

4D

# Composant 4D SVG

Le SVG (*Scalable Vector Graphics*) est un format de fichier graphique vectoriel bidimensionnel basé sur le XML. 4D comporte un moteur de rendu intégré permettant d'afficher les fichiers SVG.

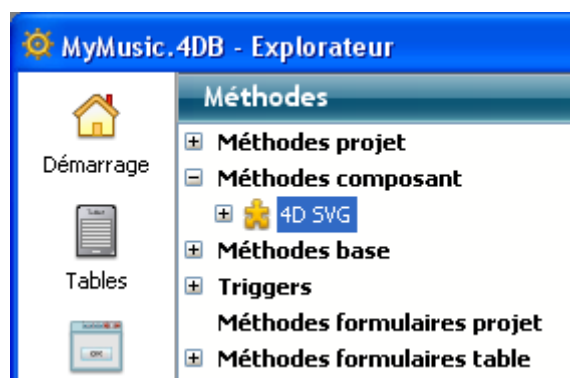
Le langage XML dédié à la manipulation des images SVG est particulièrement riche et étendu. Afin d'en simplifier l'accès et la prise en main, 4D propose le composant 4D SVG. Ce composant comporte de nombreuses commandes permettant la création et la manipulation d'objets graphiques usuels. Le but de cette bibliothèque n'est pas d'être exhaustif mais de répondre aux besoins les plus courants des développeurs 4D. A noter que tous les besoins spécifiques supplémentaires pourront être traités avec les commandes XML de 4D.

## Installation et mise en œuvre

Le Composant 4D SVG doit être installé au minimum dans 4D v11 SQL release 3 (version 11.3).

Comme tout composant 4D, le Composant 4D SVG s'installe par la copie du dossier du composant (**4D SVG.4dbase**) dans le dossier **Components** de la base. Le dossier Components de la base doit être situé au même niveau que le fichier de structure. Les composants étant chargés au démarrage, la base ne doit pas être lancée avant la copie complète de tous les éléments.

Si le composant est correctement installé, l'élément **4D SVG** apparaît dans la page Méthodes de la base, rubriques "Méthodes composant" :



Vous pouvez déployer cet élément afin de visualiser l'ensemble des commandes du composant. Les commandes s'utilisent dans l'éditeur de méthodes de 4D comme des commandes 4D ou de plug-in standard.

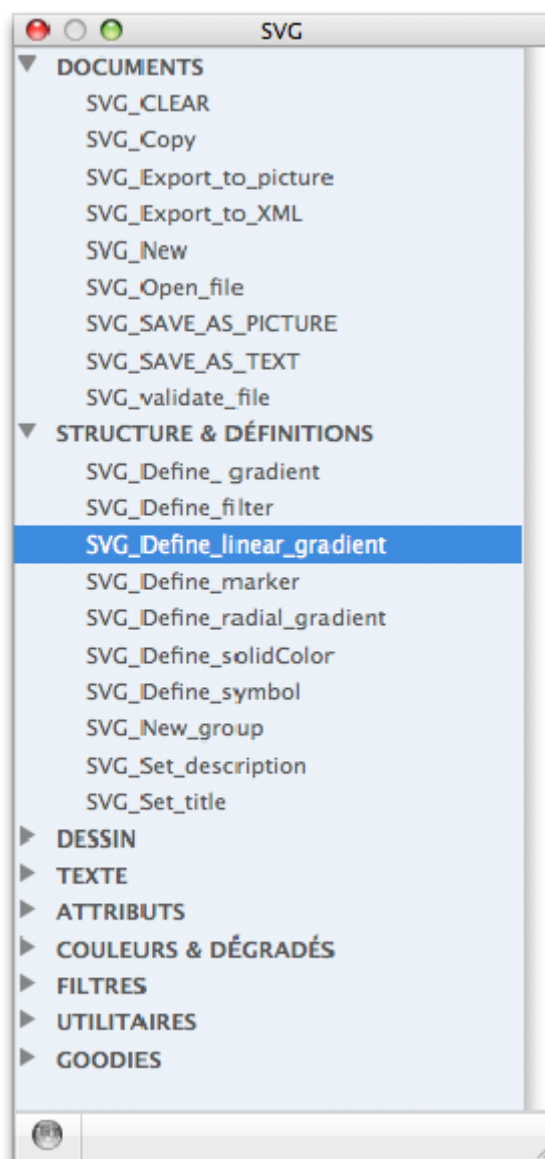
A noter que le Composant 4D SVG vous permet de bénéficier de fenêtres supplémentaires pour la sélection des commandes et le rendu du code SVG. Pour plus d'informations, reportez-vous à la section Outils de développement.

Le composant 4D SVG propose un ensemble d'outils destinés à faciliter la saisie du code et la prévisualisation des graphiques SVG :

- la palette de syntaxe
- la palette de couleurs
- le visualisateur SVG

### Palette de syntaxe

La palette de syntaxe liste les commandes du composant 4D SVG regroupées par thèmes :



La palette permet d'insérer les commandes du composant dans l'éditeur de méthodes par simple glissé-déposer. La commande est alors collée dans la méthode avec ses paramètres. Les paramètres optionnels sont préfixés d'un trait de soulignement.

Pour afficher la palette de syntaxe, vous pouvez soit :

- exécuter la méthode `SVGTool_Display_syntax`,
- cliquer sur le bouton **SVG** et choisir la commande **Syntaxe du composant SVG** dans la palette du composant 4D Pop si vous l'utilisez (cf. ci-dessous).

### Palette de couleurs

La palette de couleurs affiche le nom et un échantillon de chaque couleur définie dans la norme SVG, ainsi qu'un curseur permettant de faire varier le taux d'opacité :



Vous pouvez utiliser cette palette pour insérer par glisser-déposer une référence de couleur SVG dans l'éditeur de méthodes de 4D. La couleur est insérée sous forme de chaîne incluant éventuellement le taux d'opacité (par exemple "lavender:30" pour la couleur lavande et une opacité de 30 %). Pour plus d'informations sur les références de couleurs, reportez-vous à la section Couleurs SVG.

Vous pouvez également glisser-déposer une couleur dans l'éditeur de formulaires de 4D. Cette action crée un carré de couleur sous forme d'image SVG statique.

Pour afficher la palette de couleurs, il vous suffit d'exécuter la méthode `SVGTool_Display_colors`.

## Visualisateur SVG

4D SVG propose un visualisateur SVG, particulièrement utile en phase de développement :

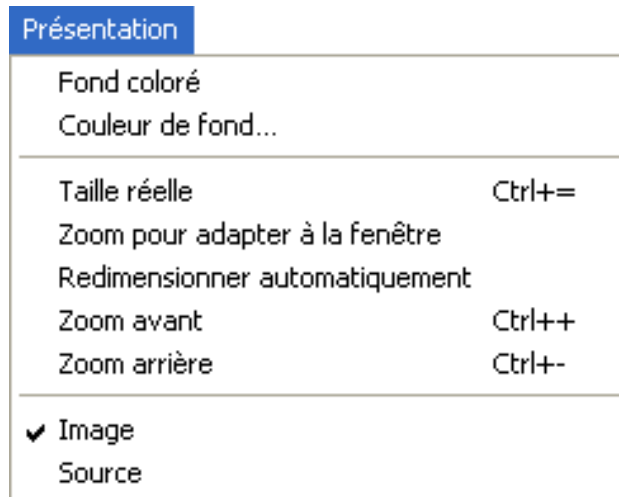


La fenêtre du visualisateur comporte deux pages, accessibles via les boutons **Image** et **Source** ou le menu **Présentation** :

- **Image** : cette page présente une zone de visualisation dans laquelle vous pouvez glisser-déposer ou ouvrir un fichier image SVG (via le menu **Fichier**). Vous pouvez également y afficher une référence SVG valide à l'aide de la commande `SVGTool_SHOW_IN_VIEWER`.

- **Source** : cette page permet de visualiser le code XML associé à l'image. Vous pouvez sélectionner et copier le code, mais vous ne pouvez pas le modifier.

Lorsque la fenêtre est au premier plan, vous pouvez modifier plusieurs options d'affichage et enregistrer le fichier image sur disque via le menu **Présentation** :



**Note :** La page "Image" dispose d'un menu contextuel standard.

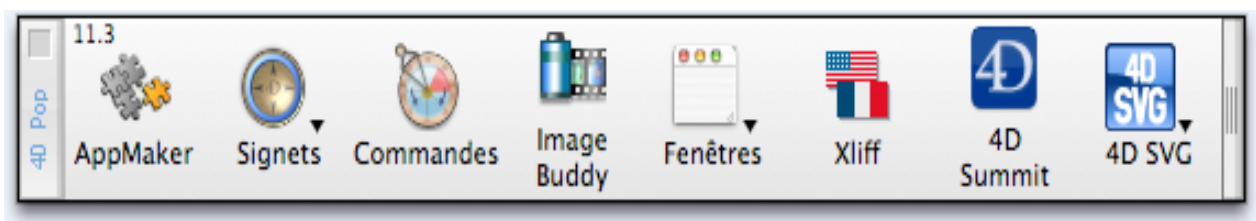
Pour afficher la fenêtre du visualisateur, vous pouvez soit :

- exécuter la méthode `SVGTool_Display_viewer`. Dans ce cas, la fenêtre s'affiche vide.
- appeler la méthode `SVGTool_SHOW_IN_VIEWER` en lui passant une référence SVG valide afin de prévisualiser l'image référencée (voir la description de la commande)
- cliquer sur le bouton **SVG** et choisir la commande **Visualisateur SVG** dans la palette du composant 4D Pop si vous l'utilisez (cf. ci-dessous).

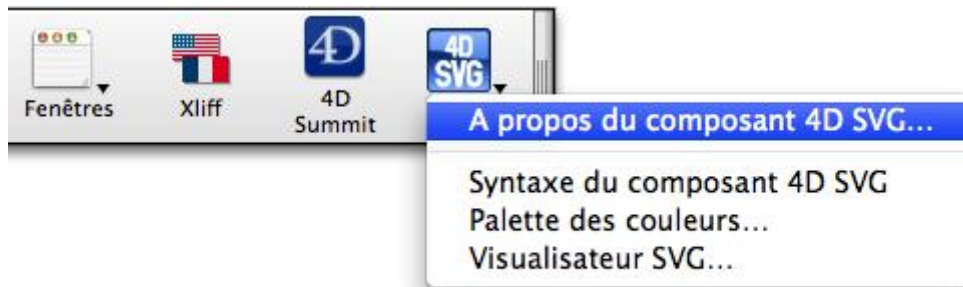
### Intégration à 4D Pop

4D Pop est un ensemble de composants dédiés à la productivité du développeur et regroupés dans une barre d'outils qui s'intègre dans l'environnement de développement de 4D (voir <http://www.4d.fr/products/4dpop.html>).

Lorsque vous utilisez conjointement 4D Pop et 4D SVG, un nouveau bouton est disponible dans la barre d'outils de 4D Pop :



Ce bouton permet l'accès direct aux outils d'aide au développement de 4D SVG :



## Ref\_SVG

La plupart des commandes du composant 4D SVG manipulent les structures SVG via des références de type **Ref\_SVG**.

Une Ref\_SVG est une expression de type Chaîne de 16 caractères. Elle identifie de façon unique une structure SVG chargée en mémoire. Il peut s'agir d'un document SVG chargé via les commandes SVG\_Copy, SVG\_New, SVG\_Open\_picture ou SVG\_Open\_file, ou de toute structure SVG manipulée par programmation (objet, filtre, tracé, etc.).

Une Ref\_SVG est une référence XML. Toutes les références Ref\_SVG peuvent être utilisées comme paramètres refElément des commandes XML DOM de 4D.

Une fois que vous n'en avez plus besoin, n'oubliez pas d'appeler la commande SVG\_CLEAR avec les références Ref\_SVG afin de libérer la mémoire.

## Paramètres optionnels

Sauf mention contraire, les arguments numériques optionnels sont ignorés si leur valeur est égale à -1 et les arguments texte sont ignorés si la chaîne passée est vide.

## Coordonnées

Sauf mention contraire, les paramètres de position ( $x$ ,  $y$ ) et de dimensions (largeur, hauteur, rayon) sont attendus dans le système de coordonnées utilisateur courant.

SVG\_GET\_ATTRIBUTES (objetSVG; pointTabNoms; pointTabValeurs)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence SVG
pointTabNoms	Pointeur	→ Tableau chaîne des libellés d'attributs
pointTabValeurs	Pointeur	→ Tableau chaîne des valeurs d'attributs

## Description

La commande `SVG_GET_ATTRIBUTES` remplit les tableaux pointés par `pointTabNoms` et `pointTabValeurs` respectivement des noms et des valeurs des attributs de l'élément dont la référence est passée dans `objetSVG`. Si `objetSVG` n'est pas valide ou si cet attribut n'existe pas, une erreur est générée.

`SVG_Get_ID (objetSVG) → Chaîne`

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'élément SVG
Résultat	Chaîne	← Nom de l'élément

## Description

La commande `SVG_Get_ID` retourne la valeur de l'attribut 'id' de l'élément dont la référence est passée en paramètre.

Si `objetSVG` n'est pas valide ou si cet attribut n'existe pas, une erreur est générée.

`SVG_SET_ATTRIBUTES (objetSVG; nomAttribut; valeurAttribut{; nomAttribut2; valeurAttribut2; ...; nomAttributN; valeurAttributN})`

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
nomAttribut	Chaîne	→ Nom de l'attribut à fixer
valeurAttribut	Chaîne	→ Valeur de l'attribut

## Description

La commande `SVG_SET_ATTRIBUTES` permet d'affecter un ou plusieurs attribut(s) personnalisé(s) à un objet SVG de référence `objetSVG`. Si le ou les attribut(s) existaient déjà, leurs valeurs sont remplacées par celles passées en paramètre.

Les attributs et leurs valeurs sont passés par couples de paramètres.

## Exemple

```
$svg:=SVG New
$object:=SVG New rect ($svg; 10; 10; 200; 200; 0; 0; "black";"white"; 2)
SVG\_SET\_ATTRIBUTES($object; "style"; "fill:red; stroke:blue; stroke-
width:3")
```

`SVG_SET_ATTRIBUTES_BY_ARRAYS (objetSVG; pointTabNoms; pointTabValeurs)`

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
pointTabNoms	Pointeur	→ Noms des attributs
pointTabValeurs	Pointeur	→ Valeurs synchronisées des attributs



## Description

La commande `SVG_SET_ATTRIBUTES_BY_ARRAYS` permet d'affecter un ou plusieurs attribut(s) personnalisé(s) à un objet SVG de référence `objetSVG`. Si le ou les attribut(s) existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres.

Les attributs et leurs valeurs sont passés par l'intermédiaire de deux tableaux, sur lesquels pointent `pointTabNoms` et `pointTabValeurs`.

## Exemple

```
$svg:=SVG_New
$object:=SVG_New_rect ($svg; 10; 10; 200; 200; 0; 0; "black";"white"; 2)
TABLEAU TEXTE($attributes; 0)
TABLEAU TEXTE ($values; 0)
AJOUTER A TABLEAU($attributes; "fill")
AJOUTER A TABLEAU($values; "red")
AJOUTER A TABLEAU($attributes; "stroke")
AJOUTER A TABLEAU($values; "blue")
AJOUTER A TABLEAU($attributes; "stroke-width")
AJOUTER A TABLEAU($values; "3")
SVG_SET_ATTRIBUTES_BY_ARRAYS ($object; -> $attributes; -> $values)
```

`SVG_SET_DIMENSIONS (objetSVG; largeur{; hauteur{; unité} )`

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
largeur	Numérique →	Dimension sur l'axe x
hauteur	Numérique →	Dimension sur l'axe y
unité	Chaîne →	Unité

## Description

La commande `SVG_SET_DIMENSIONS` permet de fixer les dimensions de l'objet SVG de référence `objetSVG`.

Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres.

Si le paramètre `unité` est passé, il sera utilisé. Les valeurs attendues sont : *px*, *pt*, *pc*, *cm*, *mm* ou *in*. Une valeur d'unité incorrecte génère une erreur. Si le paramètre est omis, les valeurs de `largeur` et `hauteur` sont attendues dans le système de coordonnées utilisateur.

## Exemple

```
$svg :=SVG New `Créer un nouveau document
$object:=SVG New rect ($svg; 10; 10; 200; 200; 0; 0; "black";"white"; 2)
SVG_SET_DIMENSIONS ($object; -1; 400) `Nouvelle hauteur
```

SVG\_SET\_FILL\_BRUSH (objetSVG; couleur)

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
couleur	Chaîne →	Expression couleur

## Description

La commande `SVG_SET_FILL_BRUSH` permet de fixer la couleur de remplissage de l'objet SVG de référence `objetSVG`. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Pour plus d'informations sur les références de couleurs, reportez-vous à la section "[Couleurs SVG](#)".

## Exemple

```
$svg:=SVG New
$object:=SVG New rect ($svg; 10; 10; 200; 200; 0; 0; "black";"white"; 2)
SVG\_SET\_FILL\_BRUSH ($object; "blue")
```

SVG\_SET\_FILTER (objetSVG; url)

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'élément SVG
url	Chaîne →	Nom du filtre

## Description

La commande `SVG_SET_FILTER` permet d'associer un filtre à l'objet de référence `objetSVG`. Si `objetSVG` n'est pas une référence valide, une erreur est générée. Si l'attribut existait déjà, sa valeur est remplacée.

Le paramètre `url` est le nom du filtre à utiliser, tel que défini avec la commande [SVG Define filter](#). Si ce nom n'existe pas, une erreur est générée.

## Exemple

Reportez-vous à l'exemple de la commande [SVG Define filter](#).

SVG\_SET\_ID (objetSVG; id)

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
id	Chaîne →	Identifiant à attribuer à l'objet

## Description

La commande `SVG_SET_ID` permet de fixer la propriété 'ID' de l'objet SVG de référence `objetSVG`. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

L'ID d'un objet est utilisée pour référencer un objet. Cette référence peut ensuite être retrouvée grâce à la commande `SVG_Get_ID`. L'ID est également utilisée par la commande de 4D SVG Chercher ID element par coordonnees (voir la documentation de 4D).

## Exemple

```
$svg:=SVG_New
$object:=SVG_New_rect ($svg; 10; 10; 200; 200; 0; 0; "black"; "white";
2)
SVG_SET_ID ($object; "bordure")
```

`SVG_SET_MARKER (objetSVG; url{; position})`

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'élément SVG
url	Chaîne →	Nom du marqueur
position	Chaîne →	Position du marqueur

## Description

La commande `SVG_SET_MARKER` permet d'associer un marqueur à l'objet de référence `objetSVG`. Si `objetSVG` n'est pas la référence d'un élément 'line', 'path', 'polyline' ou 'polygon', une erreur est générée.

Si l'attribut existait déjà, sa valeur est remplacée.

Le paramètre `url` est le nom d'un élément marqueur à utiliser tel que défini avec la commande [SVG\\_Define\\_marker](#). Si ce nom n'existe pas, une erreur est générée.

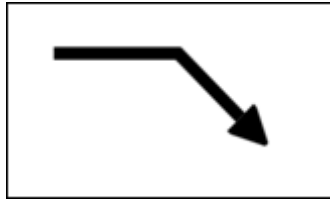
Le paramètre optionnel `position` permet de fixer la position du marqueur par rapport à l'objet. Il est possible de placer un marqueur différent ou non au début, à la fin ou à tous les autres sommets d'un tracé. Les valeurs peuvent être :

- `start` pour placer un marqueur en début de tracé
- `end` pour placer un marqueur à la fin du tracé
- `middle` pour placer un marqueur à tous les sommets autres que le début et la fin.
- `all` pour placer un marqueur sur tous les sommets d'un tracé.

Si ce paramètre est omis, le marqueur est placé à la fin du tracé.

## Exemples

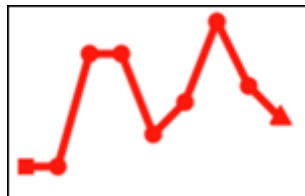
### 1. Dessiner une flèche :



```
$SVG:= SVG_New
  `Définir la flèche
$arrow:=SVG_Define_marker ($SVG;"fleche";0;5;4;3;-1)
SVG_SET_VIEWBOX ($arrow;0;0;10;10)
$path:=SVG_New_path ($arrow;0;0)
SVG_SET_FILL_BRUSH ($path;"black")
SVG_PATH_LINE_TO ($path;10;5)
SVG_PATH_LINE_TO ($path;0;10)
SVG_PATH_CLOSE ($path)

$line:=SVG_New_path ($SVG;100;75)
SVG_SET_STROKE_WIDTH ($line;10)
SVG_PATH_LINE_TO ($line;200;75)
SVG_PATH_LINE_TO ($line;250;125)
  `Mettre la flèche à la fin du tracé
SVG_SET_MARKER ($line;"fleche")
```

### 2. Dessiner un diagramme avec des marqueurs différents au début et à la fin :



```
$SVG:= SVG_New
SVG_SET_DEFAULT_BRUSHES ("red";"red")

  `Définir un cercle pour marquer les points
$point:=SVG_Define_marker ($SVG;"pointMarker";2;2;3;3)
SVG_SET_VIEWBOX ($point;0;0;4;4)
SVG_New_circle ($point;2;2;1)

  `Définir un carré pour le point de début
$start:=SVG_Define_marker ($SVG;"startMarker";1;1;2;2)
SVG_New_rect ($start;0;0;2;2)

  Définir un triangle pour le point de fin
$end:=SVG_Define_marker ($SVG;"endMarker";5;5;3;3;60)
SVG_SET_VIEWBOX ($end;0;0;10;10)
SVG_New_regular_polygon ($end;10;3)

TABLEAU ENTIER LONG($tX;0)
TABLEAU ENTIER LONG ($tY;0)
  `Axe des x
Boucle ($Lon_i;0;200;20)
```

```

    AJOUTER A TABLEAU($tX;$Lon_i+10)
Fin de boucle
`Données
AJOUTER A TABLEAU($tY;100)
AJOUTER A TABLEAU($tY;100)
AJOUTER A TABLEAU($tY;30)
AJOUTER A TABLEAU($tY;30)
AJOUTER A TABLEAU($tY;80)
AJOUTER A TABLEAU($tY;60)
AJOUTER A TABLEAU($tY;10)
AJOUTER A TABLEAU($tY;40)
AJOUTER A TABLEAU($tY;50)
AJOUTER A TABLEAU($tY;70)
$line:=SVG_New_polyline_by_arrays ($SVG;->$tX;->$tY;"red";"none";5)
`Disposer les marqueurs :
SVG_SET_MARKER ($line;"startMarker";"start")
SVG_SET_MARKER ($line;"pointMarker";"middle")
SVG_SET_MARKER ($line;"endMarker";"end")

```

**SVG\_SET\_OPACITY** (objetSVG; opacitéFond{; opacitéTrait})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
opacitéFond	Entier long →	Opacité (%)
opacitéTrait	Entier long →	Opacité (%)

## Description

La commande **SVG\_SET\_OPACITY** permet de fixer l'opacité du remplissage et du trait de l'objet de référence `objetSVG`.

Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres.

Les valeurs attendues sont comprises entre 0 et 100.

## Exemple

```

$svg :=SVG_New `Créer un nouveau document
$object:=SVG_New_rect ($svg ;10;10;200;100;0;0;"red";"blue")
SVG_SET_OPACITY ($object; -1; 50) `Fixer l'opacité du trait à 50%

```

**SVG\_SET\_ROUNDING\_RECT** (objetSVG; rayonX{; rayonY})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un rectangle SVG
rayonX	Numérique →	Rayon sur l'axe x
rayonY	Numérique →	Rayon sur l'axe y

## Description

La commande `SVG_SET_ROUNDING_RECT` permet de fixer les rayons de l'ellipse utilisée pour l'arrondi des coins d'un rectangle de référence `objetSVG`. Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres. Si `objetSVG` n'est pas la référence d'un rectangle, une erreur est générée.

Les valeurs sont attendues dans le système de coordonnées utilisateur.

## Exemple

```
$svg :=SVG_New ` Créer un nouveau document
$object:=SVG_New_rect ($svg ;10;10;200;100)
SVG_SET_ROUNDING_RECT ($object; 20) `Arrondir les angles
```

`SVG_SET_STROKE_BRUSH` (`objetSVG`; couleur)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
couleur	Chaîne	→ Expression couleur

## Description

La commande `SVG_SET_STROKE_BRUSH` permet de fixer la couleur utilisée pour les traits de l'objet SVG de référence `objetSVG`. Si cet attribut existait déjà sa valeur, est remplacée par celle passée en paramètre.

Pour plus d'informations sur les couleurs, reportez-vous à la section "[Couleurs SVG](#)".

## Exemple

```
$svg:=SVG_New
$object:=SVG_New_rect ($svg; 10; 10; 200; 200; 0; 0; "black";"white"; 2)
SVG_SET_STROKE_BRUSH ($object; "red")
```

`SVG_SET_STROKE_LINECAP` (`objetSVG`; mode)

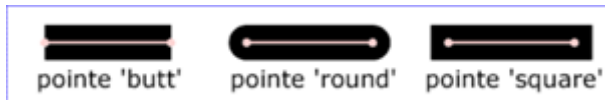
Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
mode	Chaîne	→ Mode de rendu

## Description

La commande `SVG_SET_STROKE_LINECAP` permet de spécifier la forme des extrémités des tracés de l'objet SVG de référence `objetSVG`. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Le paramètre `mode` doit contenir l'une des chaînes suivantes, gérées par le SVG :

- `butt` (défaut) : standard
- `round` : arrondi
- `square` : carré
- `inherit` : hériter de l'objet parent



Si le paramètre `mode` contient une autre valeur, une erreur est générée.

`SVG_SET_STROKE_LINEJOIN` (objetSVG; mode)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
mode	Chaîne	→ Mode de rendu

## Description

La commande `SVG_SET_STROKE_LINEJOIN` permet de spécifier la forme des sommets des tracés de l'objet SVG de référence `objetSVG`. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Le paramètre `mode` doit contenir l'une des chaînes suivantes, gérées par le SVG :

- `miter` (défaut) : standard
- `round` : arrondi
- `bevel` : biseau
- `inherit` : hériter de l'objet parent



Si le paramètre `mode` contient une autre valeur, une erreur est générée.

`SVG_SET_STROKE_WIDTH` (objetSVG; tailleDuCrayon)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
tailleDuCrayon	Entier long	→ Epaisseur du tracé

## Description

La commande `SVG_SET_STROKE_WIDTH` permet de fixer l'épaisseur des lignes de l'objet SVG de référence `objetSVG`. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

La valeur `tailleDuCrayon` est attendue dans le système de coordonnées utilisateur.

## Exemple

```
$svg :=SVG New
SVG\_SET\_STROKE\_WIDTH (SVG New rect ($svg; 10; 10; 200; 200; 0; 0;
"black"; "white"; 2); 10)
```

`SVG_SET_TRANSFORM_FLIP` (objetSVG; horizontal{; vertical})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
horizontal	Booléen	→ Miroir horizontal
vertical	Booléen	→ Miroir vertical

## Description

La commande `SVG_SET_TRANSFORM_FLIP` permet d'appliquer un miroir horizontal et/ou vertical à l'objet SVG de référence `objetSVG`.

Si le paramètre `horizontal` est égal à `Vrai`, un miroir horizontal est appliqué.

Si le paramètre `vertical` est égal à `Vrai`, un miroir vertical est appliqué

## Exemple

Effet de miroir sur un objet texte :



```
svgRef := SVG_New
SVG_SET_VIEWBOX (svgRef; 0; 0; 400; 200)
$txt:=SVG_New_text (svgRef; "4D"; 10; 0; ""; 96)
```



```

SVG_SET_FONT_COLOR ($tx;"blue") `Changer la couleur

`Effet :
$tx:=SVG_New_text (svgRef;"4D";10;0;"";96) `Reprendre le même texte
SVG_SET_FONT_COLOR ($tx;"lightblue") `Changer la couleur
SVG_SET_TRANSFORM_FLIP ($tx;Vrai) `Appliquer un miroir vertical
SVG_SET_TRANSFORM_SKEW ($tx;-10) `Inclinaison
SVG_SET_TRANSFORM_TRANSLATE ($tx;-17;-193) `Repositionner

```

## Référence

SVG\_SET\_TRANSFORM\_MATRIX (objetSVG; a; b{; c; d{; e; f}})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
a	Numérique →	Élément a de la matrice de transformation
b	Numérique →	Élément b de la matrice de transformation
c	Numérique →	Élément c de la matrice de transformation
d	Numérique →	Élément d de la matrice de transformation
e	Numérique →	Élément e de la matrice de transformation
f	Numérique →	Élément f de la matrice de transformation

## Description

La commande `SVG_SET_TRANSFORM_MATRIX` applique une transformation matricielle à l'objet SVG de référence `objetSVG`.

Ce type de transformation permet de combiner des transformations telles que, par exemple, une rotation et une translation.

## Exemple

Writing with SVG is easy

```

SVG_SET_TRANSFORM_MATRIX($ID;0,707;-0,707;0,707;0,707;255,03;111,21)
`Equivaut à appliquer les 3 transformations suivantes :
SVG_SET_TRANSFORM_TRANSLATE ($ID;50;90)
SVG_SET_TRANSFORM_ROTATE ($ID;-45)
SVG_SET_TRANSFORM_TRANSLATE ($ID;130;160)

```

SVG\_SET\_TRANSFORM\_ROTATE (objetSVG; angle{; x; y})

Paramètre	Type	Description
-----------	------	-------------

objetSVG	Ref_SVG	→	Référence d'un élément SVG
angle	Numérique	→	Angle de rotation
x	Numérique	→	Coordonnée du centre de rotation sur l'axe x
y	Numérique	→	Coordonnée du centre de rotation sur l'axe y

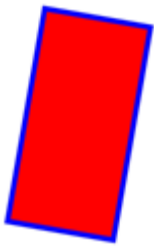
## Description

La commande `SVG_SET_TRANSFORM_ROTATE` applique une rotation de la valeur `angle` à l'objet SVG de référence `objetSVG`.

L'angle de rotation est attendu en degrés, la rotation s'effectue dans le sens horaire.

Si les paramètres optionnels `x` et `y` ne sont pas passés, la rotation s'effectue par rapport à l'origine du système de coordonnées utilisateur courant. Si ces paramètres sont passés, la rotation s'effectue par rapport au point de coordonnées (`x`, `y`).

## Exemple



```

svgRef := SVG_New
  `Dessiner un rectangle rouge avec une bordure bleue
$rec:=SVG_New_rect ($svg;150;50;200;400;0;0;"blue";"red";10)
  `Appliquer une rotation de 10° dans le sens horaire par rapport au
centre
  SVG_SET_TRANSFORM_ROTATE ($rec;370;175;225)

```

`SVG_SET_TRANSFORM_SCALE (objetSVG; échelleX{; échelleY})`

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
échelleX	Numérique →	Valeur sur l'axe x
échelleY	Numérique →	Valeur sur l'axe y

## Description

La commande `SVG_SET_TRANSFORM_SCALE` applique un changement d'échelle horizontale et/ou verticale à l'objet SVG de référence `objetSVG`.

Si la valeur `échelleX` est non nulle, l'objet est agrandi (valeur >1) ou diminué (valeur < 1) horizontalement du nombre d'unités passé. La valeur 1 équivaut à ne pas changer l'échelle de l'objet.

Si le paramètre `échelleY` est passé, l'objet est agrandi (valeur >1) ou diminué (valeur < 1) verticalement du nombre d'unités passé. La valeur 1 équivaut à ne pas changer l'échelle de l'objet. Si ce paramètre est omis, sa valeur est supposée égale à `échelleX`.

### Exemple



```
svgRef := SVG_New
$Text:=SVG_New_text ($SVG;"Hello world!";5)
SVG_SET_TRANSFORM_SCALE ($Text;3;12) `Zoom x*3 y*12
```

`SVG_SET_TRANSFORM_SKEW` (objetSVG; horizontal{; vertical})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
horizontal	Numérique →	Valeur de l'inclinaison le long de l'axe x
vertical	Numérique →	Valeur de l'inclinaison le long de l'axe y

### Description

La commande `SVG_SET_TRANSFORM_SKEW` spécifie une inclinaison horizontale et/ou verticale de l'objet SVG de référence `objetSVG`.

Si la valeur de `horizontal` est non nulle, l'objet sera incliné horizontalement du nombre d'unités passé, sinon elle est ignorée.

Si la valeur de `vertical` est non nulle, l'objet sera incliné verticalement du nombre d'unité passé.

### Exemple



```
$svg := SVG_New
`Dessin du fond
SVG_New_rect ($svg; 0; 0; 270; 160; 10; 10; "black"; "gray")
`Placer le texte...
```

```

$tx:=SVG_New_text ($svg;"Hello world!"; 100; 5; ""; 48)
  `en blanc
SVG_SET_FONT_COLOR ($tx; "white")
  `Inclinaison
SVG_SET_TRANSFORM_SKEW ($tx; -50; 10)

```

**SVG\_SET\_TRANSFORM\_TRANSLATE** (objetSVG; x{; y})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
x	Numérique →	Coordonnée sur l'axe x
y	Numérique →	Coordonnée sur l'axe y

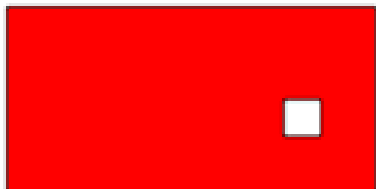
## Description

La commande **SVG\_SET\_TRANSFORM\_TRANSLATE** spécifie une translation horizontale et/ou verticale de l'objet SVG de référence `objetSVG`.

Si la valeur `x` est non nulle, l'objet sera déplacé horizontalement du nombre d'unités passé, sinon elle est ignorée.

Si le paramètre `y` est fourni, l'objet sera déplacé verticalement du nombre d'unité passé.

## Exemple



```

svgRef := SVG_New
  `Dessiner un rectangle rouge
$Object:=SVG_New_rect (svgRef;0;0;200;100;0;0;"black";"red")
  `Dessiner un carré en 0,0
$Object:=SVG_New_rect (svgRef;0;0;20;20)
  `Déplacer le carré en 150,50
SVG_SET_TRANSFORM_TRANSLATE ($Object;150;50)

```

**SVG\_SET\_VIEWBOX** (objetSVG; x; y; largeur; hauteur{; mode})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
x	Entier long →	Position X du rectangle de visualisation
y	Entier long →	Position Y du rectangle de visualisation
largeur	Entier long →	Largeur du rectangle de visualisation
hauteur	Entier long →	Hauteur du rectangle de visualisation
mode	Texte →	Adaptation au rectangle de visualisation

## Description

La commande `SVG_SET_VIEWBOX` permet de définir le rectangle de visualisation de l'objet SVG de référence `objetSVG`. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Les valeurs sont attendues dans le système de coordonnées utilisateur.

Le paramètre optionnel `mode` permet d'indiquer si le graphique doit s'adapter, et comment, à la taille du rectangle de visualisation. La valeur attendue pour `mode` doit être une de celles reconnues par le SVG : 'none', 'xMinYMin', 'xMidYMin', 'xMaxYMin', 'xMinYMid', 'xMidYMid', 'xMaxYMid', 'xMinYMax', 'xMidYMax', 'xMaxYMax' ou 'true' (équivalent à `xMidYMid`).

## Exemple

```
` Créer un document SVG de 4x8cm
$svg:=SVG_New
SVG_SET_DIMENSIONS($SVG;4;8;"cm")
Déclarer le système de coordonnées utilisateur, ici 1cm = 50 points
utilisateur
SVG_SET_VIEWBOX ($svg; 0; 0; 1000; 2000; "true")
```

`SVG_SET_VIEWPORT_FILL (objetSVG{; couleur{; opacité} })`

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'élément SVG
couleur	Chaîne →	Couleur de remplissage
opacité	Entier long →	Pourcentage d'opacité

## Description

La commande `SVG_SET_VIEWPORT_FILL` permet de fixer la couleur de fond d'un document SVG de référence `objetSVG`.

Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre. Si `objetSVG` est un élément SVG qui n'accepte pas cet attribut, une erreur est générée.

Le paramètre optionnel `couleur` indique la couleur à utiliser pour le fond de l'image. Si ce paramètre est omis ou contient une chaîne vide, c'est le blanc qui est utilisé. Pour plus d'informations sur les couleurs reportez-vous à la section [Couleurs SVG](#).

Le paramètre optionnel `opacité` permet de préciser la valeur en pourcentage de l'opacité appliqué à ce remplissage. Si ce paramètre est omis et si aucune opacité n'était définie pour le document, la valeur 100% est utilisée.

SVG\_SET\_VISIBILITY (objetSVG{; cacher})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
cacher	Booléen	→ Vrai = montrer, Faux = cacher

## Description

La commande `SVG_SET_VISIBILITY` cache ou montre l'objet SVG de référence `objetSVG`. Si `objetSVG` n'est pas la référence d'un objet pouvant être caché, une erreur est générée.

Si le paramètre optionnel `cacher` est égal à `Vrai` ou omis, l'objet est montré. S'il est égal à `Faux`, l'objet est caché.

## Exemple

```
$svg :=SVG_New
$object:=SVG_New_rect ($svg; 10; 10; 200; 200; 0; 0; "black";" white";
2)
SVG_SET_VISIBILITY ($object; Faux) `L'objet est décrit mais ne sera pas
rendu.
```

SVG\_SET\_XY (objetSVG; x{; y})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
x	Numérique	→ Coordonnée sur l'axe x
y	Numérique	→ Coordonnée sur l'axe y

## Description

La commande `SVG_SET_XY` permet de fixer les coordonnées du coin supérieur gauche de la région rectangulaire dans laquelle l'objet SVG de référence `objetSVG` se place.

Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres. Si `objetSVG` est un élément SVG qui n'accepte pas cet attribut, une erreur est générée.

Les valeurs sont attendues dans le système de coordonnées utilisateur.

## Exemple

```
$svg :=SVG_New `Créer un nouveau document
$object:=SVG_New_image ($svg;"#Pictures/logo4D.png") `Placer le logo
SVG_SET_XY ($object; 10; 40) `Modifier la position de l'image
```

## Définitions de couleurs

Le SVG reconnaît toutes les syntaxes alternatives pour les couleurs définies dans la norme CSS2. Les commandes du composant 4D SVG prennent en charge toutes ces syntaxes.

Une couleur peut être exprimée sous l'une des formes suivantes :

- *Format RVB*

<b>Format</b>	<b>Exemple</b>
#rvb	#f00
#rrvvbb	#ff0000
rvb(r,v,b)	rvb(255, 0, 0)
	rvb(100%, 0%, 0%)

- *Format mot-clé "couleur"*

Le SVG admet une liste étendue de mots-clés de noms de couleur, par exemple "red".

La liste des mots-clés ainsi que leur correspondance RVB figure dans l'Annexe A, Tableau de couleurs. Vous pouvez également visualiser cette liste et insérer directement des valeurs de couleurs via la Palette de couleurs de 4D SVG. Pour plus d'informations sur ce point, reportez-vous à la section Outils de développement.

## **Opacité**

Il est possible de spécifier l'opacité dans les expressions couleurs des commandes du composant en utilisant la syntaxe "couleur:opacité" où opacité est un nombre compris entre 0 (pas de couleur) et 100 (couleur complètement opaque). Ainsi "red:50" sera interprété comme un rouge à 50% d'opacité.

## **Dégradés**

Les dégradés sont des transitions progressives de couleur le long d'un vecteur. Ces dégradés sont définis avec les commandes `SVG_Define_linear_gradient` et `SVG_Define_radial_gradient`. Une fois définis, les dégradés sont utilisés par référence en utilisant la syntaxe "url(#NomDuDégradé)".

De même, il est possible de définir une couleur personnalisée associée à une opacité avec la commande [SVG\\_Define\\_solidColor](#).

SVG\_Color\_grey (pourcentage) → Chaîne

<b>Paramètre</b>	<b>Type</b>	<b>Description</b>
pourcentage	Entier	→ Intensité du gris
Résultat	Chaîne	← Chaîne couleur

## Description

La commande `SVG_Color_grey` retourne une chaîne exprimant une couleur grise d'intensité `pourcentage`. La chaîne retournée est de la forme "rgb(rouge, vert, bleue)" où les 3 valeurs sont égales, syntaxe reconnue par les moteurs de rendu SVG.

## Exemple

```
$txtColor:= SVG_Color_grey (60)  
`$txtColor vaut "rgb(102,102,102)"
```