

AJUI Tip

Manuel d'utilisation

1	<i>Introduction.....</i>	4
1.1	Qu'est-ce que AJUI Tip	4
1.2	Composition d'une info-bulle	4
1.2.1	Aperçu technique	5
1.3	Le positionnement de l'info-bulle	5
1.4	Arrondir la flèche d'un Tip	7
1.5	Cycle de vie d'une Info-bulle en 5 phases	8
2	<i>Liste des propriétés</i>	9
2.1	Global.....	9
2.2	Target.....	10
2.3	Container.....	11
2.4	Texte	13
2.5	Arrow	14
2.6	Sous-formulaire	15
2.7	Closebox.....	15
2.8	Debug.....	15
2.9	Animations	16
2.9.1	Blink.....	16
2.9.2	Jump	17
2.10	Fade in and out	18
3	<i>Fonction membre non liée à une propriété</i>	18
4	<i>Méthodes du composant.....</i>	19
5	<i>Prise en main</i>	20
5.1	Installation	20
5.2	Principes d'utilisation.....	20
5.2.1	Prérequis	20
5.2.2	Noms et propriétés réservées.....	20
5.2.3	Mise en place d'un premier Tip	22
5.2.4	Utiliser un formulaire à la place du texte.....	23
5.2.5	Ajouter des animations	23
5.2.6	Bonnes pratiques de stockage des instances.....	25
5.3	Cas spécifiques.....	25
5.3.1	Listbox	25
5.3.2	Multipages.....	26
5.3.3	Sous-formulaire	26
5.3.4	Utilisation de la commande Form de 4D dans un sous-formulaire	27
6	<i>Conclusion</i>	27

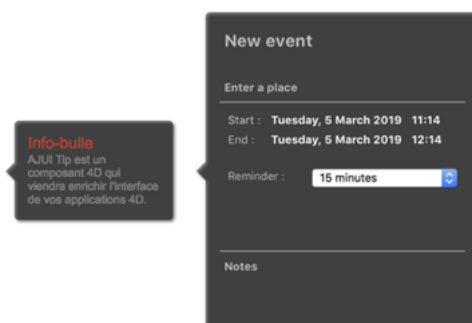
Version Control	Date	Commentaires	Autheur
1.0	15.02.2019	Première version	Gary Criblez
1.1	27.06.2019	Légères corrections	Gary Criblez
1.2	12.07.2019	Rework fade in / fade out et retrait du subform connector.	Gabriel Inzirillo
1.3	19.08.2019	Utilisation de l'objet Form de 4D dans un sous-formulaire.	Gary Criblez
1.4	05.09.2019	Arrondir la flèche d'un Tip	Gary Criblez
1.5	13.12.2019	Modification import et export	Gary Criblez

1 Introduction

1.1 Qu'est-ce que AJUI Tip

AJUI Tip est un composant développé dans le langage 4D. Son utilisation est destinée aux développeurs 4D. Il permet de générer et afficher dynamiquement des info-bulles (Tip) dans le contexte d'un formulaire.

Le composant offre plusieurs méthodes permettant de les personnaliser et de les manipuler via un objet et des fonctions. Ceux-ci peuvent être conservés et réutilisés sous formats JSON en tant que modèle.



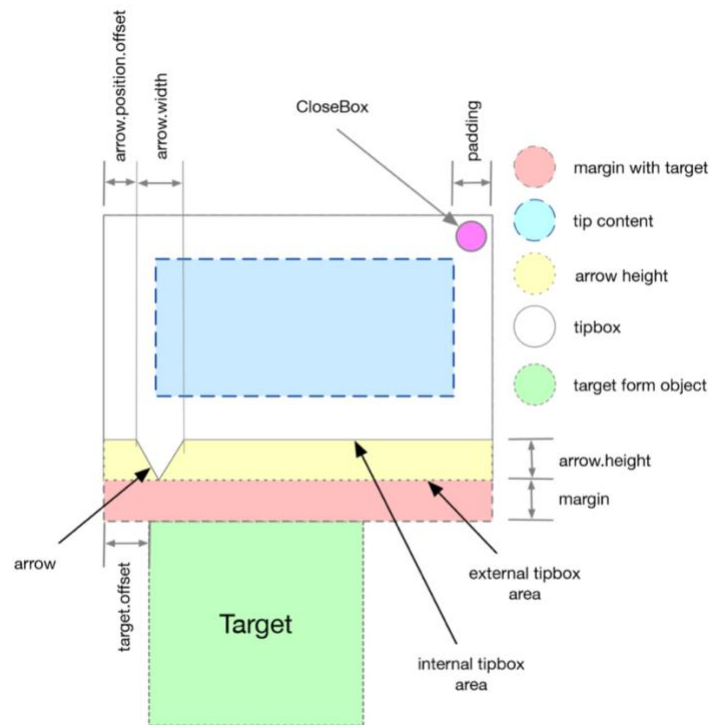
1.2 Composition d'une info-bulle

Fondamentalement l'info-bulle est un formulaire partagé (container) présent dans le composant qui sera affiché via un sous-formulaire dans le formulaire cible de la base principale. Celui-ci contient trois éléments :

- **Une variable image** : Pour construire l'image, le composant s'appuie sur un bon nombre de propriétés. Celles-ci vont permettre de dessiner un objet en langage SVG. Il en découle, un ensemble de calcul basé sur différents critères (taille du contenu, padding, radius,...) qui sera effectué afin de tracer le dessin du contour de l'info-bulle et de sa flèche. Le SVG pourra également intégrer une zone de texte et un bouton de fermeture de l'info-bulle (closebox).
- **Un sous-formulaire** : Celui-ci permet d'intégrer à l'info-bulle un formulaire provenant de la base principale. Le sous-formulaire prendra la place de la zone de texte s'il est défini.
- **Une variable objet** : Permet de conserver les propriétés de l'info-bulle nécessaire lors de diverses opérations du composant.

Cet ensemble forme ce qu'on appelle l'**info-bulle** ou **tipbox** en anglais.

1.2.1 Aperçu technique



1.3 Le positionnement de l'info-bulle

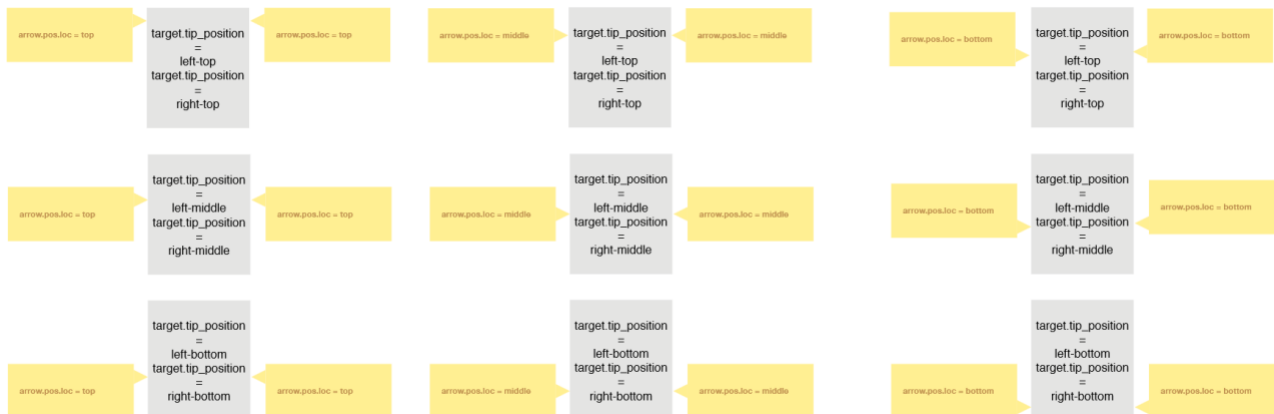
Le positionnement de l'info-bulle peut se faire manuellement en entrant les coordonnées d'un point ou en désignant un objet formulaire qui fera office de cible (target). Le composant se chargera de récupérer les coordonnées dans ce second cas.

Ensuite, il est possible de définir le placement de l'info-bulle et de la flèche qui la compose :

Positions horizontales



Positions verticales

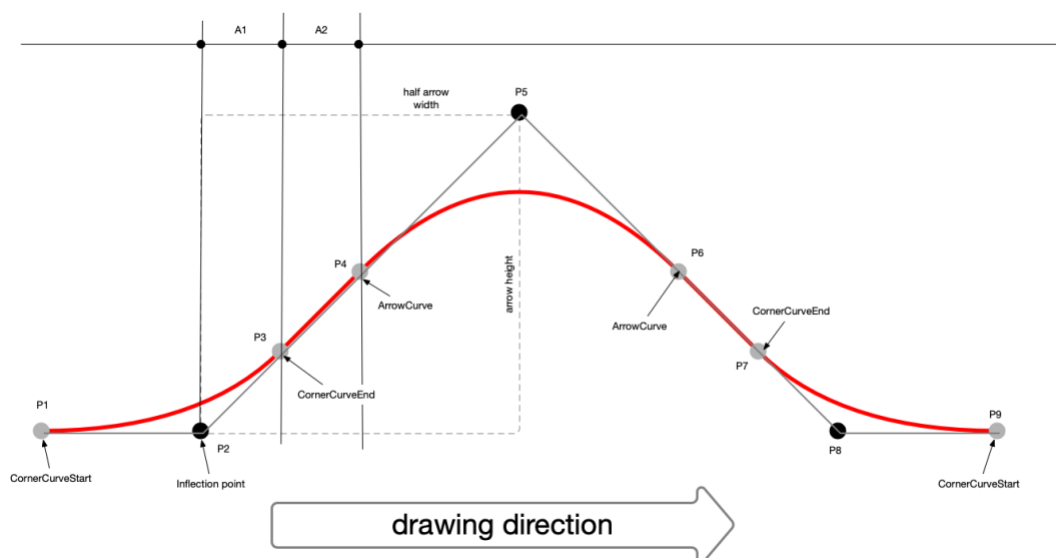


Il est également possible de définir une marge (margin) et/ou un décalage (Offset) par rapport à la cible (target). (Voir le schéma au chapitre 1.2.1 - Aperçu technique)

1.4 Arrondir la flèche d'un Tip

Depuis la version 1.4 d'AJUI Tip, il est possible d'arrondir les trois angles de la flèche d'un Tip. Pour cela, quatre formules sont mises à disposition (voir les propriétés concernant la flèche). Dans ce chapitre, nous allons vous expliquer comment les arrondis sont calculés, afin de mieux comprendre l'utilisation de ces nouvelles formules.

Schéma de calcul des arrondis :



Avant toute chose, il est important de spécifier qu'un Tip est toujours dessiné en partant de la droite depuis son coin supérieur gauche.

La première courbe est calculée à partir du point de départ (P1) jusqu'au point d'arrivée (P3) et en utilisant le point d'inflexion (P2). La deuxième courbe est calculée à partir du point de départ (P4) jusqu'au point d'arrivée (P6) et en utilisant le point d'inflexion (P5). La troisième courbe est calculée à partir du point de départ (P7) jusqu'au point d'arrivée (P9) et en utilisant le point d'inflexion (P8).

Les trois points noirs sur le schéma correspondent aux trois angles de la initial de la flèche. Dans le cas de la fonction d'arrondi, ils vont servir de point d'inflexion pour dessiner les courbes.

Les six points gris sur le schéma sont déterminés grâce à des ratios que vous pouvez affecter pour le dessin de la flèche à l'aide des formules : **CornerCurveStart**, **CornerCurveEnd**, **ArrowCurve**.

Sur quelles données les ratios sont appliqués afin d'obtenir ces six points ?

Les ratios de **CornerCurveEnd** et **ArrowCurve** vont être appliqués sur la hauteur de la flèche et sur la moitié de la largeur de la flèche. Cela va nous permettre d'obtenir les valeurs en X et Y pour la position de ces points (P3, P4, P6 et P7).

Concernant **CornerCurveStart**, le ratio s'applique uniquement par rapport à la hauteur de la flèche. Le résultat de ce calcul sera ensuite soustrait ou additionné par rapport aux points d'inflexion (P2 et P8) sur la largeur.

Le fait d'appliquer ces trois ratios permet de s'assurer la réalisation d'un rendu cohérent de l'arrondi de la flèche. AJUI Tip Lab inclut un "HDI Curves" qui vous permettra de jouer avec ces ratios et fonctions.

1.5 Cycle de vie d'une Info-bulle en 5 phases

Phase 1 : initialisation et paramétrage

La première phase consiste à **définir** les propriétés de l'info-bulle qui va être affichée. Pour cela, le composant offre une méthode générant un objet de base représentant une définition par défaut. Libre ensuite à l'utilisateur d'utiliser les fonctions attachées à l'objet afin de personnaliser ses propriétés à sa guise. A noter que le composant propose d'importer et d'exporter ces objets au format JSON, ceci permet d'utiliser des modèles.

Cette étape est très importante, car toutes les autres étapes se baseront sur ce qui a été défini dans l'objet.

Phase 2 : calculations et génération

La deuxième phase qui est la plus conséquente pour le composant est la création de l'info-bulle basée sur sa définition. Durant cette phase, il va se charger de récupérer les propriétés de l'objet afin d'effectuer une série de calcul et de contrôle afin de générer/affecter les différents éléments qui vont composer cet ensemble qu'on nomme l'**info-bulle** ou **tipbox** (anglais).

Cette phase se termine avec le positionnement de l'info-bulle en fonction de la cible (Target) et des éléments associés qui ont été abordé précédemment.

Phase 3 : Affichage

La troisième phase consiste à rendre visible l'info-bulle à l'utilisateur. Elle se déclenche automatiquement à la fin de l'étape précédente.

Son apparition peut se faire de deux façons différentes. Dans le premier cas tout simplement en passant l'objet d'un état invisible à visible. Dans le second cas, si l'utilisateur a paramétré un effet de fondu à l'affichage (effet de fondu enchainé ou fade in) alors le composant se chargera d'augmenter graduellement son opacité jusqu'à atteindre sa forme finale.

Phase 4 : Durée de vie et animations

Dans la quatrième phase, le composant se charge de la durée d'affichage ou durée de vie et des animations préalablement paramétrées en phase 1. Actuellement deux animations sont disponibles, le saut (jump) et le clignotement (blink). Il n'est pas possible de lancer plusieurs animations simultanément pour une même info-bulle. Le composant se chargera de lancer une seule et unique animation lors de cette phase. La priorité étant le clignotement si les deux animations sont définies et activées. Les animations sont sensibles à la durée de vie de l'objet.

Par défaut, la durée de vie d'une info-bulle est égale à zéro (indéterminé et donc affichage illimité) et donc n'impacte pas le déclenchement d'une animation. Par contre, si le temps calculé d'une animation est supérieur à la durée de vie définie alors, le composant n'aura pas le temps de finir l'animation et l'info-bulle sera fermée.

Phase 5 : Fermeture

La cinquième phase est déclenchée automatiquement et gérée par le composant si une durée de vie est définie. Dans le cas contraire, c'est à l'utilisateur de la déclencher soit par l'utilisation du bouton de fermeture sur l'info-bulle (CloseBox) s'il est actif, soit en déclenchant la méthode du composant qui permet de la cacher. Il est à noter que si d'autres info-bulles sont générées et s'il s'agit d'une même instance alors la fermeture sera faite automatiquement.

Quant à la fermeture en soi, elle s'effectue de la même façon que pour l'affichage mais en sens inverse. C'est-à-dire soit par le passage de visible à invisible de l'objet, soit par en fondu en sorti (fade out), dans ce cas le composant réduit graduellement l'opacité du Tip jusqu'à zéro %.

2 Liste des propriétés

Une info-bulle possède un ensemble de propriétés qui vont permettre de définir sa représentation et ses interactions avec le formulaire hôte. Dans ce chapitre, nous allons passer en revue les différentes propriétés existantes accessibles par une formule faisant office de **Setter** et également de **Getter** si on ne leur passe aucun paramètre. Toutes les formules pourront être appelées au premier niveau de l'objet.

Il est recommandé de ne pas modifier manuellement les propriétés dont les valeurs ne sont pas définies par des formules car elles sont gérées en interne par le composant.

*NB : Dans les sections suivantes de ce chapitre 2 nous serons amenés à utiliser le terme en anglais **Tip** en lieu et place d'**Info-bulle** pour des soucis de cohérences avec les noms définis anglais dans les formules.*

2.1 Global

Propriétés globales se trouvant au deuxième niveau de l'objet : **tip.global**

Nom	Type	Description	Par défaut	Formule
name	string	Nom de l'info-bulle (tip).	Chaîne vide	TipName
tipDuration	longint	Temps d'apparition du Tip (valeur en milliseconde). En cas de valeur zéro, le Tip n'aura pas de limite de temps d'affichage.	0	TipDuration
autoRebuild	booléen	Permet de forcer la reconstruction du Tip automatiquement après chaque utilisation d'une formule (à l'exception des formules liées aux animations). Activer ce mode risque de produire de nombreux appels, il doit être utilisé modérément.	False	AutoRebuild

Global – Formules et leurs paramètres

Nom de formule	Paramètre(s)
TipName (nom)	- Nom du Tip (texte)
TipDuration (durée)	- Durée de vie/affichage du Tip (réel) <i>Valeur exprimée en millisecondes</i>
AutoRebuild (activer)	- Activer l'auto rebuild (booléen)

2.2 Target

Propriétés liées au target se trouvant au deuxième niveau de l'objet : **tip.target**.

Nom	Type	Description	Par défaut	Formule
name	string	Nom de l'objet formulaire cible qui sera lié au Tip.	Chaîne vide	TargetName
coordinate	objet	Cet objet va contenir les valeurs des coordonnées du target (left, top, right et bottom). Les valeurs sont calculées automatiquement si la cible est définie.	0	TargetCoordinates
tipPosition	string	Position du Tip par rapport à la cible (target).	right-middle	TipPosition
margin	longint	Permet de fixer un écart entre le Tip et la cible (target).	0	TargetMargin
offset	longint	Permet de décaler le Tip rapport à la cible (target).	0	TargetOffset

Target – Formules et leurs paramètres

Nom de formule	Paramètre(s)
TargetName (nom)	- Nom de l'objet cible (texte)
TargetCoordinates (gauche { ; haut { ; droit { ; bas }}})	- Coordonnée gauche (longint) - Coordonnée haut (longint) (optionnel) - Coordonnée droite (longint) (optionnel) - Coordonnée bas (longint) (optionnel) <i>Coordonnées exprimées en pixels</i>
TipPosition (position {flèche par défaut })	- Positon Tip (texte) voir ch 1.3 - Applique la position de flèche par défaut (booléen) (optionnel)
TargetMargin (gauche { ; haut { ; droit { ; bas }}})	- Marge gauche (longint) - Marge haut (longint) (optionnel) - Marge droite (longint) (optionnel) - Marge bas (longint) (optionnel) <i>Valeur exprimée en pixels</i>
TargetOffset (décalage)	- Décalage du Tip (longint) <i>Valeur exprimée en pixels</i>

2.3 Container

Propriétés liées au container se trouvant au deuxième niveau de l'objet : **tip.container**.

Nom	Type	Description	Par défaut	Formule - setter
backgroundColor	string	Couleur de fond du container.	yellow	BackgroundColor
backgroundOpacity	longint	Opacité du fond du container (%).	50	BackgroundOpacity
borderColor	string	Couleur de la bordure.	gray	BorderColor
borderOpacity	longint	Opacité des bordures (%).	100	BorderOpacity
borderSize	longint	Taille de la bordure.	1	BorderSize
cornerRadius	longint	Valeur du radius appliqué aux 4 angles du Tip.	5	CornerRadius
padding	longint	Contient le padding a appliqué au Tip.	10	Padding
maxHeight	longint	Permet de définir la hauteur maximum de la zone de texte dans le container. N'est pas pris en compte si maxWidth est supérieur à zéro.	0	TipboxMaxHeight
maxWidth	longint	Permet de définir la largeur maximum de la zone de texte dans le container. A la priorité sur maxHeight pour des raisons pratiques de redimensionnement de la zone.	0	TipboxMaxWidth
shadowOn	booléen	Activer l'effet d'ombre	False	ActivateShadow
shadowOffset	longint	Décalage de l'ombre par rapport à l'info-bulle	2	ShadowOffset

Container – Formules et leurs paramètres

Nom de formule	Paramètre(s)
BackgroundColor (couleur)	- Nom/Code de couleur (texte) <i>Peut-être indifféremment le nom d'une couleur (Ex : "red") ou une valeur en hexa représentant la couleur RGB (Ex : "#FF0000")</i>
BackgroundOpacity (opacité)	- Pourcentage d'opacité (longint) <i>Valeur exprimée en pourcent (0-100)</i>
BorderColor (couleur)	- Nom/Code de couleur (texte) <i>Peut-être indifféremment le nom d'une couleur (ex : "red") ou une valeur haxa représentant la couleur RGB (Ex : "#FF0000")</i>
BorderOpacity (opacité)	- Pourcentage d'opacité (longint) <i>Valeur exprimée en pourcent (0-100)</i>
BorderSize (largeur)	- Largeur de bordure (longint) <i>Valeur exprimée en pixels</i>
CornerRadius (rayon)	- Valeur du rayon (longint) <i>Valeur exprimée en pixels</i>
Padding (marge)	- Taille de la marge intérieur (longint)
TipboxMaxHeight (largeur)	- Hauteur max du conteneur (longint) <i>Valeur exprimée en pixels</i>
TipboxMaxWidht (hauteur)	- Largeur max du conteneur (longint) <i>Valeur exprimée en pixels</i>
ActivateShadow (activer)	- Active l'effet d'ombre (booléen)
ShadowOffset (décalage)	- Valeur du décalage (longint)

2.4 Texte

Propriétés liées au texte affiché dans le container se trouvant au deuxième niveau de l'objet : **tip.text**.

Nom	Type	Description	Par défaut	Formule
value	string	Texte à afficher.	Texte d'exemple stylé	TextLabel
fontColor	string	Couleur du texte	black	TextFontColor
fontName	string	Nom de la police à utiliser. Plusieurs polices peuvent être entrées en les séparant par une virgule sans espace. Le composant recherchera la première police utilisable par rapport à votre OS.	Helvetica, Arial, MS Sans Serif	TextFontName
fontSize	longint	Taille du texte.	11	TextFontSize
opacity	longint	Opacité du texte (%).	100	TextOpacity
rotation	string	Rotation de la zone de texte	Chaîne vide	TextRotation

Texte – Formules et leurs paramètres

Nom de formule	Paramètre(s)
TextLabel (texte)	- Texte du conteneur (texte) <i>La chaîne texte peut contenir du texte stylé</i>
TextFontColor (couleur)	- Nom/code de couleur (texte) <i>Peut-être indifféremment le nom d'une couleur (Ex : "red") ou une valeur en hexa représentant la couleur RGB (Ex : "#FF0000")</i>
TextFontName (police)	- Nom de police (texte)
TextFontSize (taille)	- Taille du texte (longint) <i>Valeur exprimée en pixels</i>
TextOpacity (opacité)	- Pourcentage d'opacité (longint) <i>Valeur exprimée en pourcent (0-100)</i>
TextRotation (orientation)	- Sens de rotation, left ou right (string)

2.5 Arrow

Propriétés liées au dessin de la flèche du Tip se trouvant au deuxième niveau de l'objet : **tip.arrow**.

Nom	Type	Description	Par défaut	Formule
position	string	Position de la flèche au niveau du Tip.	middle	ArrowPosition
height	longint	Taille en hauteur de la flèche.	16	ArrowHeight
width	longint	Taille en largeur de la flèche.	20	ArrowWidth
offset	longint	Permet d'ajouter un décalage de la flèche par rapport à sa position initiale (aucun effet sur les positions center et middle).	10	ArrowOffset
curved	booléen	Permet d'arrondir les angles de la flèche.	false	Curved
arrowCurve	réel	Ratio à appliquer sur l'angle de la pointe de la flèche (voir le chapitre sur l'arrondi de la flèche).	0.9	ArrowCurve
cornerCurveStart	réel	Ratio à appliquer et qui permet de définir le point de départ de l'arrondi concernant les angles de départ et de fin de la flèche (voir le chapitre sur l'arrondi de la flèche).	0.5	CornerCurveStart
cornerCurveEnd	réel	Ratio à appliquer et qui permet de définir le point de fin de l'arrondi concernant les angles de départ et de fin de la flèche (voir le chapitre sur l'arrondi de la flèche).	0.15	CornerCurveEnd

Arrow – Formules et leurs paramètres

Nom de formule	Paramètre(s)
ArrowPosition (position)	- Position de la flèche (texte) voir ch 1.3
ArrowHeight (hauteur)	- Hauteur de la flèche (longint) <i>Valeur exprimée en pixels</i>
ArrowWidth (largeur)	- Largeur de la flèche (longint) <i>Valeur exprimée en pixels</i>
ArrowOffset (décalage)	- Décalage de la flèche (longint) <i>Valeur exprimée en pixels</i>
Curved (arrondir)	- Arrondir la flèche (booléen)
ArrowCurve (ratio)	- Ratio de la pointe de la flèche (réel)
CornerCurveStart (ratio)	- Ratio du point de départ de l'arrondi (réel)
CornerCurveEnd (ratio)	- Ratio du point de fin de l'arrondi (réel)

2.6 Sous-formulaire

Propriétés liées au sous-formulaire du conteneur se trouvant au deuxième niveau de l'objet : **tip.subform**.

Nom	Type	Description	Par défaut	Formule
name	string	Nom du formulaire associé au sous-formulaire du conteneur.	Chaîne vide	SubformName

Sous-formulaire – Formules et leurs paramètres

Nom de formule	Paramètre(s)
SubformName (nom)	- Nom du formulaire de la base Hôte (texte)

2.7 Closebox

Propriétés liées à la closebox se trouvant au deuxième niveau de l'objet : **tip.closeBox**.

Nom	Type	Description	Par défaut	Formule
display	booléen	Affichage de la closebox dans le Tip. Quand cette valeur est passé à vrai, la propriété « onClick » l'est aussi.	False	CloseBoxDisplay
onClick	booléen	Activation de l'événement onClick de la closebox	False	ActivateOnClickCB
size	longint	Taille de la closebox. Comprise entre 2 et 20	5	CloseBoxSize

Closebox – Formules et leurs paramètres

Nom de formule	Paramètre(s)
CloseBoxDisplay (affichage)	- Affichage (booléen)
ActivateOnClickCB (activation)	- Activer l'évènement (booléen)
CloseBoxSize (taille)	- Taille entre 2 et 20 (longint) <i>Valeur exprimée en pixels</i>

2.8 Debug

Propriétés activant des éléments de debug se trouvant au deuxième niveau de l'objet : **tip.debug**.

Nom	Type	Description	Par défaut	Formule
showSVGArea	booléen	Affiche la zone de dessin du Tip	False	non
showSVGViewer	booléen	Lance SVGViewer sur la méthode show	False	non

2.9 Animations

Propriétés liées aux animations se trouvant au deuxième niveau de l'objet : **tip.animation**.

Nom	Type	Description	Par défaut	Formule
isActive	booléen	Active les animations (ne concerne pas le fade in ni le fade out)	True	ActivateAnimation
refresh	longint	Valeur de rafraichissement des animations. Exprimé en nombre de rafraichissements par seconde. Cette valeur s'applique aux animations et pour les effets de Fade In et Fade Out.	20	AnimationRefresh

Animations – Formules et leurs paramètres

Nom de formule	Paramètre(s)
ActivateAnimation (activation)	- Activer les animations (booléen)
AnimationRefresh (rafraichissement)	- Valeur de rafraichissement (longint) <i>Valeur exprimée en taux de rafraichissement par seconde.</i>

2.9.1 Blink

Propriétés liées à l'animation Blink se trouvant au troisième niveau de l'objet : **tip.animation.blink**.

Nom	Type	Description	Par défaut	Formule
number	longint	Nombre de Blink.	0	BlinkNumber
duration	longint	Durée d'un Blink (en millisecondes).	1000	BlinkDuration
on	longint	Répartition du temps d'affichage par rapport à la durée d'un blink (%).	50	BlinkOn

Blink – Formules et leurs paramètres

Nom de formule	Paramètre(s)
BlinkNumber (nombre)	- Nombre d'effet (longint)
BlinkDuration (durée)	- Durée d'un blink (longint) <i>Valeur exprimée en millisecondes</i>
BlinkOn (pourcent)	- Pourcentage (longint) <i>Valeur exprimée en pourcent (0-100)</i>

2.9.2 Jump

Propriétés liées à l'animation Jump se trouvant au troisième niveau de l'objet : **tip.animation.jump**.

Nom	Type	Description	Par défaut	Formule
number	longint	Nombre d'effet de jump.	0	JumpNumber
duration	longint	Durée de l'effet du jump, cela comprend l'aller et le retour (millisecondes).	1000	JumpDuration
offset	longint	Distance à parcourir pour l'effet de jump.	20	JumpOffset
on	longint	Répartition du temps pour le Jump montant par rapport à la durée totale du Jump (%)	50	JumpOn

Jump – Formules et leurs paramètres

Nom de formule	Paramètre(s)
JumpNumber (nombre)	- Nombre d'effet (longint)
JumpDuration (durée)	- Durée d'un effet (longint) <i>Valeur exprimée en millisecondes</i>
JumpOffset (distance)	- Distance de saut (longint) <i>Valeur exprimée en pixels</i>
JumpOn (pourcent)	- Pourcentage de la durée à l'aller (longint) <i>Valeur exprimée en pourcent (0-100)</i>

2.10 Fade in and out

Propriétés liées aux animations de Fade se trouvant au troisième niveau de l'objet : **tip.animation.fade**.

Nom	Type	Description	Par défaut	Formule
in	booléen	Active l'animation fade in	False	ActivateFadeIn
out	booléen	Active l'animation fade out	False	ActivateFadeOut
timer.in	longint	Durée de l'effet de fade in	1000	TimerFadeIn
timer.out	longint	Durée de l'effet de fade out	1000	TimerFadeOut

Fade in and out – Formules et leurs paramètres

Nom de formule	Paramètre(s)
ActivateFadeIn (activer)	- Activer l'effet (booléen)
ActivateFadeOut (activer)	- Activer l'effet (booléen)
TimerFadeIn (durée)	- Durée de l'effet Fade In (longint) <i>Valeur exprimée en millisecondes</i>
TimerFadeOut (durée)	- Durée de l'effet Fade Out (longint) <i>Valeur exprimée en millisecondes</i>

3 Fonction membre non liée à une propriété

Show() et **Hide()** sont deux formules du composant qui ne sont pas directement liées à une propriété.

La méthode **Show()** permet de lancer la génération du Tip et de le rendre visible.

La méthode **Hide()** permet de rendre le Tip invisible en appliquant éventuellement un effet de fade out s'il a été défini. Elle s'occupe également d'interrompre le ou les différents Worker(s) à l'œuvre si une animation était en cours.

Export (templateName { ; TemplateFolderPath}) : Cette formule de créer un template de votre objet AJUI Tip dans un fichier au format JSON. La formule requiert en premier paramètre le nom à associer au template (nom du fichier). Un deuxième paramètre optionnel vous permet de spécifier le chemin de dossier, sinon il sera exporté dans le dossier par défaut (.../Ressources/AJUI_Tip_Templates/).

4 Méthodes du composant

AJUI_Tip_info

Paramètre(s)	- Aucun
Valeur de retour	Numéro de version (texte)
Description	Cette méthode retourne le numéro de version du composant lorsqu'il a été compilé.

AJUI_TIP_loadTemplates ({ FolderPath })

Paramètre(s)	- FolderPath : Chemin du dossier (texte) (optionnel)
Valeur de retour	Collection de Tip (collection d'objet)
Description	Cette méthode permet de charger tous les fichiers modèle contenu dans un répertoire. Elle retourne le contenu du fichier sous forme de collection d'objet Tip. Il est possible d'y passer le chemin du dossier en paramètre, si ce n'est pas le cas, la méthode utilisera le chemin par défaut (voir la méthode AJUI_Tip_saveTemplate)

New AJUI_Tip ({ tip })

Paramètre(s)	- Template (objet) (optionnel)
Valeur de retour	Une instance de Tip
Description	<p>Cette méthode retourne une variable objet qui représente une instance d'AJUI Tip. Elle contient l'ensemble des propriétés et leurs valeurs par défaut ainsi que les formules (fonctions membres) pour les manipuler. Il est possible de lui passer en paramètre un objet afin d'importer un template d'AJUI Tip (fichier JSON). L'objet attend comme propriétés :</p> <ul style="list-style-type: none"> - templateName : Corresponds au nom du fichier JSON à importer (template). Si le fichier n'est pas trouvé, la méthode retournera une nouvelle instance d'AJUI Tip. - templatePath (optionnel) : Vous pouvez préciser un chemin pour récupérer le fichier sinon, le composant cherchera dans le dossier par défaut se trouvant dans les ressources de la base hôte (.../Ressources/AJUI_Tip_Templates/).

AJUI_Tip_clearCache ()

Paramètre(s)	- Aucun
Valeur de retour	Aucune
Description	Cette méthode vous permet de faire un nettoyage des instances stockées en interne dans une variable interprocess. Cela est important afin de libérer de la mémoire. Nous vous recommandons de le faire lors ce que vous fermez le process qui contient le ou les formulaires qui affiche des info-bulles. Cette méthode nettoie les instances du process courant seulement.

5 Prise en main

5.1 Installation

AJUI Tip doit être placé dans le dossier composant (Components) de votre application.

5.2 Principes d'utilisation

Dans cette partie, nous décrivons la suite d'opération à réaliser afin de générer un simple Tip dans le contexte du formulaire principale.

5.2.1 Prérequis

AJUI Tip possède un formulaire partagé (container) qui sert à créer visuellement l'ensemble des éléments du Tip. Cependant, il faut le rendre disponible au niveau du formulaire de l'application principale. Pour cela, AJUI Tip a besoin d'un objet sous-formulaire qui va servir de modèle. Cet objet est fourni avec la librairie "AJUI_Tip_Library.4IL" que vous pouvez ouvrir depuis votre application (/File/Open/Object Library...) et il vous suffira de faire glisser (Drag & Drop) l'objet "subform Tip" sur votre formulaire principal.



Pourquoi cela ? Parce que lorsqu'on voudra afficher un Tip avec la méthode « Show », le composant va se baser sur le nom passé en propriétés pour voir si un objet Tip correspondant à celui-ci existe déjà. Si ce n'est pas le cas, il va dupliquer le sous-formulaire modèle que vous aurez ajouté et y associer une instance du formulaire partagé.

Donc, la première chose lorsque vous avez décidé quel formulaire va accueillir le ou les Tip que vous allez utiliser, c'est d'y ajouter cet objet.

5.2.2 Noms et propriétés réservées

Le composant utilise des noms ou propriétés qui lui sont propres et qu'il considère comme réservé.

- **AJUI_Tip_container** : Nom du formulaire partagé par le composant.
- **AJUI_Tip_sf** : Nom que doit porter le sous-formulaire servant de modèle (voir prérequis).

5.2.3 Mise en place d'un premier Tip

La première étape consiste à appeler la méthode d'initialisation (new AJUI_Tip) afin de récupérer dans un objet, une instance contenant l'ensemble des propriétés d'un Tip par défaut ainsi que ses formules. Nous vous recommandons d'utiliser directement la variable « Form » pour stocker l'instance courante afin de facilement naviguer entre votre méthode formulaire et celles de vos objets formulaires.

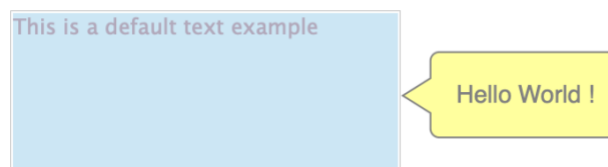
La deuxième étape consiste à utiliser les formules de votre objet afin de définir les propriétés de l'instance Tip selon vos besoins. Le minimum consiste à définir le nom d'une cible (target), c'est à dire l'objet de formulaire, auquel l'instance Tip sera attaché. Il est aussi possible de définir des coordonnées précises le cas échéant, si le tip ne s'applique pas à un objet 4D de formulaire. Ensuite il vous faut définir un nom qui restera associé à l'instance Tip. Et finalement assurez-vous de définir un texte à afficher. Ce texte pourra être du texte multistyle, il sera possible de mettre des mots ou lettres en gras, en italique ou en couleur à l'intérieur d'une zone de texte.

```
$evt:=Form event
Case of
  ▾ : ($evt=On_Load)
    Form.tip:=New AJUI_Tip
    Form.tip.TargetName("test_tip")
    Form.tip.TipName("Test1")
    Form.tip.TextLabel("Hello World !")
End case
```

Pour finir, il faut utiliser la formule « Show » afin de lancer la génération de l'instance Tip. Faites-le par exemple sur un de vos objets de formulaire avec un événement « On Clicked ».

```
$evt:=Form event
Case of
  ▾ : ($evt=On_Clicked)
    Form.tip.Show()
End case
```

Si les précédentes étapes ont été bien respectées, vous devriez voir s'afficher votre Tip.



5.2.4 Utiliser un formulaire à la place du texte

Les opérations faites dans ce chapitre sont les mêmes à la différence qu'il est possible d'associer un formulaire plutôt que du texte dans un Tip. Pour cela, utiliser la formule « SubformName » afin d'y associer le nom du formulaire de votre application que vous souhaitez afficher dans la tipBox.



Il n'est pas possible de définir l'affichage d'un texte et d'un formulaire dans un Tip et donc dans un tel cas le composant utilisera en priorité le formulaire. **Attention l'affichage d'un formulaire désactive les effets de fondu en entrée et de fondu en sortie (fade in/ fade out).**

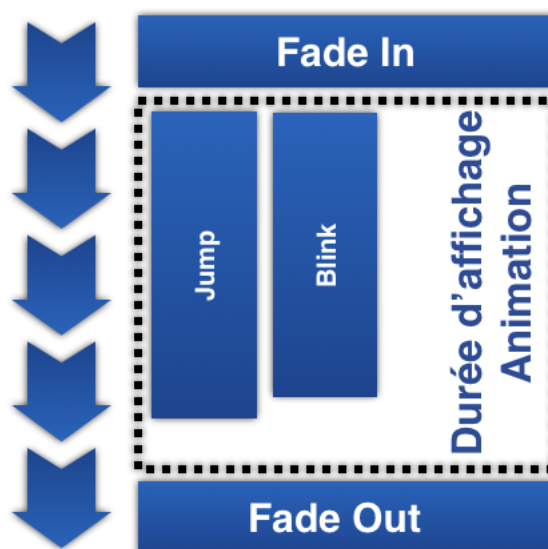
5.2.5 Ajouter des animations

AJUI Tip offre la possibilité aux utilisateurs d'animer une info-bulle. (voir le chapitre sur le cycle de vie).

Actuellement, le composant permet de réaliser les animations suivantes :

- Fondu en entrée
- Clignotement
- Saut
- Fondu en sortie

Les fondus ont leur propre cycle d'animation donc il est possible de tous les activer. Par contre les autres animations ont un cycle commun donc une seule pourra être ajoutée aux effets de fondus.



Passons à la mise en place des animations. Nous partons du principe que vous avez préparé votre info-bulle tel que décrit dans le chapitre 4.2.3 « mise en place d'un premier Tip ». Il ne reste plus qu'à définir les propriétés des animations que vous désirez appliquer.

Fondus en entrée et en sortie :

La mise en place de ces deux effets est à peu près similaire. Il faut activer les effets (*activateFadeIn*, *activateFadeOut*) via les formules. Il est également possible de spécifier la durée de chaque effet.

```
Form.tip.ActivateFadeIn(True)
Form.tip.ActivateFadeOut(True)

Form.tip.TimerFadeIn(3000)
Form.tip.TimerFadeOut(3000)

Form.tip.TipDuration(3000)
```

Clignotement :

La mise en place de l'effet de **clignotement** (Blink) nécessite la définition de trois propriétés dont une obligatoire (BlinkNumber) qui définit le nombre de cycles de l'effet (une valeur de zéro n'activera pas l'effet). Les deux autres propriétés concernent la durée d'un cycle ainsi que la proportion de temps alloué (en pourcentage) à sa visibilité.

```
Form.tip.BlinkNumber(5)
Form.tip.BlinkDuration(1000)
Form.tip.BlinkOn(50)

Form.tip.Show()
```

Saut :

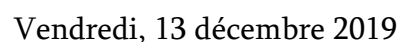

```
Form.tip.JumpNumber(3)
Form.tip.JumpDuration(1000)
Form.tip.JumpOn(50)
Form.tip.JumpOffset(20)

Form.tip.Show()
```

Il est recommandé d'utiliser un conteneur différent pour chaque tip différents. Un **tip** est différent lorsque son **tip** name change. Si vous définissez un tip "info" puis un tip "alert" constitué de propriétés spécifiques, il est recommandé d'utiliser un conteneur spécifique à chacune de ces définitions pour éviter des conflits d'instance lors d'animations (Ex: Form.info et Form.alert).

5.3.1 Listbox

Concernant la récupération des coordonnées, nous vous conseillons de récupérer les valeurs Mouse X et Mouse Y sur le clic sur la listbox pour ensuite déterminer la cellule pointée. Une fois le calcul réalisé, vous pourrez définir les coordonnées de la cellule pour générer l'info-bulle. Un exemple vous est fourni avec l'application de test [AJUI Tip Lab](#).



5.3.2 Multipages

Nous allons traiter ici le cas où votre formulaire comprend plusieurs pages et que vous désirez afficher des info-bulles sur différentes pages.

Concrètement rien de changé par rapport aux prérequis habituels. Il faudra simplement prendre la précaution de dupliquer l'objet sous-formulaire « *AJUI_Tip_sf* » sur chacune des pages et d'ajouter le suffixe "_page" + numéro de page. (Exemple : *AJUI_Tip_sf_page2* pour la page 2). Cette règle ne s'applique pas à la page 1 qui conserve le nom initial « *AJUI_Tip_sf* ».

Faites attention à toujours bien générer vos info-bulles dans le bon contexte de page. Par exemple, si vous générez sur la méthode de chargement du formulaire votre info-bulle et qu'ensuite vous enchaînez sur un changement de page, il y aura un problème car le composant l'aura créée dans la première page mais pas dans la page ouverte courante.

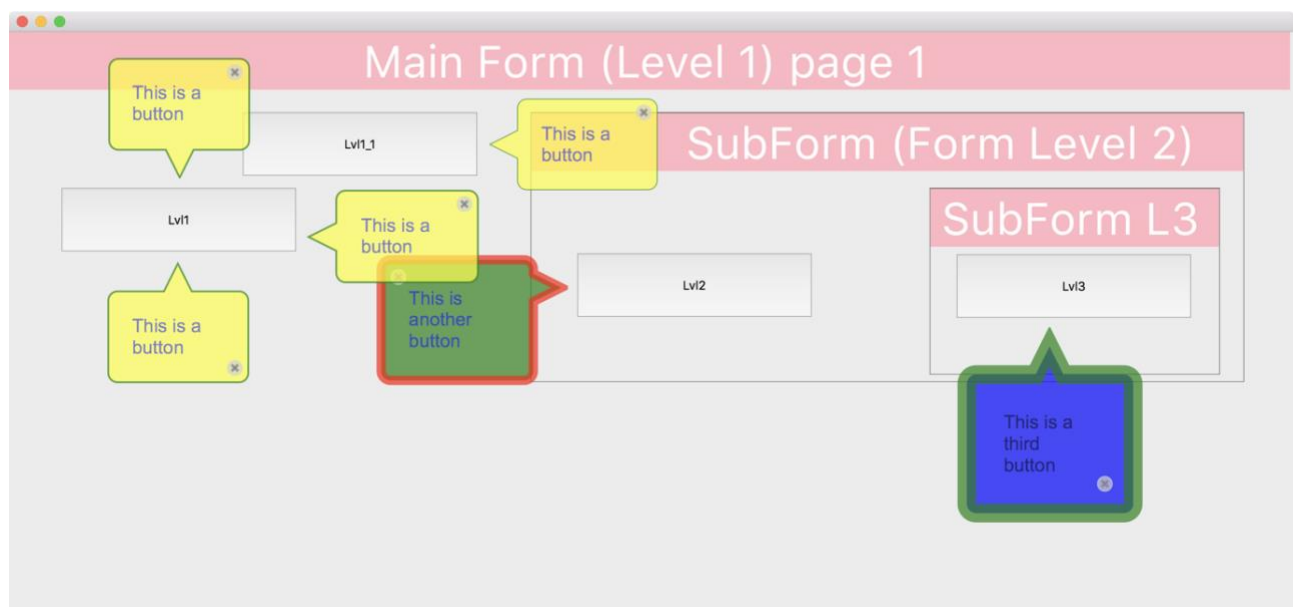
5.3.3 Sous-formulaire

Dans ce chapitre nous traitons de l'utilisation des info-bulles associées à un objet cible inclut dans un sous-formulaire. Ce cas de figure est assez simple à traiter.

*NB: Il est important de comprendre que l'objet sous-formulaire « *AJUI_Tip_sf* » doit absolument être au niveau du formulaire principal afin de pouvoir être totalement visible et au premier plan.*

Une fois défini, le composant se chargera de remonter les informations de l'info-bulle au premier niveau afin que celle-ci apparaisse dans son contexte.

Aperçu sur trois niveaux :



5.3.4 Utilisation de la commande Form de 4D dans un sous-formulaire

Il est possible d'associer l'objet Form de 4D dans le contexte du sous-formulaire du Tip. Pour cela, vous DEVEZ définir « Form.tip_sf_details:=new object () » dans le formulaire principal dans lequel le Tip sera affiché. Vous pouvez ensuite ajouter toutes les propriétés donc vous avez besoin à cet objet. Une fois cela fait, il est possible de communiquer via cet attribut entre le formulaire principal et le sous-formulaire chargé dans le Tip.

6 Conclusion

Le but de ce document était de vous présenter les principes théoriques du composant ainsi que les différentes méthodes, formules et propriétés à votre disposition pour pouvoir gérer la construction et l'animation des info-bulles.

Quant aux éléments pratiques présentés ils ont pour but de vous permettre de mettre un premier pied à l'étrier et d'aborder quelques cas particuliers qui pourrait intervenir dans l'utilisation du composant. Pour plus d'informations pratiques et une meilleure visualisation du rendu de la génération des info-bulles, nous vous invitons à essayer l'application de test (lab) qui est fourni avec le composant.

Vous désirez une aide pour l'implémentation du composant AJUI_Tip dans votre application. Vous désirez modifier ou étendre ses fonctionnalités pour un usage spécifique. Vous désirez disposer du code source du composant AJUI_Tip afin de pérenniser son usage dans votre application avec les futures versions de 4D. N'hésitez pas à nous contacter pour en discuter.