HUMAN HEALTH

ENVIRONMENTAL HEALTH

PerkinElmer

ChemDraw

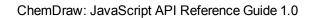
JavaScript API Reference Guide 1.0





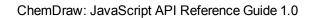
Table of Contents

ChemDraw JavaScript API Reference Guide	7
Global Objects	8
ChemDrawAPI	8
activeDocument : Document	8
Example	8
window : Window	8
Example	8
version : string	8
Example	8
Classes	10
Document	10
addCDXML(cdxmlText) : void	10
Parameters	10
Returns	10
Throws Exceptions	10
Example	10
getCDXML(): string	10
Parameters	10
Returns	10
Throws Exceptions	10
Example	11
addCDXBase64Encoded(base64EncodedCDXText) : void	11
Parameters	11
Returns	11
Throws Exceptions	11
Example	11
getCDXBase64Encoded(): string	11



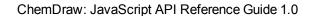


Parameters	11
Returns	11
Throws Exceptions	11
Example	12
addSMILES(smilesText) : void	12
Parameters	12
Returns	12
Throws Exceptions	12
Example	12
getSMILES(): string	12
Parameters	12
Returns	12
Throws Exceptions	13
Example	13
addInChI(inChIText) : void	13
Parameters	13
Returns	13
Throws Exceptions	13
Example	13
getInChI(): string	13
Parameters	13
Returns	14
Throws Exceptions	14
Example	14
getInChIKey(): string	14
Parameters	14
Returns	14
Throws Exceptions	14





Example	14
addMolV2000(molV2000Text) : void	15
Parameters	15
Returns	15
Throws Exceptions	15
Example	15
getMolV2000(): string	15
Parameters	15
Returns	15
Throws Exceptions	15
Example	15
addMolV3000(molV3000Text) : void	16
Parameters	16
Returns	16
Throws Exceptions	16
Example	16
getMolV3000(): string	16
Parameters	16
Returns	16
Throws Exceptions	16
Example	17
addMol(molText) : void	17
Parameters	17
Returns	17
Throws Exceptions	17
Example	17
addRXNV2000(rxnV2000Text) : void	17
Parameters	17





Returns	18
Throws Exceptions	18
Example	18
getRXNV2000(): string	18
Parameters	18
Returns	18
Throws Exceptions	18
Example	18
addRXNV3000(rxnV3000Text) : void	18
Parameters	19
Returns	19
Throws Exceptions	19
Example	19
getRXNV3000(): string	19
Parameters	19
Returns	19
Throws Exceptions	19
Example	19
getPNGBase64Encoded(options): string	20
Parameters	20
Returns	20
Throws Exceptions	20
Example	
Window	20
resizeTo(width, height): void	20
Parameters	21
Returns	21
Throws Exceptions	21



Example	21
close([callback]): void	21
Parameters	21
Returns	21
Throws Exceptions	21
Example	21



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.

```
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 <u>Base64</u>, to the active document. The active document acquires the style of the supplied CDX document.

Parameters

Name	Туре	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 <u>SMILES</u> string, to the active document. The document retains its original styling. 2D coordinates will automatically be generated for the structure.

Parameters

Name	Туре	Description
smilesText	string	The <u>SMILES</u> 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 <u>SMILES</u> 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 <u>SMILES</u> 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);
}
```

addInChl(inChlText): void

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

Parameters

Name	Туре	Description
inChlText	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 InChl 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 InChl string will return them in its native format.

Parameters

None



Returns

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

Throws Exceptions

Yes

Example

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

getInChlKey(): string

Gets the contents of the active document as an <u>InChlKey</u> 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 <u>InChlKey</u> string will return them in its native format.

Note: The InChlKey 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 InChlKey.

Parameters

None

Returns

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

Throws Exceptions

Yes

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



addMoIV2000(moIV2000Text): 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	Туре	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 <u>MolV2000</u> 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 <u>MolV2000</u> 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

```
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	Туре	Description	
molV3000Text	<u>string</u>	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 <u>MolV3000</u> 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 <u>MolV3000</u> 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	Туре	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	Туре	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	Туре	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 <u>RXNV3000</u> 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 <u>RXNV3000</u> 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

```
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	Туре	Default Value	Description
options.transparent	<u>boolean</u>	true	If true, the image will have a transparent background. If false, the background is generated along with the document contents
options.scalePercent	integer		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	U	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

Window

resizeTo(width, height): void

Resizes the window using the specified width and height.



Parameters

Name	Туре	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	Туре	Description
callback	function	Optional The callback function that will called when the
Caliback		window closes

Returns

None

Throws Exceptions

Yes

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



ı