

Migrating from SCAYT 2 to SCAYT 3

This is to provide you with step-by-step instructions that how to migrate from an old deprecated version of SCAYT product (v.2) to a new SCAYT (v.3).

SCAYT 2 is a very old version of the SpellCheckAsYouType product, all the development work including bug fixing, enhancements was stopped a while ago. The WebSpellChecker team decided to develop a new version of the product which based on the modern technologies and approaches.

Right now SCAYT 3, as well as components/plugins based on it, are stabled and being actively developed and maintained. New updates, features and bug fixing are announced in our each major release.

Thus, we strongly recommend you to consider the migration to a newer version of the SCAYT product in order to have access to the service and continue receiving updates including the latest features and enhancements.

1. How to migrate from SCAYT 2 to SCAYT 3 plugin for CKEditor
2. How to migrate from SCAYT 2 to SCAYT 3 plugin for TinyMCE
 - TinyMCE 3.0 and less
 - TinyMCE 4.0+
3. How to migrate from SCAYT 2 to SCAYT 3 for Editable Controls
 - 3.1. Contenteditable DIV Element
 - 3.2. Contenteditable IFRAME Element
 - 3.3. Contenteditable INPUT Element
 - 3.4. Contenteditable TEXTAREA Element

1. How to migrate from SCAYT 2 to SCAYT 3 plugin for CKEditor

SCAYT 2 product was used on the old versions of CKEditor 4.3.3 and less. The new version of the SCAYT 3 product was launched starting CKEditor 4.4+.

1. Upgrade the version of the CKEditor editor to the latest one available (or at least to the version 4.4+). The latest version of CKEditor plugin can be downloaded from [CKEditor website](#).
2. Specify your encrypted customer ID for the paid SCAYT service in the CKEditor **config.js** file using the **scayt_customerid** parameter:

```
config.scayt_customerid = 'encrypted-customer-ID';
```



You can find your long encrypted customer ID in your previous configuration or you may download it from your account page on the [webspellchecker.net](#) website. [Login](#) using your Customer ID and password. Then click **Download Configuration** file.

You can find more information about SCAYT 3 parameters for CKEditor [here](#).

2. How to migrate from SCAYT 2 to SCAYT 3 plugin for TinyMCE

Before to proceed with the migration steps, you need to define the exact version of the TinyMCE editor:

- Go to Development tools (e.g. F12 in Chrome).
- Choose the Console tab.
- Write after ">": **tinymce.majorVersion + '.' + tinymce.minorVersion**;
- Click Enter to see the version of the TinyMCE editor.

Depending on the version of TinyMCE editor, please follow the next instructions below:

TinyMCE 3.0 and less

- Upgrade the version of the TinyMCE editor to the latest one available (or at least to the version 4.0+). The latest version of TinyMCE (4.6.4) can be downloaded from the [TinyMCE website](#).
- [Download](#) SCAYT 3 plugin for TinyMCE.
- Unpack the downloaded ZIP archive with the SCAYT plugin.
- Copy the extracted SCAYT folder to the TinyMCE 4 plugins directory (by default: `tinymce4\js\tinymce\plugins\`). **Important:** The folder name must be **"scayt"**.
- Go further with instructions for TinyMCE 4+.

TinyMCE 4.0+

Edit your HTML page that contains TinyMCE as shown below.

- Register the SCAYT plugin and add it to the TinyMCE toolbar:

```
<script src="/tinymce/js/tinymce/tinymce.min.js"></script>
<script>
  tinymce.init({
    selector: "textarea#myEditor",
    theme: "modern",
    language: "en",
    plugins: [ "scayt link image table contextmenu" ],
    toolbar: "scayt undo redo | bold italic | alignleft" +
      "aligncenter alignright alignjustify |" +
      "bullist numlist | link image",
```

- Specify the SCAYT settings as follows:

```
/* SCAYT Parameters */
    scayt_autoStartup: true,
    scayt_customerId: "encrypted-customer-ID",
    scayt_moreSuggestions:"on",
    scayt_contextCommands:"add,ignore",
    scayt_contextMenuItemsOrder: "control,moresuggest,suggest",
    scayt_maxSuggestions: 6,
    scayt_minWordLength: 4,
    scayt_slang: "en_US",
    scayt_uiTabs: "1,1,1",
    scayt_context_mode: "default",
    scayt_elementsToIgnore: "del,pre",

    class_filter : function(cls, rule) {
      return cls == 'scayt-ignore' ? false : cls;
    }
  })
```



You can find your long encrypted customer ID in your previous configuration or you may download it from your account page on the webspellchecker.net website. [Login](#) using your Customer ID and password. Then click **Download Configuration** file.

You can find more information about available parameters on the [SCAYT plugin for TinyMCE 4.0+](#) demo page.

3. How to migrate from SCAYT 2 to SCAYT 3 for Editable Controls

1. Change the SCAYT application URL:

First of all, you need to add a script on your page and define a path to the SCAYT application.

```
<script type="text/javascript" src="https://svc.webspellchecker.net/spellcheck31/1f/scayt3/scayt/scayt.js"></script>
```

2. Update SCAYT initialization in controls. Depending on the type of the controls (HTML elements) where the SCAYT functionality is embedded and used, please update your configuration as follows.

3.1. Contenteditable DIV Element



SCAYT 2:

```
<div contenteditable sc_dojotype="scayt.ui">
  This is an exampl of a sentence with two misspelled words. Just type text with misspelling to see how
  it works.
</div>
```

Initialize SCAYT functionality in <div> element using either imperative or declarative notation as it is shown in the examples below.

SCAYT 3 (Imperative Notation):

```
<div contenteditable id="container1">
  This is an exampl of a sentence with two misspelled words.
  Just type text with misspelling to see how it works.
</div>

<script>
  var instance1 = new SCAYT.SCAYT({
    container: document.getElementById("container1"),
    autoStartup: true,
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  });
</script>
```

SCAYT 3 (Declarative Notation):

```
<script>
  window.SCAYT_CONFIG = {
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  };
</script>

<div contenteditable id="container1" data-scayt-autocreate="true">
  This is an exampl of a sentence with two misspelled words.
  Just type text with misspelling to see how it works.
</div>
```

3.2. Contenteditable IFRAME Element



SCAYT 2:

```
<iframe sc_dojoType="scayt.ui" src="editable_doc.html">
```

Initialize SCAYT functionality in <iframe> element using either imperative or declarative notation as it is shown in the examples below.

SCAYT 3 (Imperative Notation):

```
<iframe id="container2" src="editable_doc.html"></iframe>

<script>
  var iframeInstance;
  var iframeElement = document.getElementById("container2");

  SCAYT.SCAYT.createScaytControl(iframeElement, {
    container: document.getElementById("container1"),
    autoStartup: true,
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  },
  function(instance) {
    iframeInstance = instance;
  }
  );
</script>
```

SCAYT 3 (Declarative Notation):

```
<script>
  window.SCAYT_CONFIG = {
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  };
</script>

<iframe id="container2" data-scayt-autocreate="true" src="editable_doc.html"></iframe>

/*Editable_doc.html example: */

<html>
  <head>
    <title>SCAYT for Editable Controls</title>
  </head>
  <body contenteditable>This is an exampl of a sentence with two misspelled words. Just type text with
  misspelling to see how it works.
  </body>
</html>
```

3.3. Contenteditable INPUT Element



SCAYT 2:

```
<input name="scayt_input" size="50" sc_dojoType="scayt.Input" lang="<language_short_code>"
value="This is an exampl of a sentence with two misspelled words.">
```

Initialize SCAYT functionality in <input> element using either imperative or declarative notation as it is shown in the examples below.

SCAYT 3 (Imperative Notation):

```
<input id="container3" type="text" value="This is an exampl of a sentence with two misspelled words.">

<script>
  var instance3 = new SCAYT.SCAYT({
    container: document.getElementById("container3"),
    autoStartup: true,
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  });
</script>
```

SCAYT 3 (Declarative Notation):

```
<script>
  window.SCAYT_CONFIG = {
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  };
</script>

<input id="container3" data-scayt-autocreate="true" type="text" value="This is an exampl of a sentence with
two misspelled words.">
```

3.4. Contenteditable TEXTAREA Element



SCAYT 2:

```
<textarea name="scayt_textarea" sc_dojoType="scayt.Textarea" scaytConfig.customDictionaryIds =
[1,3001];
lang="<language_short_code>" cols="40" rows="7">
This is an exampl of a sentence with two misspelled words.
</textarea>
```

Initialize SCAYT functionality in <textarea> element using either imperative or declarative notation as it is shown in the examples below.

SCAYT 3 (Imperative Notation):

```
<textarea id="container4">
  This is an exampl of a sentence with two misspelled words.
</textarea>

<script>
  var instance4 = new SCAYT.SCAYT({
    container: document.getElementById("container4"),
    autoStartup: true,
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  });
</script>
```

SCAYT 3 (Declarative Notation):

```
<script>
  window.SCAYT_CONFIG = {
    spellcheckLang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'svc.webspellchecker.net',
    servicePort: '443',
    servicePath: 'spellcheck31/script/ssrv.cgi',
    customerId: 'encrypted-customer-ID'
  };
</script>

<textarea data-scayt-autocreate="true" id="container4">
  This is an exampl of a sentence with two misspelled words.
</textarea>
```

You can find more information about SCAYT 3 parameters for Editable Controls [here](#).