HUMAN HEALTH

ENVIRONMENTAL HEALTH

PerkinElmer

ChemDraw

JavaScript API Reference Guide 1.0

PerkinElmer
For the Better

Last Updated: 01/02/18

Table of Contents

ChemDraw JavaScript API Reference Guide

This guide provides information on the ChemDraw JavaScript API.

The ChemDraw JavaScript API enables ChemDraw add-ins to access the various features of ChemDraw with JavaScript. With the latest version, the API can support limited access to the active document and the add-in container window.

To access the guide and tutorials for creating ChemDraw add-ins and using ChemDraw JavaScript API in ChemDraw add-ins, go to **Help>Contents>ChemDraw Add-ins**.

> ***Note:*** *The ChemDraw JavaScript API works only in ChemDraw add-ins. To access the add-in examples that demonstrate the usage of the API, go to ChemDraw Add-in Examples.*

Global Objects

ChemDrawAPI

The singleton global object to access the active document, the add-in container window, and etc.

activeDocument : Document

Gets the active document that is opened in ChemDraw.

Example

```
try {
    var activeDocument = ChemDrawAPI.activeDocument;
}
catch (err) {
    alert(err.message);
}
```

window : Window

Gets the add-in container window that contains the add-in that is currently running.

Example

```
try {
    var addinContainerWindow = ChemDrawAPI.window;
}
catch (err) {
    alert(err.message);
}
```

version : string

Gets the version of the ChemDraw JavaScript API.

Example

```
try {
    document.write(ChemDrawAPI.version);
}
catch (err) {
    alert(err.message);
}
```

Note: *The add-in example* ***Hello World*** *demonstrates the usage of* ***ChemDrawAPI.version****. To access it, go to ChemDraw Add-in Examples*

Classes

Document

Provides API to add and get data in a ChemDraw document.

> ***Note:*** *The add-in example* ***Document Data Importer and Exporter*** *demonstrates the usage of the API for adding and getting data in the active document. To access it, go to ChemDraw Add-in Examples.*

addCDXML(cdxmlText) : void

Adds a CDXML document to the active document. The active document acquires the style of the supplied CDXML document.

Parameters

Name	Type	Description
cdxmlText	string	The CDXML document to add to the active document

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addCDXML(cdxmlText);
}
catch (err) {
    alert(err.message);
}
```

getCDXML() : string

Gets the contents of the active document as CDXML.

Parameters

None

Returns

Return Type	Description
string	The contents of the active document as a CDXML string

Throws Exceptions

Yes

Example

```
try {
    var cdxmlText = ChemDrawAPI.activeDocument.getCDXML();
}
catch (err) {
    alert(err.message);
}
```

addCDXBase64Encoded(base64EncodedCDXText) : void

Adds a CDX document, encoded as Base64, to the active document. The active document acquires the style of the supplied CDX document.

Parameters

Name	Type	Description
base64EncodedCDXText	string	The CDX document to add to the active document. The string must be encoded as Base64

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addCDXBase64Encoded(base64EncodedCDXText);
}
catch (err) {
    alert(err.message);
}
```

getCDXBase64Encoded() : string

Gets the contents of the active document as Base64 encoded CDX.

Parameters

None

Returns

Return Type	Description
string	The contents of the active document as a Base64 encoded string

Throws Exceptions

Yes

Example

```
try {
    var base64EncodedCDXText = ChemDrawAPI.activeDocument.getCDXBase64Encoded();
}
catch (err) {
    alert(err.message);
}
```

addSMILES(smilesText) : void

Adds a structure, encoded in a SMILES string, to the active document. The document retains its original styling. 2D coordinates will automatically be generated for the structure.

Parameters

Name	Type	Description
smilesText	string	The SMILES encoded structure to add to the active document

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addSMILES(smilesText);
}
catch (err) {
    alert(err.message);
}
```

getSMILES() : string

Gets the contents of the active document as a SMILES string. If there is a mixture of structures and non-structures (e.g. clip-art, text etc), only the structures will be returned. If there are multiple structures, the SMILES string will be returned in dot-separated notation.

Parameters

None

Returns

Return Type	Description
string	The contents of the active document as a SMILES string

Throws Exceptions
Yes

Example

```
try {
    var smilesText = ChemDrawAPI.activeDocument.getSMILES();
}
catch (err) {
    alert(err.message);
}
```

addInChI(inChIText) : void

Adds a structure, encoded in an InChI string, to the active document. The document retains its original styling. 2D coordinates will automatically be generated for the structure.

Parameters

Name	Type	Description
inChIText	string	The InChI encoded structure to add to the active document

Returns
None

Throws Exceptions
Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addInChI(inChIText);
}
catch (err) {
    alert(err.message);
}
```

getInChI() : string

Gets the contents of the active document as an InChI string. If there is a mixture of structures and non-structures (e.g. clip-art, text etc), only the structures will be returned. If there are multiple structures, the InChI string will return them in its native format.

Parameters
None

Returns

Return Type	Description
string	The contents of the active document as an InChI string

Throws Exceptions

Yes

Example

```
try {
    var inChIText = ChemDrawAPI.activeDocument.getInChI();
}
catch (err) {
    alert(err.message);
}
```

getInChIKey() : string

Gets the contents of the active document as an InChIKey string. If there is a mixture of structures and non-structures (e.g. clip-art, text etc), only the structures will be returned. If there are multiple structures, the InChIKey string will return them in its native format.

> ***Note:*** *The InChIKey is a one-way format. A key can be generated from a structure, but a structure cannot be generated from a key. Thus there is no Add method for InChIKey.*

Parameters

None

Returns

Return Type	Description
string	The contents of the active document as an InChIKey string

Throws Exceptions

Yes

Example

```
 try {
    var inChIKeyText = ChemDrawAPI.activeDocument.getInChIKey();
}
catch (err) {
    alert(err.message);
}
```

addMolV2000(molV2000Text) : void

Adds a structure, encoded in a MolV2000 string to the active document. The document retains its original styling. If present, 2D coordinates will honored for the structures, otherwise they will automatically be generated.

Parameters

Name	Type	Description
molV2000Text	string	The MolV2000 encoded structure to add to the active document

Returns

None

Throws Exceptions

Yes

Example

```
 try {
    ChemDrawAPI.activeDocument.addMolV2000(molV2000Text);
}
catch (err) {
    alert(err.message);
}
```

getMolV2000() : string

Gets the contents of the active document as a MolV2000 string. If there is a mixture of structures and non-structures (e.g. clip-art, text etc), only the structures will be returned. If there are multiple structures, the MolV2000 string will return them in its native format.

Parameters

None

Returns

Return Type	Description
string	The contents of the active document as a MolV2000 string

Throws Exceptions

Yes

Example

```
try {
    var molV2000Text = ChemDrawAPI.activeDocument.getMolV2000();
```

```
}
catch (err) {
    alert(err.message);
}
```

addMolV3000(molV3000Text) : void

Adds a structure, encoded in a MolV3000 string to the active document. The document retains its original styling. If present, 2D coordinates will honored for the structures, otherwise they will automatically be generated.

Parameters

Name	Type	Description
molV3000Text	string	The MolV3000 encoded structure to add to the active document

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addMolV3000(molV3000Text);
}
catch (err) {
    alert(err.message);
}
```

getMolV3000() : string

Gets the contents of the active document as a MolV3000 string. If there is a mixture of structures and non-structures (e.g. clip-art, text etc), only the structures will be returned. If there are multiple structures, the MolV3000 string will return them in its native format.

Parameters

None

Returns

Return Type	Description
string	The contents of the active document as a MolV3000 string

Throws Exceptions

Yes

Example

```
try {
    var molV3000Text = ChemDrawAPI.activeDocument.getMolV3000();
}
catch (err) {
    alert(err.message);
}
```

addMol(molText) : void

This is a convenience method that will automatically detect whether the string is in MolV2000 or MolV3000 format, and add structures to the active document. If the string is in neither format, an exception will be thrown.

Note: *There is no equivalent GetMol() method. Call the GetMolV2000() or GetMolV3000() methods instead.*

Parameters

Name	Type	Description
molText	string	The MolV2000 or MolV3000 encoded structure to add to the active document

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addMol(molText);
}
catch (err) {
    alert(err.message);
}
```

addRXNV2000(rxnV2000Text) : void

Adds a reaction, encoded in an RXNV2000 string, to the active document. The document retains its original styling. If present, 2D coordinates will honored for the reactions, otherwise they will automatically be generated.

Parameters

Name	Type	Description
rxnV2000Text	string	The RXNV2000 encoded structure to add to the active document

Returns
None

Throws Exceptions
Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addRXNV2000(rxnV2000Text);
}
catch (err) {
    alert(err.message);
}
```

getRXNV2000() : string

Gets the contents of the active document as an RXNV2000 string. If there is a mixture of reactions and non-reactions (e.g. clip-art, text etc), only the reactions will be returned. If there are multiple reactions, the RXNV2000 string will return them in its native format.

Parameters
None

Returns

Return Type	Description
string	The contents of the active document as a RXNV2000 string

Throws Exceptions
Yes

Example

```
try {
    var rxnV2000Text =  ChemDrawAPI.activeDocument.getRXNV2000();
}
catch (err) {
    alert(err.message);
}
```

addRXNV3000(rxnV3000Text) : void

Adds a structure, encoded in an RXNV3000 string, to the active document. The document retains its original styling. If present, 2D coordinates will honored for the structures, otherwise they will automatically be generated for the structure.

Parameters

Name	Type	Description
rxnV3000Text	string	The RXNV3000 encoded structure to add to the active document.

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.activeDocument.addRXNV3000(rxnV3000Text);
}
catch (err) {
    alert(err.message);
}
```

getRXNV3000() : string

Gets the contents of the active document as a RXNV3000 string. If there is a mixture of reactions and non-reactions (e.g. clip-art, text etc), only the reactions will be returned. If there are multiple reactions, the RXNV3000 string will return them in its native format.

Parameters

None

Returns

Return Type	Description
string	The contents of the active document as a RXNV3000 string

Throws Exceptions

Yes

Example

```
try {
    var rxnV3000Text = ChemDrawAPI.activeDocument.getRXNV3000();
}
catch (err) {
    alert(err.message);
}
```

getPNGBase64Encoded(options) : string

Gets the contents of the active document as a PNG image.

Parameters

Name	Type	Default Value	Description
options.transparent	boolean	true	If true, the image will have a transparent background. If false, the background is generated along with the document contents
options.scalePercent	integer	100	The scaling applied to the image. 100% is actual size. A value greater than zero is expected, otherwise an exception will be thrown
options.borderSizeInPixels	integer	0	The border (gap) to be placed around the image in pixels. A value of zero or greater is expected, otherwise an exception will be thrown

Returns

Return Type	Description
string	The contents of the active document as a Base64 encoded PNG string

Throws Exceptions

Yes

Example

```
try {
    var pngBase64EncodedText = ChemDrawAPI.activeDocument.getPNGBase64Encoded({
                transparent: false,
                scalePercent: 100,
                borderSizeInPixels: 100
            });
}
catch (err) {
    alert(err.message);
}
```

Window

resizeTo(width, height) : void

Resizes the window using the specified width and height.

Parameters

Name	Type	Description
width	integer	The new width of the window. A value greater than zero is expected, otherwise an exception will be thrown
height	integer	The new height of the window. A value greater than zero is expected, otherwise an exception will be thrown

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.window.resizeTo(width, height);
}
catch (err) {
    alert(err.message);
}
```

close([callback]) : void

Closes the add-in container window if there is no parameter, or registers a callback function that will be called when the window closes if a parameter is provided.

Parameters

Name	Type	Description
callback	function	**Optional** The callback function that will called when the window closes

Returns

None

Throws Exceptions

Yes

Example

```
try {
    ChemDrawAPI.window.close();
}
catch (err) {
    alert(err.message);
```

```
}
```