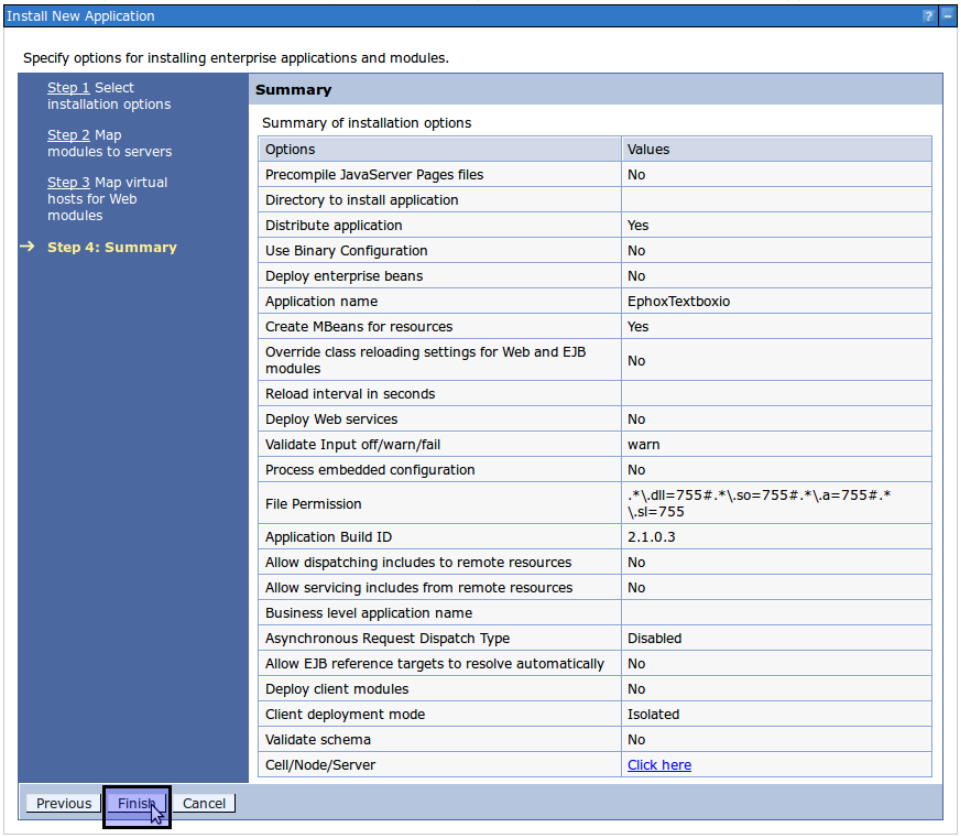
7. Make sure that the modules are deployed to the server running WebSphere Portal. Typically this will be the server called "WebSphere_Portal". If this server does not appear in the list, it may not have been started yet.
   To ensure that the ear is deployed to the right location, you will need to select the server, select the module, and then click "Apply".
   Once the module is mapped to the correct server, click "Next".



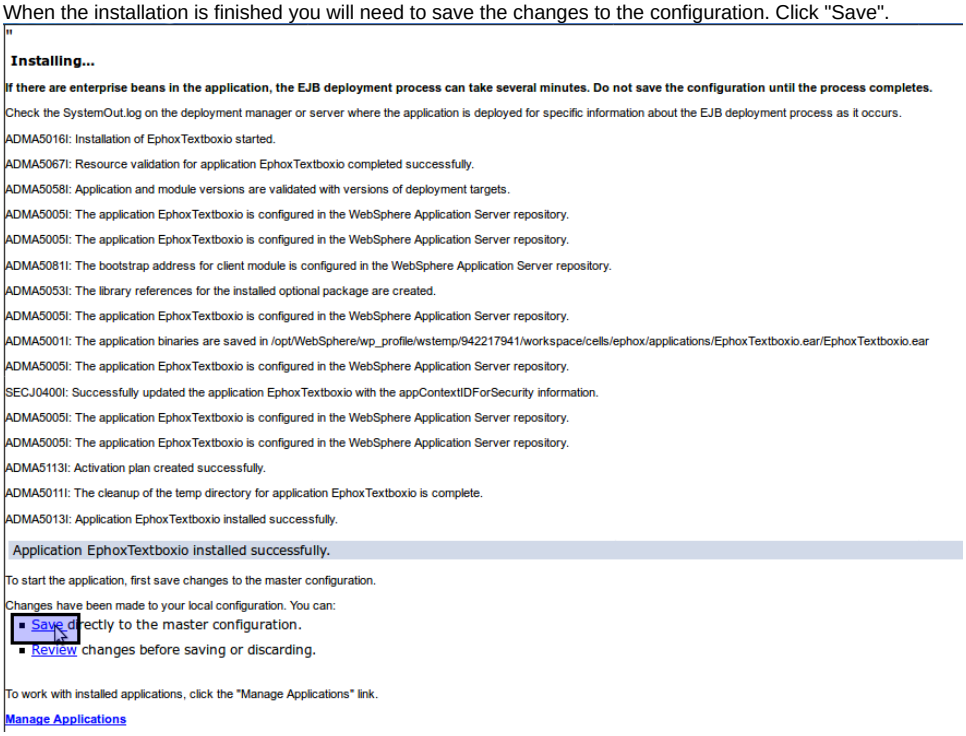
8. No action is required for mapping virtual hosts for Web modules. Click "Next".



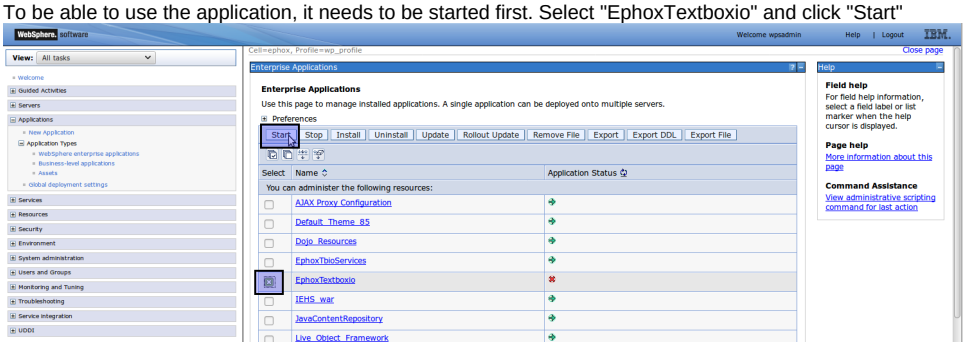
9. You will be presented with a summary of installation options. Click "Finish" to install the application.



10. When the installation is finished you will need to save the changes to the configuration. Click "Save".



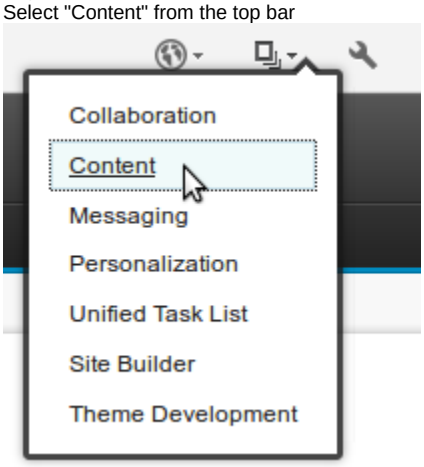
11. To be able to use the application, it needs to be started first. Select "EphoxTextboxio" and click "Start"



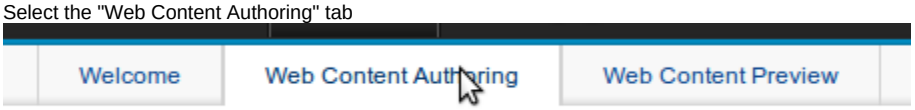
7. Enabling advanced editor

After installing textbox.io check that the editor settings are configured to use the Advanced Editor.

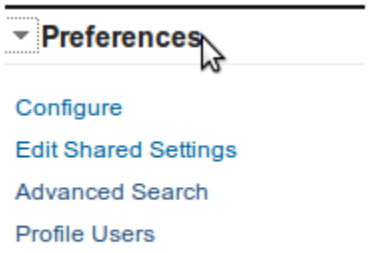
1. Log into IBM WCM with an administrator account
2. Select "Content" from the top bar



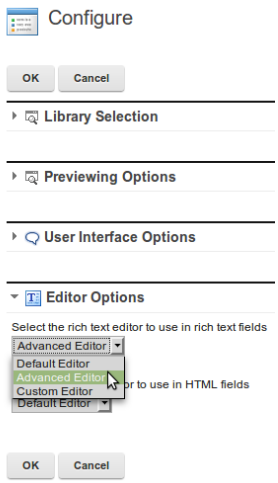
3. Select the "Web Content Authoring" tab



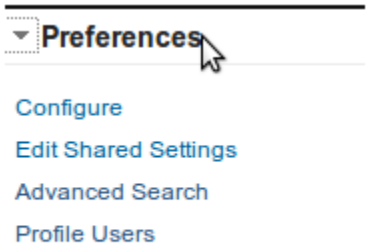
4. Open the "Preferences" section on the left side of the page and Click "Configure"



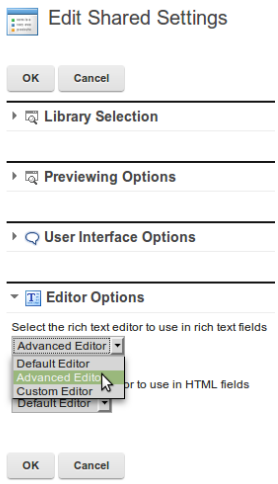
5. Click "Editor Options" to expand the section and ensure the "rich text editor" is set to "Advanced Editor". Click "OK" to save and return to the previous screen



6. Open the "Preferences" section on the left side of the page and Click "Edit Shared Settings"



7. Click "Editor Options" to expand the section and ensure the "rich text editor" is set to "Advanced Editor". Click "OK" to save



Customising Textbox.io

Please contact IBM if you require support.

Customization Steps

Please refer to the following IBM Knowledge Center article:

https://www.ibm.com/support/knowledgecenter/SSYJ99_8.5.0/wcm/wcm_config_ephox_custom.html

Examples

Info:

For more information regarding the customization of Textbox.io, please see:

http://docs.ephox.com/display/tbio/configuration

Caution:

Many items in the default configuration have WCM specific values. Altering these values may have a detrimental impact on your editing experience.

Ephox advises that changes only be made to the following items:

- paste
- ui
- `isSpellCheckingEnabled`
- `isImageProxyEnabled`
- `isLinkValidationEnabled`
- `isMediaEmbedEnabled`
- `isToneEnabled`

The sample configuration provided by IBM contains the following base configuration object:

```
...
  var config = {
    css: {
      stylesheets: styleSheetUrl ? [styleSheetUrl] : []
    },
    images: {
      editing: {
        proxy: isImageProxyEnabled ? urlImageProxyService : undefined
      },
      allowLocal: !isLimitedToLibraryImagePicker
    },
    ui: {
      locale: locale,
      toolbar: {
        items: items
      }
    },
    codeview: {
      enabled: isCodeViewEnabled
    },
    spelling: {
      url: isSpellCheckingEnabled ? urlSpellingService : undefined
    },
    links: {
      validation: {
        url: isLinkValidationEnabled ? urlLinkService : undefined
      },
      embed: {
        url: isMediaEmbedEnabled ? urlLinkService : undefined
      }
    },
    cognitive: {
```

```
      tone: {
        url: isToneEnabled ? urlToneService : undefined
      }
    },
    textboxioExtensions: [
    ]
  };
...
```

This base object will be used in the examples.

Once the customizations have been applied, the configuration must be installed as per IBM's documentation.

Enabling/Disabling Service Functionality

Textbox.io makes use of various server-side services to provide spell checking, proxying of images for editing, validation of links, generating media embeds from links, and tone analysis.

These features can be enabled/disabled in the configuration using the following true/false values, which are defined directly above the config object:

```
// Disable/Enable services
 var isSpellCheckingEnabled = true;
 var isImageProxyEnabled = true;
 var isLinkValidationEnabled = false;
 var isMediaEmbedEnabled = false;
 var isToneEnabled = false;
```

To enable a feature set the value to true, to disable set the value to false.

Note that changing the value to true/false only enables/disabled the service; it does not turn the actual service on/off.

To turn the desired service on or off, please see the instructions in the service documentation.

By default the spell checking and image proxy are enabled but the link validation, media embedding and tone analysis need to be enabled. This is because the link validation, media embedding and tone analysis will need additional configuration of the Ephox services to allow access to websites using SSL and to work correctly with proxy servers. Additionally the tone analysis service will need an IBM Watson tone analysis service key.

Modifying Paste Functionality

Textbox.io provides multiple different ways to handle the pasting of content, which can be changed modifying the value of the "paste" configuration property. The different behaviors are explained here: http://docs.ephox.com/display/tbio/paste

The default value for the "paste" configuration is "retain". If plain-text pasting is preferred you need to add the "paste" configuration property and set the value to "plain". Like this:

```
    ,
    paste: {
          style: 'plain'
        }
```

So the following configuration would be specified:

```
...
  var config = {
    css: {
      stylesheets: styleSheetUrl ? [styleSheetUrl] : []
    },
    images: {
      editing: {
        proxy: isImageProxyEnabled ? urlImageProxyService : undefined
      },
      allowLocal: !isLimitedToLibraryImagePicker
    },
    ui: {
      locale: locale,
      toolbar: {
```

```
        items: items
      }
    },
    codeview: {
      enabled: isCodeViewEnabled
    },
    spelling: {
      url: isSpellCheckingEnabled ? urlSpellingService : undefined
    },
    links: {
      validation: {
        url: isLinkValidationEnabled ? urlLinkService : undefined
      },
      embed: {
        url: isMediaEmbedEnabled ? urlLinkService : undefined
      }
    },
    cognitive: {
      tone: {
        url: isToneEnabled ? urlToneService : undefined
      }
    },
    textboxioExtensions: [
    ],
    paste: {
          style: 'plain'
        }
  };
...
```

Adding Additional Buttons

Adding buttons to the editor provides a convenient way of exposing new functionality to users.

The IBM sample configuration contains a helper function for building buttons. It has the following signature:

```
button(id, tooltipText, iconPath, action);
```

The action parameter is a function with the following signature:

```
function(wcmEditorId) {
  // do action here
}
```

The `wcmEditorId` can be used to access Textbox.io and modify its contents.

Example snippets

Refer to the API reference for full details but here are a few examples.

Get editor content

Get the body of the editor document as HTML.

```
var content = ephox.wcm.api.getEditor(wcmEditorId).getBody();
```

Get selected text

Get the currently selected text. All HTML tags will be removed from the returned value.

```
var selectedText = ephox.wcm.api.getEditor(wcmEditorId).getSelection().getText();
```

Get selected HTML

Get the currently selected HTML. Partially selected tags are automatically closed at the selection boundaries.

```
var selectedHtml = ephox.wcm.api.getEditor(wcmEditorId).getSelection().getHtml()
```

Replace editor content

Completely replace the body of the editor document with the new HTML.

Side effects

- This will clear the current selection.
- This will clear the undo list.

```
ephox.wcm.api.getEditor(wcmEditorId).setBody(content);
```

Replace selection

Overwrite the current selection with the HTML snippet or insert it at the current cursor position if no selection has been made.

Side effects

- This will clear the current selection.

```
ephox.wcm.api.getEditor(wcmEditorId).insertHtmlAtCursor(htmlSnippet);
```

Get text direction

Query the text direction.

```
var dir = ephox.wcm.api.getEditor(wcmEditorId).getDirection();
```

Basic Example

The following example:

- Creates a new button
- Creates a new button group containing the new button
- Adds the new button group to the default toolbar.

```
...

  // Create a new button using the helper function.
  var newButton = button("newButton", "New button tooltip",
                         "<path-to-an icon>", function(wcmEditorId) {alert("Click!");});
  // Create a new button group
  var newGroup = {
    label: "New Button Group",
    items: [newButton]
  };
  // Append the new button to the existing array of toolbar items.
  items.push(newGroup);

  var config = {
    css: {
      stylesheets: styleSheetUrl ? [styleSheetUrl] : []
    },
    images: {
      editing: {
        proxy: isImageProxyEnabled ? urlImageProxyService : undefined
      },
      allowLocal: !isLimitedToLibraryImagePicker
    },
```

```
    ui: {
      locale: locale,
      toolbar: {
        items: items
      }
    },
    codeview: {
      enabled: isCodeViewEnabled
    },
    spelling: {
      url: isSpellCheckingEnabled ? urlSpellingService : undefined
    },
    links: {
      validation: {
        url: isLinkValidationEnabled ? urlLinkService : undefined
      },
      embed: {
        url: isMediaEmbedEnabled ? urlLinkService : undefined
      }
    },
        cognitive: {
      tone: {
        url: isToneEnabled ? urlToneService : undefined
      }
    },
    textboxioExtensions: [
    ]
  };

...
```

Advanced Example

The following example:

- Creates two buttons.
- Creates a new item group.
- Adds the new group to the toolbar.
- Shows how you might change the document with buttons

```
...

  // ROT13 encode some English alphabet text leaving all other characters untouched
  var rot13Text = function(text) {
    var out = '';
    var char_a = 'a'.charCodeAt(0);
    var char_z = 'z'.charCodeAt(0);
    var char_A = 'A'.charCodeAt(0);
    var char_Z = 'Z'.charCodeAt(0);
    for (var i = 0; i < text.length; i++) {
      var char = text.charCodeAt(i);
      var outChar;
      if (char >= char_a && char <= char_z) {
        out += String.fromCharCode((((char - char_a) + 13) % 26) + char_a);
      } else if (char >= char_A && char <= char_Z) {
        out += String.fromCharCode((((char - char_A) + 13) % 26) + char_A);
      } else {
        out += text.charAt(i);
      }
    }
    return out;
  };

  // ROT13 encode the text nodes in some HTML, leaving the HTML untouched
  var rot13Html = function(html) {
    var container = document.createElement('div');
    container.innerHTML = html;
    // depth first traversal
    var node = container.firstChild;
```

```
  while (node !== null && node !== container) {
    // check for a text node
    if (node.nodeType === 3) {
      node.nodeValue = rot13Text(node.nodeValue);
    }
    // transition to next node in depth first traversal
    if (node.firstChild) {
      node = node.firstChild;
    } else if (node.nextSibling) {
      node = node.nextSibling;
    } else {
      // find the first parent that has a sibling and transition to it
      node = node.parentNode;
      while (node !== null && node !== container && node.nextSibling === null) {
        node = node.parentNode;
      }
      if (node !== null && node !== container) {
        node = node.nextSibling;
              }
    }
  }
  return container.innerHTML;
};

// ROT13 encode the entire editor contents
// Side effects:
//   - Will clear the undos
//   - Will loose selection
var rot13Body = function(wcmEditorId) {
  var editor = ephox.wcm.api.getEditor(wcmEditorId);
  editor.setBody(rot13Html(editor.getBody()));
};

// ROT13 encode the currently selected text
// Side effects:
//   - Will loose selection
var rot13Selection = function(wcmEditorId) {
      var editor = ephox.wcm.api.getEditor(wcmEditorId);
  var selection = editor.getSelection();
  editor.insertHtmlAtCursor(rot13Html(selection.getHtml()));
};

// Create 2 new buttons using the helper function.
var encryptDocumentButton = button("encryptDocument", "Encrypt the entire document",
                                   "<path-to-an icon>", rot13Body);
var encryptSelectionButton = button("encryptSelection", "Encrypt the current selection",
                                    "<path-to-an icon>", rot13Selection);

// Create a group to hold the buttons
var encryptionButtonGroup = {
  label: 'Encryption',
  items: [encryptDocumentButton, encryptSelectionButton]
};

// Append the new group to the existing array of toolbar items.
items.push(encryptionButtonGroup);

var config = {
  css: {
    stylesheets: styleSheetUrl ? [styleSheetUrl] : []
  },
  images: {
    editing: {
      proxy: isImageProxyEnabled ? urlImageProxyService : undefined
    },
    allowLocal: !isLimitedToLibraryImagePicker
  },
  ui: {
    locale: locale,
    toolbar: {
      items: items
```

```
      }
    },
    codeview: {
      enabled: isCodeViewEnabled
    },
    spelling: {
      url: isSpellCheckingEnabled ? urlSpellingService : undefined
    },
    links: {
      validation: {
        url: isLinkValidationEnabled ? urlLinkService : undefined
      },
      embed: {
        url: isMediaEmbedEnabled ? urlLinkService : undefined
      }
    },
    cognitive: {
      tone: {
        url: isToneEnabled ? urlToneService : undefined
      }
    },
    textboxioExtensions: [
    ]
  };

...
```

Modifying Existing Editor Configurations

It may be desirable to remove functionality from the editor in certain situations.

The following example:

- Creates a new toolbar configuration without the "Insert" button group.
- Adds the new configuration to the toolbar.

```
...

  // This is the default definition for the toolbar items in the sample config
  var items = flatten([
    ['undo', insertGroup, 'style', 'emphasis', 'align', 'listindent'],
    styleGroup('format'),
    [languageGroup, toolsGroup]
  ]);

  // Creating a new array of items, filtering out the one
  var newItems = items.filter(function(item) {return item !== insertGroup});

  var config = {
    css: {
      stylesheets: styleSheetUrl ? [styleSheetUrl] : []
    },
    images: {
      editing: {
        proxy: isImageProxyEnabled ? urlImageProxyService : undefined
      },
      allowLocal: !isLimitedToLibraryImagePicker
    },
    ui: {
      locale: locale,
      toolbar: {
        items: items
      }
    },
    codeview: {
      enabled: isCodeViewEnabled
    },
```

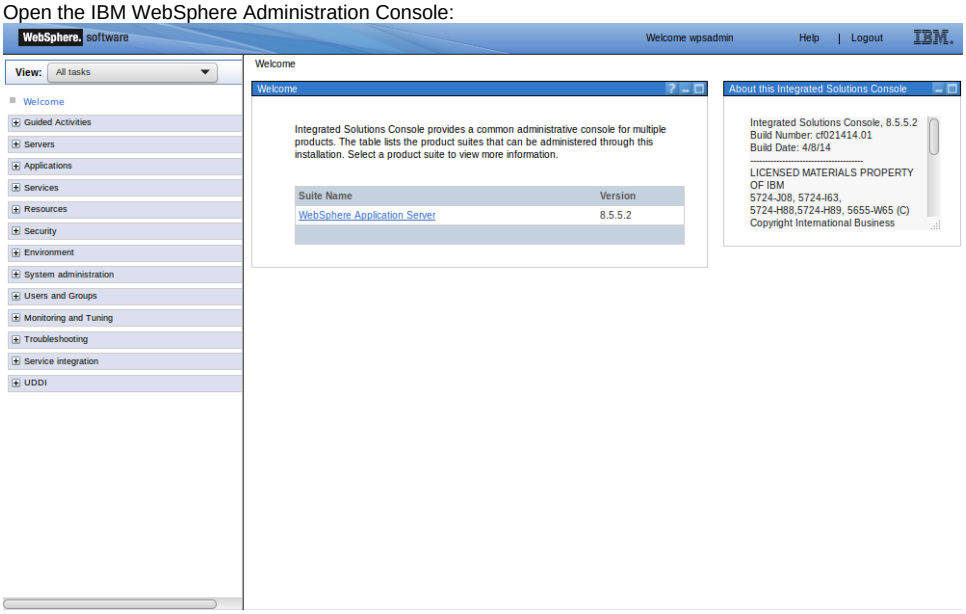
```
    spelling: {
      url: isSpellCheckingEnabled ? urlSpellingService : undefined
    },
    links: {
      validation: {
        url: isLinkValidationEnabled ? urlLinkService : undefined
      },
      embed: {
        url: isMediaEmbedEnabled ? urlLinkService : undefined
      }
    },
    cognitive: {
      tone: {
        url: isToneEnabled ? urlToneService : undefined
      }
    },
    textboxioExtensions: [
    ]
  };

...
```

Enabling inplace editing

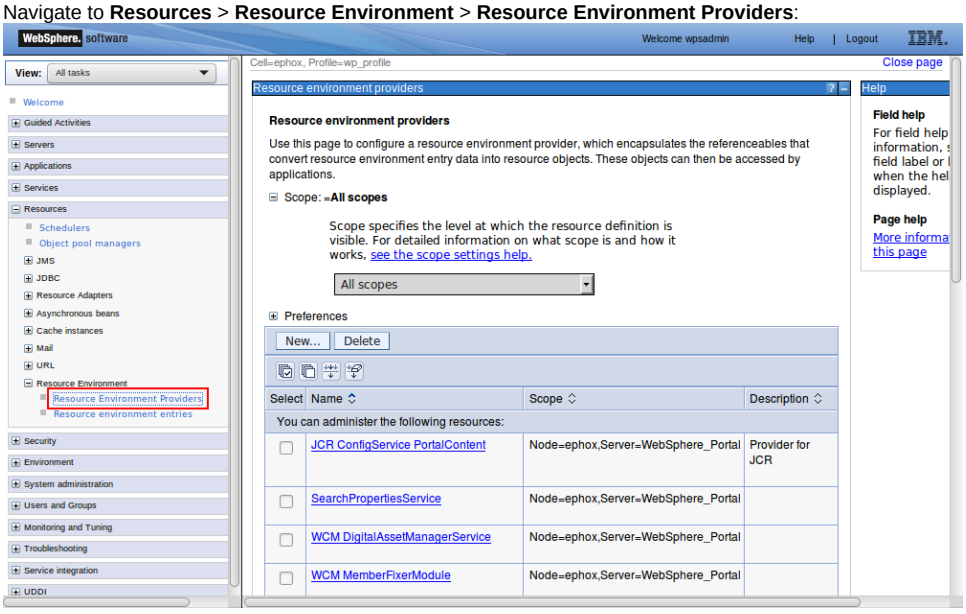
Please contact IBM if you require support.
Inplace editing does not currently support the Tone Analyzer.

To Enable Textbox.io for Inplace editing:

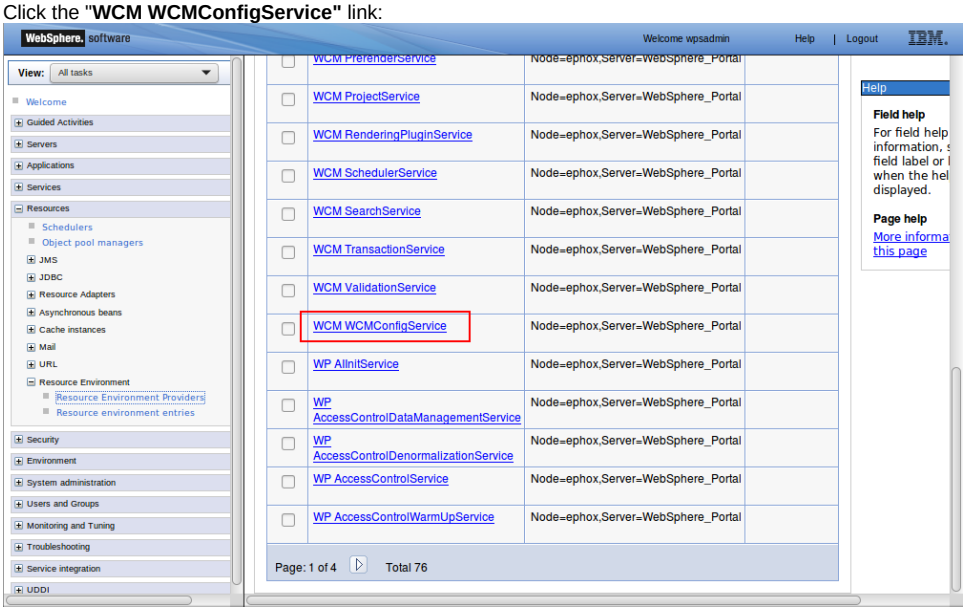
1. Open the IBM WebSphere Administration Console:



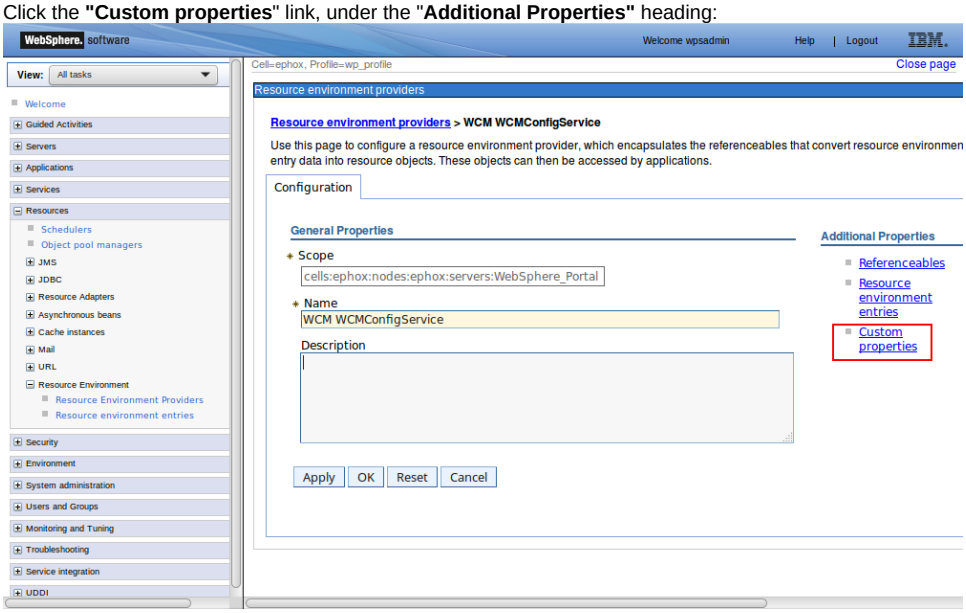
2. Navigate to **Resources** > **Resource Environment** > **Resource Environment Providers**:



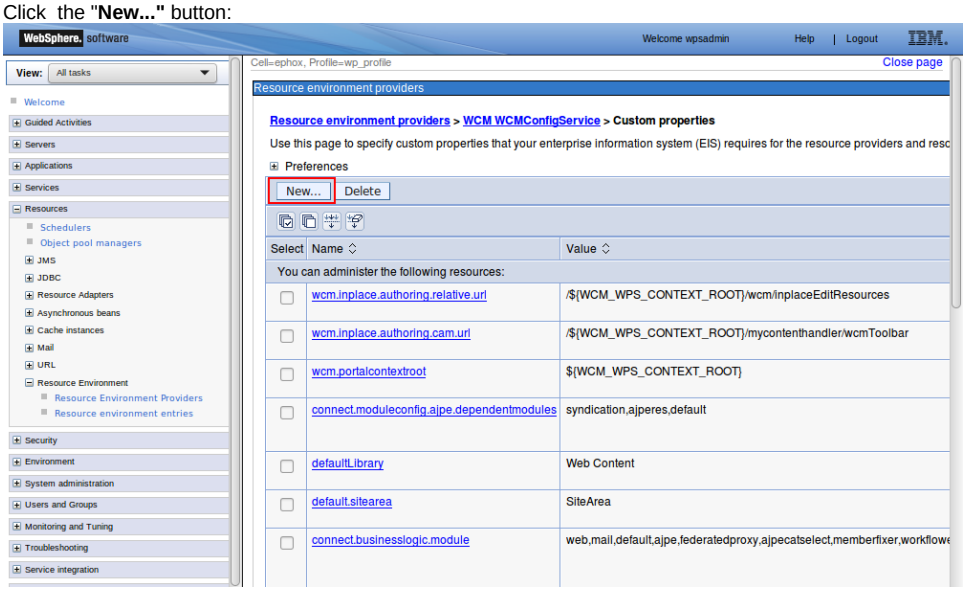
3. Click the "**WCM WCMConfigService"** link:



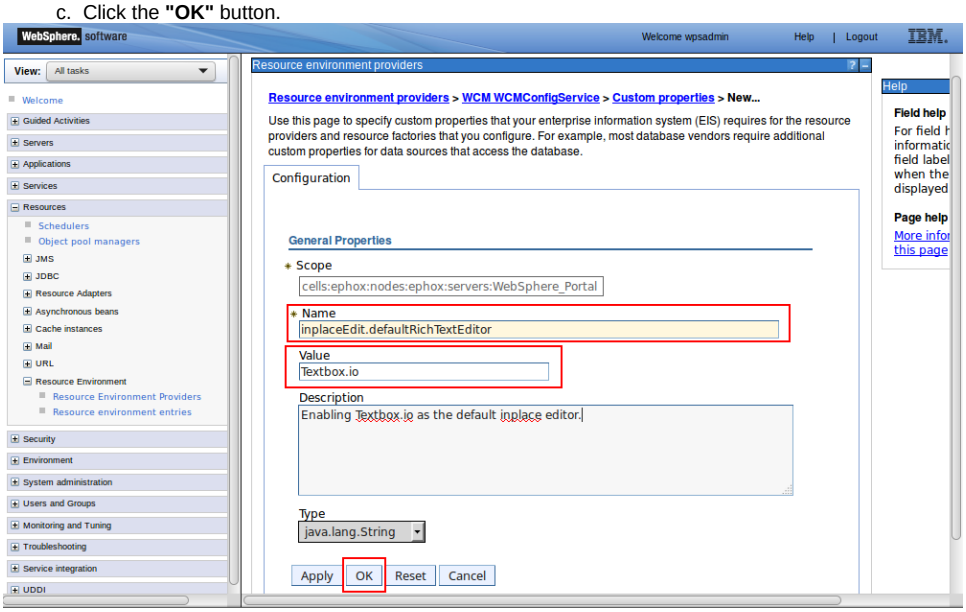
4. Click the **"Custom properties**" link, under the "**Additional Properties"** heading:



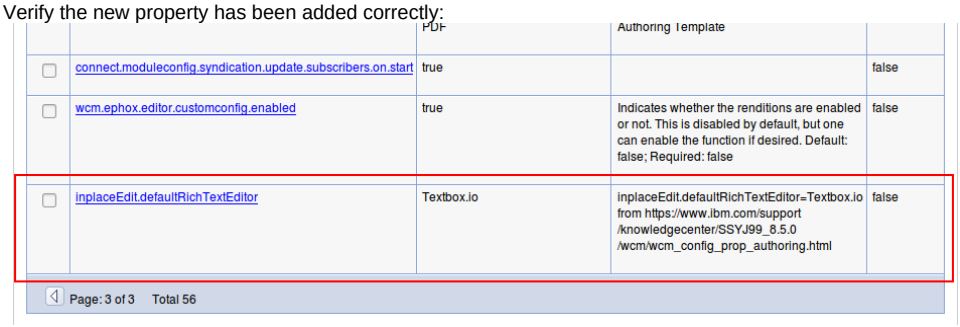
5. Click the "**New...**" button:



6. Apply the required values:
    a. In the "**Name**" field, enter: "inplaceEdit.defaultRichTextEditor".
    b. In the "**Value**" field, enter "Textbox.io".
    c. Click the **"OK"** button.



7. Verify the new property has been added correctly:



8. Restart the server

Additional Information

For additional information regarding web content authoring options and setting service configuration properties, please refer to the following IBM resources:

https://www.ibm.com/support/knowledgecenter/SSYJ99_8.5.0/wcm/wcm_config_prop_authoring.html

https://www.ibm.com/support/knowledgecenter/en/SSYJ99_8.5.0/admin-system/adsetcfg.html

API reference

Please contact IBM if you require support.

Note for backwards compatibility some global functions are available that provide some of the same functionality but they may be removed in future releases so please do not use them.

API

The WCM integration API is loaded at the `ephox.wcm.api` namespace and provides all the functionality needed to access the editors.

Each of the following functions should be prefixed by `ephox.wcm.api`.

Example

```
var editor = ephox.wcm.api.getEditor(wcmEditorId);
editor.insertHtmlAtCursor("<strong>Hello world</strong>");
```

Name	Parameters	Returns	Description
getEditor	String id	Editor	The Editor with the specific Id.
getEditors	none	Array	An array of available Editors.
getEditorMap	none	Object	A map with the available editors and their Ids.
anyEditorsDirty	none	boolean	A boolean value indicating if any of the available editors is dirty.

Editor

The Editor object represents an instance of a rich content editor and provides methods to interact with its contents.

Name	Parameters	Returns	Description
getId	none	String	The editor id.
getBody	none	String	Get the body element of the editor.
setBody	String html	void	Set the body element inside the editor.
insertHtmlAtCursor	String html	void	Insert the string passed as HTML in the position of the cursor
getSelection	none	Selection	Get the Selection object of the editor.
getDirection	none	String	Get the direction of the editor.
isDirty	none	Boolean	A boolean value indicating if there has been any changes in the editor content.

Legacy API

These global functions are provided for backwards compatibility, but should be avoided in favor of the current API as they may be removed in future versions.

These global functions should be avoided in favor of the current API as they may be removed in future versions.

Name	Parameters	Returns	Description
setHtml	String id String html	void	Set the passed HTML as the body of the editor with the matching Id.
getHtml	String id	String	Get the HTML body of the editor with the matching Id.
insertHtmlAtCursor	String id String html	void	Insert HTML content at the cursor position for the editor with the matching Id.
getEditorSelection	String id	Selection	Get the Selection object of the editor with the matching Id.
getDirection	String id	String	Get the text direction attribute value.
runCallbackUsingEphoxSelectedText	Function callback	void	Run a callback function with the selected text and a set of parameters

	Object parameters		
getEphoxCallbackParameters	none	Object	Get the parameters set in the last callback