NAME

smi2gif.cgi/fcgi/cas - chemical structure image generator for WWW applications

VERSION

1.10, 2008-3-2

SYNOPSYS

smi2gif.cgi?cgi-arg1&cgi-arg2..

DESCRIPTION

The *smi2gif.cgi* CGI application is a flexible tool for the dynamic generation of images of chemical structures or reactions for WWW applications. Contrary to the name of the application, the structure information can be supplied in various formats, not just SMILES.

The application can be run either as a traditional CGI script, as an FCGI script, or under the control of a custom Apache WWW server module. The latter two approaches have the advantage that the software operates as a constantly running image generation server and avoids the overhead of starting the program anew for every generated image.

The software is controlled by parameters which are passed in standard CGI style. A CGI URL can either be specified explicitly (i.e. /cgi-cbin/smi2gif.cgi?smiles=CCC), or indirectly via HTML forms with action URLS. The type of parameter passing (GET forms or the equivalent direct specification, POST forms, and parameter encoding variations (*application/x-www-form-urlencoded* or *multipart/form-data*) are detected automatically. For the preparation of HTML forms please refer to any HTML textbook. The names and possible values of the parameters are independent of the data transfer methodology. The examples in this documentation will, for the sake of simplicity, use the direct URL specification style.

When constructing CGI URLS for use with this application, please be careful so use correct URL-encoding conventions. The most common mistakes are to forget the encoding of space characters as '+', and the use of &, % or # as plain characters, not using the required hexadecimal encodings %26, %25 and %23. In the last case, the part after the # is interpreted as page anchor location.

Since these URL parts are interpreted by the Web server before the application script sees anything, incorrectly encoded CGI URLS can lead to confusing errors because the decoded parameter set may appear mysteriously shortened. Be aware that these special characters are prone to occur unexpectedly in variable parts of automatically generated URLs, such as comment texts. The characters % and # can occur in standard SMILES strings as ring linkages and triple bonds, and & in SMARTS as part of atomic or bond pattern expressions. The majority of SMILES strings does not contain these characters, so an encoding problem may not become immediately obvious. If the application fails mysteriously on certain compounds - check the SMILES parameter encoding!

In case of syntax errors in certain parameters (such as SMILES or SMARTS strings), unreadable files, or illegal parameter values, an error image with a message is generated instead of the structure depiction. However, not every malformed parameter will result in an error. Generally, all parameters which are not specific to the operation of the application script will be passed to the image generation procedure. If the parameter has an unrecognized name, it will be silently ignored.

If the structure source for the image does not possess 2D coordinates, they will be

automatically generated. If they are present, the available coordinate set will be used. If 3D-coordinates are present instead of 2D coordinates, this will be detected and 2D display coordinates are again computed. The 2D display is not a projection from 3D, but rather the result of a very sophisticated 2D layout algorithm. However, the presence of 3D coordinates can indirectly influence the display, for example by serving as a source of stereochemistry information, which will be automatically translated into a reasonable choice of dashed and solid wedge bonds in the 2D depiction.

INSTALLATION ON UNIX SYSTEMS

The simplest way to install the software is from a pre-packed distribution which contains the CGI script and the script interpreter. In this case,

- Create an empty temporary directory;
- Copy the compressed distribution tar file into this directory;
- Unpack with the command gunzip *tarfile* | tar xf -
- In case the distribution uses the simple stand-alone script interpreter *csweb*, simply copy it into the *cgi-bin* directory of the Web server. You can verify executability and print version and licensing information by invoking it as **csweb -v**
- In case the distribution uses a full CACTVS toolkit distribution, run the **installme** script in the unpacked directory and answer its questions. When asked whether this installation is a Web installation, answer with 'y'.
- For manual installation from other sources, first install a licensed CACTVS Web distribution in the CGI directory of the Web server. Note that, depending on the licensing status, a standard evaluation or educational distribution will not work with this application. Make sure that the standard structure processing interpreter wrapper *csts* is installed in the CGI directory and is executable. You can check it by manually starting *csts* - if everything is configured correctly, you will see a command prompt > and no error messages. The interactive command line interpreter can be left by pressing *Ctrl-D*, or by entering the *exit* command. Version information can be obtained by starting the program as **csts -v**.

Finally, copy the *smi2gif.cgi* CGI script into the CGI or FCGI directory of the Web server. Open the script with an editor and adjust the path to the *csts* or *csweb* interpreter (line 3) and the communication port (line 10, set to *none* for standard CGI operation, to *fcgi* to run it as an FCGI application, or the port number used by the custom *Apache* accelerator module). The script is intentionally provided as source code so that it can be locally customized.

In many Web server installations, the FCGI module is configured to automatically execute CGI applications as FCGI servers if the suffix of the executable or script is *.fcgi*. If this is the case, the script name suffix needs to be changed.

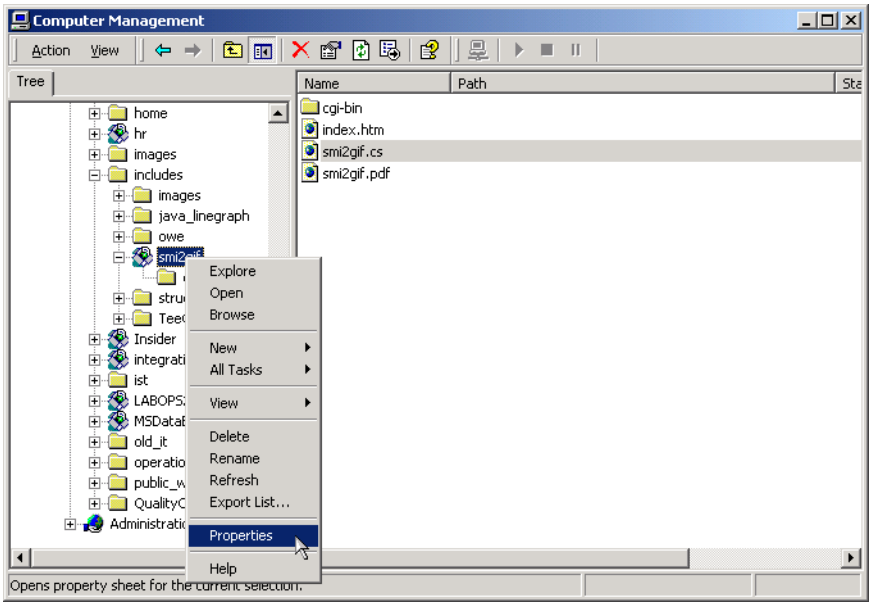
In case you edit the application script to customize it, be aware that the script interpreter, when running under the custom *Apache* module versions and potentially (depending on the Web server configuration) also under FCGI control, read the application script only once at start-up and do not monitor whether the script has changed. In order to observe the effects of edits, any currently running interpreter process must be killed. It will then be restarted by the custom *Apache* module or FCGI module when the first image generation

request is sent from a client browser. For standard CGI operation, where the interpreter is started anew for each program invocation, no interpreter process termination is required.

INSTALLATION ON WINDOWS SYSTEMS

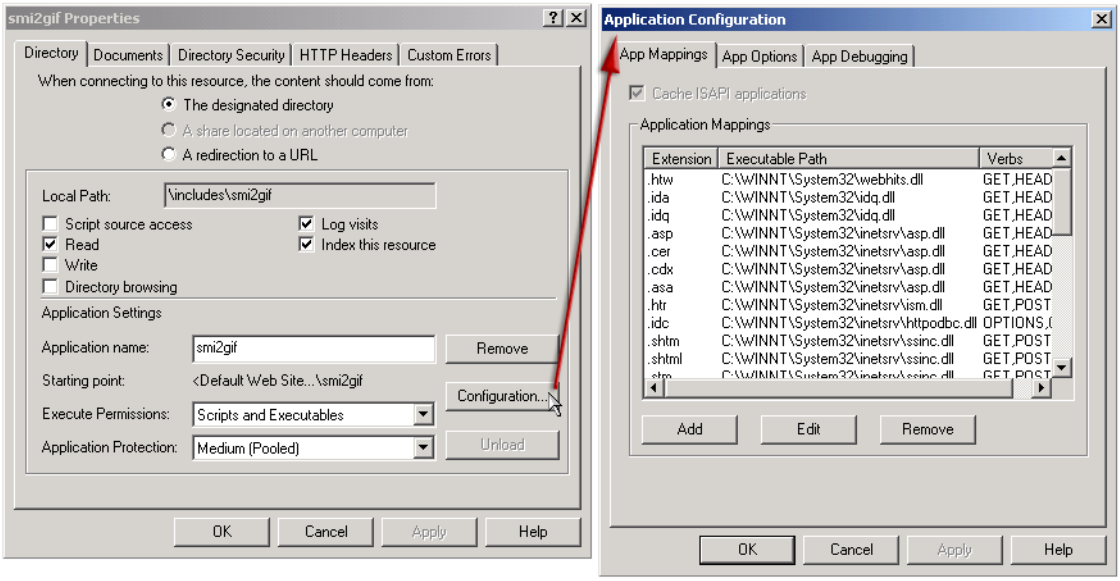
The Windows package is delivered as a *zip* file with the CGI set-up and a demo page. Follow the following steps for installation:

- Create a directory (such as *smi2gif*) in the IIS Web server file system area. It is usually located below C:\inetpub\wwwroot.
- Unpack the *zip* file in this directory
- Make sure that scripts can be executed in that directory (use the *inetmgr* tool)
- Associate the script file suffix *.cas* with the executable *csweb.exe* in the *cgi-bin* subdirectory. This is done via the *Properties* menu of *inetmgr* which is attached to the directory. Enter the path of the executable, followed by the parameter string '-f %s %s' - but *without* the quotes, similar to
  C:\inetpub\wwwroot\smi2gif\cgi-bin\csweb.exe -f %s %s
- The sequence of steps for the suffix association is further explained in the images below.
  Open in *inetmgr* the *Properties* panel via the right-button menu on the installation directory:

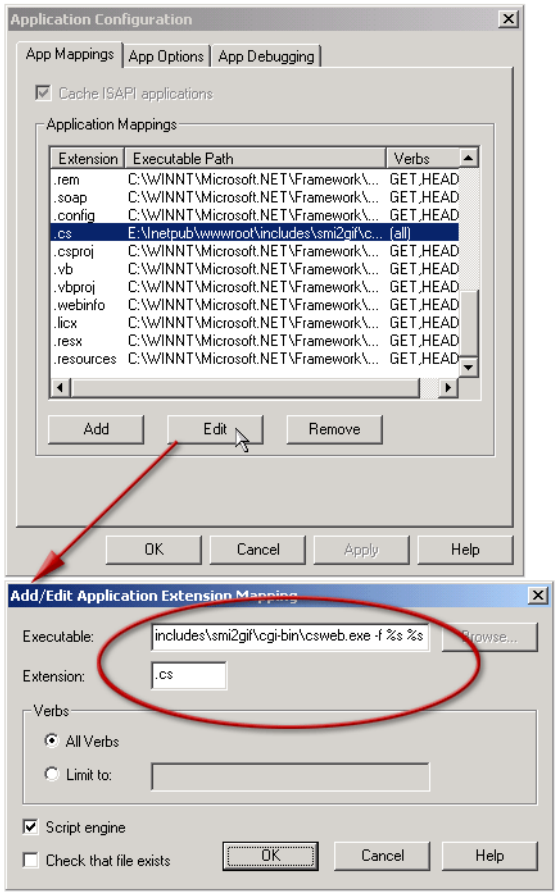


Make sure that scripts can be executed in the directory by setting the execution permis-

sion filed. Next press the *Configuration* button:



In the new panel, add an entry for the *.cas* extension if it is not yet present. Associate the suffix *.cas* with the full path to *csweb.exe*, including the parameter string as described above:



- If you already have another CACTVS Web scripting application installed, and have set up a global (not directory-specific) association of the *.cas* suffix with the *csweb.exe*

executable, this step can be omitted. The *csweb.exe* CACTVS System Web interpreters are identical. The application-specific capabilities are provided by the application scripts - in this case, the file *smi2gif.cas*. If this is the first *csweb.exe* installation, it is up to you whether you want to set up a global interpreter association or want to limit it to the *smi2gif* directory.

- No you can load the *index.htm* test file in the installation directory into a Web browser in order to verify the operation of the software. If everything is set up correctly, a simple table with six structure images which were generated on the fly will show up. If the images do not appear, something is wrong.

EXAMPLES

A typical simple Apache Web server use:

```
<IMG SRC=”/cgi-bin/smi2gif.cgi?smiles=CCO&atomcolor=black&height=60&width=60&bgcolor=black”>
```

A typical, a little bit more complex IIS command:

```
<IMG
SRC="smi2gif.cas?smiles=CC(=O)c1ccc(OCc2c([R2]C)cc([R3])cc2[R1])cc1&smarts=c1cccc(c1)C([%2
36])=O&bondcolor=black&width=200&height=200&bgcolor=white&atomcolor=black&R1=Cl&R2=S(
=O)(=O)&highlightatoms=1,2,5&highlightatomstyle=invert" ALT="test image" WIDTH="200"
HEIGHT="200">
```

Note that the only major difference between the Unix and Windows use is the location of the script - on Unix, these are typically installed in a dedicated script directory, while on IIS they tend to reside in the same directory as the rest of the pages. Another minor difference is the script file suffix. Other than that, the supported script parameters are absolutely identical, and both Web servers can be configured to operate in a style similar to the other.

OPTIONS

This application recognizes two classes of options: application-specific options and those which are passed to the generic image generator algorithm. The application-specific options are used to prepare and manipulate a chemical structure. For this structure an, image is then drawn, using the second set of generic image-generation options.

Application-specific options

smiles=*smiles_string*

Specification of a structure or reaction source as a SMILES string. Note that this is a structure or reaction definition, not a SMARTS/SMIRKS substructure definition. While they are still understood to a very limited degree in this context, SMARTS query-type specification should be avoided in this parameter. In addition to the syntax of standard structure or reaction SMILES, it is possible to use superatoms. Every atom specification in square brackets which is not a simple element is interpreted as a superatom. Example: *C[R1]*. The superatom will be displayed by default with the specified text label. However, superatoms may be expanded by providing additional parameters with the superatom definition(s).

Besides the basic structure definition, the optional text label part to the right (separated by white space) of the connectivity definition in the string is also analyzed. This is essentially a Daylight *depict* compatibility feature. The special definitions *#h=num*, *#w=num*, and *#c=num* are recognized, decoded and removed from that string. The remainder of the string is stored as structure name. The #h and #w parameters are used

as image height and width settings. #c is interpreted according to the somewhat cryptic *depict* color scheme: 0 - black on black (useless), 1 - black on white, 2 - black on paper (transparent), 10 - white on black, 11 - white on white (useless), 12 - white on paper (transparent), 20 - color on black, 21 - color on white, 22 - color on paper (transparent). An example in URL encoding (remember, + is the space character, %23 the hash/pound character #): COC+%23c=20+Dimethylether+%23w=100+%23h=100 will display dimethylether with colored atom symbols on a black background in an image of size 100x100 pixels, and the name of the structure will be set to „Dimethylether“. These parameters extracted from the name are merged with the standard parameters and can be used simultaneously.

As a final expert feature, this parameter will not only decode basic SMILES strings, but also detect and accept hex-encoded SMILES (as used by various Daylight applications) and CACTVS system serialized object strings (which are a compact and lossless method to encode structure data).

R1=*smiles_string*

Parameters similar to this are a template for a substituent group and not necessarily encoded literally as *R1*. Every parameter which has the same name as a previously defined superatom in a SMILES specification, or a molecule set read from an inline or file system structure file, will be interpreted as a definition of a structure fragment. Before display, all defined fragments will be expanded. The attachment point of the fragment is its first atom. It is legal to define only a subset of the superatom fragments for expansion, or not to provide any fragment definitions at all.

smarts=*smarts_string*

If this option is specified, the SMARTS string will be decoded as substructure and matched onto the structure (which was obtained from decoding a SMILES string, or reading a structure file). If a match was obtained, both the atoms and bonds of the first match are marked. By default, matched bonds will be highlighted with a special color, typically red. In this structure specification, the full SMARTS syntax is understood, including Recursive SMARTS, but superatom fragments are not allowed. The substructure match command (*match ss*) which processes this option is called with a very generic parameter setting in the generic script. Since it has dozens of options, it is a useful target for local modifications of the CGI script. The command is fully documented in the CACTVS Scripting Language manual.

molfile=*file_name*

The name of a file in a structure exchange format. The files may contain one or more structures or reactions. The file format is detected automatically. It does not need to be an MDL molfile. For example, SMILES or PDB files are acceptable as well. Note that the path of the file refers to the file system as seen from the server, which is usually different from what is seen on the client. In order to speed up the start of the CGI script, only a limited set of I/O modules are loaded by default. If you need to read structure files which are not handled by built-in I/O modules but can be handled by one of the numerous CACTVS file format handler extension modules, the required I/O modules must be loaded by a manually inserted *filex load* command in the CGI script.

inlinefile=*file_data*

This parameter is essentially the same as the **molfile** parameter. The only difference is that the *contents* of the file are passed, not its name. When using this kind of parameter passing mechanism, please take extra care to properly escape white space, carriage return characters, etc. in the URL. The usefulness of this parameter passing style is

limited by the maximum usable URL length, which varies with different client browsers. For example, IE5 simply ignores URLS above 4K in length and will not even contact the server with these. Therefore, these simple inline files should be used only for files of very limited size.

cinlinefile=*compressed_file_data*

This method of structure specification is the same as the **inlinefile** parameter, but the parameter value is the content of a *base64-encoded zlib-compressed* file. The file contents must first have been compressed with the standard *gzip* compressor, or *zlib* compression library, which is standard on most Unix platforms, and then *base64*-encoded to avoid non-printable characters. Because most structure exchange formats are typically quite compressible, formats such as MDL Molfiles can be used with much bigger structures than without compression. Normal drug-sized molecules should no longer be a danger by exceeding the URL length limits.

template=*smarts_string*

A SMARTS string which encodes a structure drawing template. The interpretation of this string is the same as with the **-smarts** option.

inlinetemplate=*file_data*

An image of a template file, such as an MDL molfile. The file contents must be properly URL-encoded. For caveats regarding the use of this option, refer to the **-infilefile** option.

cinlinetemplate=*compressed_file_data*

This option is used to pass a *base64-encoded, zlib-compressed* image of a template file. This option is a variant of the **-inlinetemplate** option, and the encoding is the same as in the **-cinlinefile** option.

templatemode=*mode*

When multiple structures are displayed on one page, it is often beneficial if they are all aligned according to a common substructure, or at least a common axis. This can be achieved with the **templatemode** and **template** parameters. The template mode may be one of *x*, *y*, *diagonal*, *rotate*, *redraw* or *besteffort*. The first three options align the displayed structure to the x, y or diagonal axis. They do not require a template. The *rotate* and *redraw* variants require such a template structure, which must be a matching substructure for the display structure in order to have any effect. *Rotate* will rotate the structure in multiples of 30 degrees, possibly combined with a flip, to the maximum alignment with the template structure, but will otherwise keep the structure layout, which may have been read from file, intact. *Redraw* will redraw the structure, using the template as a starting point. One important limitation is that in this mode the template cannot match ring systems partially. If the template contains a ring system, it must match the same class of ring system (same ring sizes and ring linkages) in the structure. A phenyl substructure fragment will not match a naphthalene structure part, and neither will any substructure chain fragment! The only exception are terminal atoms (atoms with only a single neighbor) in the substructure. These may match onto ring atoms in the structure. If the match is successful, the matched atoms will have exactly the same coordinates as the template, and the rest of the structure will be drawn around this starting structure. The *besteffort* mode combines the last two modes: If the *redraw* mode fails, a second attempt with *rotate* is automatically initiated. The template mode *none* can be used to explicitly disable the template drawing mechanism.

offset=*n*

By default, the first structure is read from a multi-record structure file source.

However, by setting a non-zero offset parameter the first *n* records will be skipped and the following entry read and displayed. While this is a convenient mechanism for quick display of, for example, SD-files, without the need to split the data into single-record subfiles, skipping a large number of records in a file which does not contain a record offset index (such as an SD file) can be time-consuming. On files which do contain such an index (such as CACTVS database files) this option can be freely used with files of any size.

highlightatoms=*atomlist*

This option selects a set of atoms, identified by their labels, for highlighting. If the structure was obtained from a source without explicit labelling, such as a SMILES string or an MDL Molfile, the atom label is the sequence number, beginning with one. The list can be separated by whitespace or commas. The highlight style is determined by the **highlightatomstyle** parameter. If it is not set, it defaults to *circled*.

highlightbondlist=*atompairlist*

This parameter works similar to the **highlightatoms** parameter, but selects bonds. Bonds are identified by the pair of atoms which form the bond. The bond highlight style is determined by the **highlightbondstyle** parameter and defaults to *bold* if not specified.

highlightatomstyle=*style*

Select a style for atom highlighting with the **highlightatoms** parameter. The possible values for this attribute are a subset of the bits in the A_FLAGS property. Supported and actually useful flag values are: *starred, inverted, boxed, invisible, nosymbol, circled, label, highlight, diamond, triangle, cross.*

highlightbondstyle=style

As above, but for the bonds selected with the **highlightbondlist** parameter. The useful and supported styles, taken from the B_FLAGS property, are: *bold, dotted, dashed, slash, circle, square, cross, anchor, colorsplit, highlight, doubleslash, hash, invisible*.

Generic image generation options:

All options which are not recognized as application options are passed on to the image generator. Options which are not understood by that subsystem are silently discarded.

Name	Default	Explanation
annotationcolor	*(empty)*	Specify a color for atom and bond annotations, such as formal charges, stereo descriptors, atom mapping labels, etc. If this parameter is an empty string, the color of the individual atoms and bonds is used for the annotations.
annotationfontsize	-1	The font size used for atom and bond annotations. If set to a negative value, a suitable font size will be automatically chosen.

Name	Default	Explanation
antialiasing	3	If bit one (value 0/1) set, TrueType fonts (*fonts* parameter) are drawn with anti-aliasing. They appear much smoother that way. Since more colors are used, file sizes will be bigger. The built-in default font cannot be anti-aliased. Bit 2 (value 0/2) controls the same feature for the drawing of lines. The parameter can be set to any sum of those two sub-settings.
atomcolor	black	Color of the atom. This parameter is either the name of a color (specified either as a color name from the X11 color database [such as *red*], or in #-notation [such as *#FF0000* or *#F00* for red]), or the special value *type*. If this value is used, the color is taken from the A_COLOR property. If it was not set beforehand, it is set to a standard value dependent on the element. The Windows program version contains a standard X11 color name table - it is not necessary to run an X server for this.
asymbol	symbol	Style of atom symbol. Possible values: *none*: no symbol *symbol*: standard atom symbol *xsymbol*: extended atom symbol (i.e. search attributes, super atoms, etc.) *label*: atom label *index*: atom index *box*: colored square instead of atom symbol *compact*: atom symbol with hydrogen count
bead	0	If this parameter is set, certain super atom types (such as *polymer*) will not be displayed as symbol, but rather as a round bead shape.
beadcolor1	black	The outline of bead symbols (see *bead* parameter)
beadcolor2	gray60	The fill color of bead symbols (see *bead* parameter)
bgcolor	transparent	The background color of the image. The special value t*ransparent* can be used to mark the background as transparent. Other colors can be specified either as a color name from the X11 color database (such as *red*), or in #-notation (such as *#FF0000* or *#F00* for red).
bondcolor	black	Color of bonds. Colors can be specified either as a color name from the X11 color database (such as *red*), or in #-notation (such as *#FF0000* or *#F00* for red). The special value *split* can be used to split the bond into two segments which are each colored according to the color of the closer bond atom.

Name	Default	Explanation
bonds	8	Number of standard bonds in x-direction. Smaller molecules will be centered in the image box. Molecules which exceed the number of bonds in x-direction (as specified) or y-direction (computed from aspect ration) will be shrunk to fit into the box tightly. A larger value of this parameter will let the average molecule appear smaller, and less structures will have to be resized to that a more uniform impression is maintained.
border	12	Number of pixels between the extreme atom coordinates and the border of the image. Note that the coordinates of invisible atoms (such as suppressed hydrogens) are also taken into account. Since the characters of atom symbols take up some space beyond the atomic coordinate, this parameter should not be decreased too much, depending on the font size.
boxgroups	*(empty)*	A list of labels of groups which should be boxed. The selected groups are highlighted by drawing an enclosing rectangular box around the individual selected groups. The color of the enclosing box can be adjusted by the *groupcolor* parameter.
comment	E_NAME	Either a free-form comment text, which is embedded into the image, or the name of a property which contains the text of the hidden comment text. The meaning of this field is controlled by the *commenttype* parameter.
commenttype	property	Type of the hidden embedded image comment. Possible values are *none* (no comment), *text* (copy content directly from *comment* parameter) or *property* (the *comment* parameter contains the name of a property such as E_NAME, which is looked up for the current structure ensemble).
crop	-1	If this value is greater or equal to zero, the unused image borders on all four sides are removed. The processing will remove, beginning from the outsides, all consecutive rows or columns which contain only pixels in the background color. If this parameter is larger than zero, a new border in the background color with a width corresponding to the value of this parameter is added.

Name	Default	Explanation
csymbol	special	Indicates which carbon atoms are displayed with an atom symbol. Other carbon atoms are displayed as bond graph nodes only. In order to make a carbon atom disappear altogether, the *invisible* A_FLAGS attribute must have been set. Possible values: *none*: do not display carbon atoms *special*: display only selected carbon atoms (charged, triple bonds, etc.) *all*: display all carbon atoms with a C symbol
dashes	1	If set (the default), dashed and bold lines are printed as such. If both the *dashes* and *wedges* parameters are unset, atomic stereochemistry output is suppressed.
font	*(empty)*	The name of a TrueType font. It can either be the name of a font file in a standard location (*data/fonts* directory in the standard CACTVS toolkit distribution), or the full path name of a font. This font is used for atom symbols and annotations in various scaling steps. The standard distribution contains a number of TrueType fonts from a Linux X11 distribution, such as *Arial* and *ArialNarrow*. If no font is specified, a built-in standard pixel font is used. TrueType fonts can be anti-aliased (*antialiasing* parameter) and can thus appear much smoother than the blocky, not anti-aliased default font.
footer	*(empty)*	Text of footer line. If the string is empty, the footer text may be obtained from a structure property (*footerproperty* parameter)
footercolor	black	The color of the footer text (taken from the *footer* or *footerproperty* parameters)
footerproperty	*(empty)*	The name of a property which is read from the current structure to fill the footer line. If the footer line is set explicitly (*footer* parameter), that text has precedence. If the parameter is empty, it is ignored.

Name	Default	Explanation
format	gif	The image format. Currently, GIF, PNG, PNG8, WMF, EMF and EPS and various Windows bitmap formats (BMP24, BMP8, BMP4, BMP1, CBMP8, CBMP4) are supported. The difference between PNG8 and PNG is that the former is always an 8-bit colormap image, which may imply a reduction in color space. By default, any image with line antialiasing (parameter *antialiasing*) is internally drawn as a true-color 24-bit image. For normal structure images (8-10 different atom symbol base colors, antialiased lines, and a few pure colors for the background etc.) no color reduction is required, so for most applications, PNG8 is preferred because it is only about half the size of a true-color PNG. GIF images are always colormap-based, because this format does not support any other model.
groupcolor	purple	The color in which the enclosing box is draw around groups specified with the *boxgroups* parameter.
hcolor	*(empty)*	Override color value for hydrogen atoms. If this color is not set (or set to an empty string), the usual atom color (see *atomcolor* parameter) is used.
headercolor	black	The color of the header text (taken from the *header* or *headerproperty* parameters)
header	*(empty)*	Text of header line. If the string is empty, the header text may be obtained from a structure property (*headerproperty* parameter)
headerproperty	*(empty)*	The name of a property which is read from the current structure to fill the header line. If the header line is set explicitly (*header* parameter), that text has precedence. If the parameter is empty, it is ignored.
height	180	Height of image in pixels.
highlightbonds	*(empty)*	A list of bonds (expressed as bond labels) which are not plotted in the usual bond color (see *bondcolor* parameter), but in the color specified with the *highlightcolor* parameter.
highlightcolor	red	The color used to highlight selected atoms and bonds. Highlighting is activated for atoms with a set *highlight* flag in the A_FLAGS property, for bond with a set *highlight* flag in the B_FLAGS property, and bonds listed in the *highlightbonds* parameter.

Name	Default	Explanation
highlightgroups	*(empty)*	A list of labels of groups which should be highlighted. Highlighting is performed by implicitly adding the *highlight* flag to all bonds between the atoms of the selected groups.
hsymbol	special	Indicates which hydrogen atoms are displayed. Hydrogen atoms which are not displayed are suppressed completely, no bonds leading to them will be drawn. Possible values: *none*: display no hydrogens *special*: display only selected hydrogens (aldehyde, stereo bonds) *all*: display all hydrogens
httpheader	0	Flag to specify whether the image file output should be prefixed by a standard http protocol header (=1), a http protocol header with a status code (=2) or no header (=0). Very useful if output is via the standard output channel in a CGI environment, and a script generates an image on the fly. The http protocol header type 2 will additionally add a 200 HTTP status code. This variant should be used in an NPH (non-parsed-headers) context. This attribute is modified internally in the script and should not be changed.
interlace	1	Flag whether the result image should be interlaced (=1) or not (=0). Interlaces images display an initial coarse image faster in a Web browser, becoming clearer gradually. However, they are somewhat larger in file size.
isotopemapping	none	If this parameter is set a a valid value of the A_FLAGS property (such as *boxed* or *circle*), that flag will be automatically added to the attribute set of the atom if it has an isotopic label. This option is intended to be used when isotope labelling is abused to mark some atoms. In this case, it is usually combined with a *showisotope 0* flag in order to suppress the display of spurious isotope data.
linewidth	1.4	Standard width of a line. This is rounded down to 1 for simple lines in pixel formats. Bold lines etc. are drawn with a fixed multiplicator. Increasing this factor lets the structure appear bolder, but on simple pixel formats a base line width of two is overkill. For vector formats (WMF, EMF) a gradual adjustment is useful.

Name	Default	Explanation
logofile	*(empty)*	The name of a GIF or PNG image file which will be added as a trademark logo on the upper left corner of the image. If cropping is active (*crop* parameter), the image will be added after cropping, potentially obscuring part of the structure. Otherwise, structures are drawn over the logo if they extend into the upper left corner.
logoscale	1.0	Scaling of the image read from the *logofile* parameter. This is a floating.point number.
markcolor	blue	Color used to highlight atoms with various symbols (such as the A_FLAGS attributes *diamond*, *circle*, *triangle*, *boxed*, *cross*)
metadata	1	Flag whether to embed a standard Dublin Core metadata set in the image or not. Only useful if you intend to store the image in an image library and want to use automatic indexing.
showcharge	1	A flag indicating whether formal atomic charges should be displayed (=1) or not (=0).
showempty	0	If this flag is set, the name (property E_NAME) of an empty structure (with zero atoms) will be plotted in the middle of the display box as an indicator of its content. By default, a blank image is output for these structures.
showisotope	1	A flag indicating whether isotope labelling should be displayed (=1) or not (=0)
showmapping	0	If set, atom mapping information will be drawn if it is available in the input data. Atom mapping information is often used in encoding reactions.
showradical	1	A flag indicating whether radical markers on atoms should be displayed (=1) or not (=0)
showstereo	1	A flag indicating whether stereo descriptors (such as R/S) on atoms should be displayed (=1) or not (=0). Note that this parameter is independent of wedge bond display (*wedges* and *dashes* parameters). Stereo descriptors are not computed, they must be available when the image is computed in order to be displayed.
showstereoh	1	A flag indicating whether a hydrogen at a stereo center should always be displayed (=1), regardless whether it is bonded via a wedge bond or not (=0).

Name	Default	Explanation
structure	none	Type of the embedded structure information in the image. Possible values are *none* (no embedded structure), *smiles* (embed SMILES string), *cactvs* (embed CACTVS serialized object) or *molfile* (embedding a molfile of the structure). The embedded structure information can be read by suitable software (such as the CACTVS system) to reclaim the structure information from the image, without using chemical OCR. Embedding structure information, especially in CACTVS or *molfile* types, will increase the file size notably.
suppressedmols	*(empty list)*	A list of the labels of molecules which should not be plotted. All atoms of these molecules are invisible. However, their coordinates are still taken into account for the overall layout.
symbolfontsize	-1	The font size in points used for atomic symbols and the header and footer lines. If set to a negative value, a suitable point size is automatically chosen.
voronoicolorproperty	*(empty)*	If this atom property is specified, a Voronoi polygon is drawn behind all (or a filtered subset of the) displayed atoms, and the polygon colored with the color taken from this property.
voronoifilter	*(empty)*	A list of names of CACTVS toolkit filters which are applied to select a subset of atoms for Voronoi polygon coloring. The default is an empty list, which corresponds to no filtering except the atom visibility criterion.
wedges	1	If set, wedge bonds will be drawn. If the parameter is unset, bold and dashed lines will be used instead.
width	180	Width of image in pixels

COPYRIGHT