

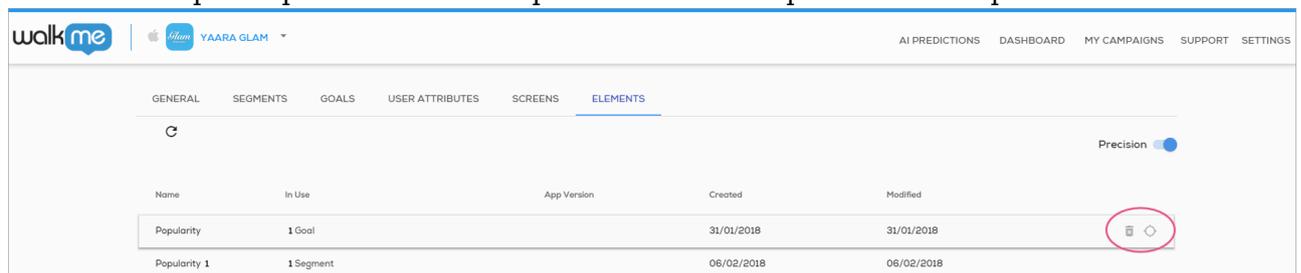
# Mobile : les paramètres de précision de l'élément

À partir du SDK WalkMe Mobile version 1.7.0, vous pouvez modifier les paramètres de précision des éléments de votre application pour améliorer la capacité WalkMe à identifier des éléments spécifiques dans votre ou vos applications. Les modifications disponibles diffèrent entre les applications OS, natives et hybrides en raison des différences de l'algorithme.

En tant que **meilleure pratique**, il est recommandé d'étendre votre sélection d'élément à la région la plus grande possible pour maximiser la région couverte pour l'utilisateur final. De cette façon, l'élément comprend le plus de détails possibles dans la capture.

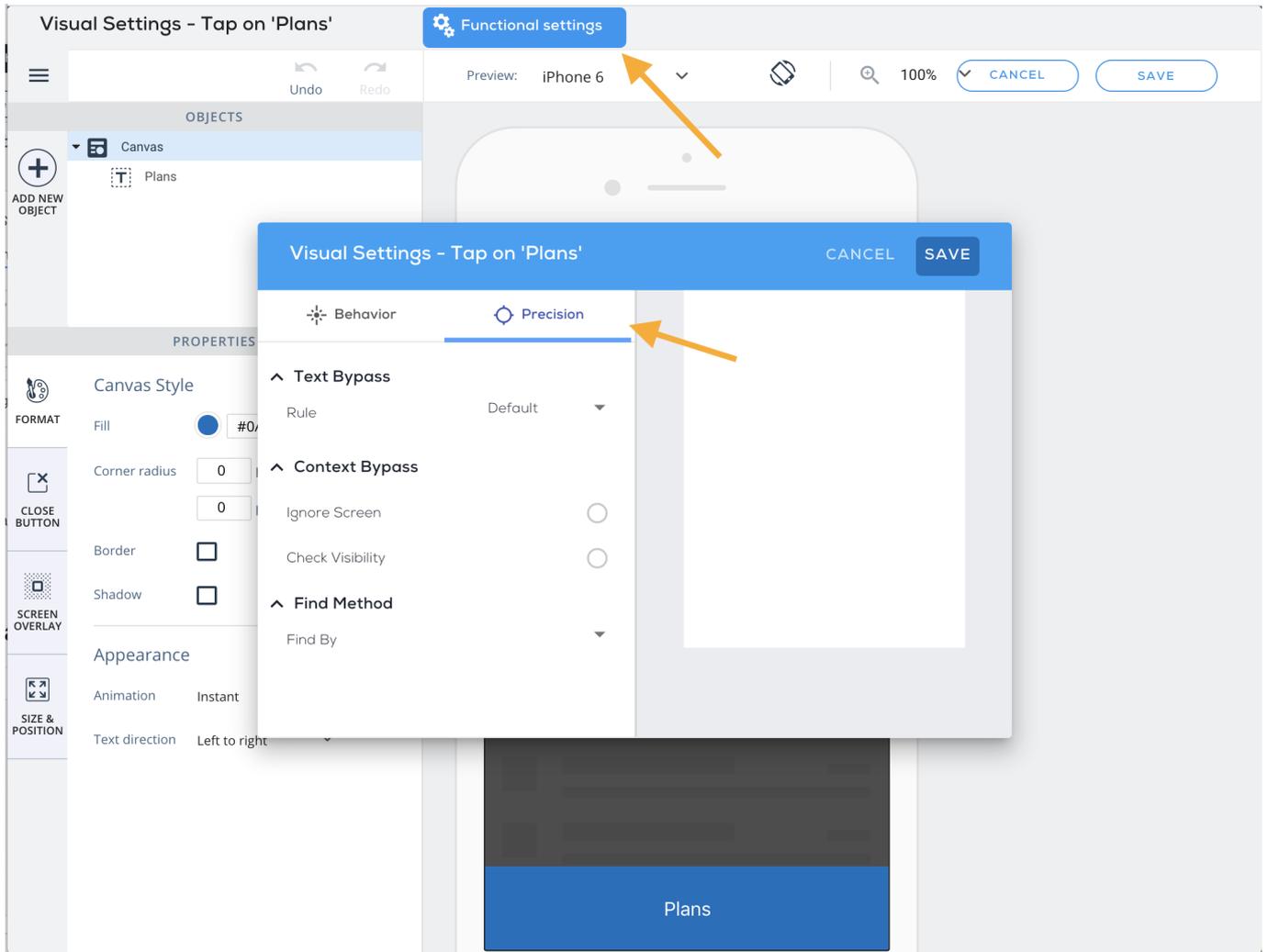
## Étapes pour ajouter les paramètres de précision

1. Dans la console mobile, naviguez jusqu'à **PARAMÈTRES > Paramètres de l'application > ÉLÉMENTS** ;
2. Survolez n'importe quel élément et cliquez sur l'icône de paramètres de précision :



3. Sélectionnez les paramètres de précision souhaités ;
4. Cliquez sur **Save** (enregistrer).

Toutes les informations ci-dessous s'appliquent aux étapes des Smart WalkThrus :



## Matrice de paramètres de précision

iOS natif	iOS hybride	Android natif	Android hybride
-----------	-------------	---------------	-----------------

Par défaut Ignorer le texte Faire correspondre le texte Ignorer l'écran Ignorer l'ID de l'accessibilité Vérifier la visibilité Trouver par Hiérarchie (xpath) Trouver par type	Par défaut Ignorer le texte Identifier par texte Ignorer l'écran Trouver par Hiérarchie (xpath) Trouver par type	Par défaut Ignorer le texte Faire correspondre le texte Ignorer l'écran Ignorer le nom de la classe Ignorer l'ID Ignorer l'ID de l'accessibilité Trouver par Hiérarchie (xpath) Trouver par type	Par défaut Ignorer le texte Identifier par texte Ignorer l'écran Ignorer le nom de la classe Ignorer l'ID Trouver par Hiérarchie (xpath) Trouver par type
---	---	--	--

## Modifications de précision - <synchronisation

Toutes les mises à jour de précision ne sont pas immédiates. Ceci est dû à la façon dont les modifications sont migrées vers la campagne elle-même côté serveur. Voici un visuel rapide sur les 4 façons de lire le contenu contre les 2 façons de mettre à jour la précision :

	aperçu	Simuler	Éléments capturés	Publié
Étape SWT	Automatique	~ 1-2 minutes	S/O	~ 1-2 minutes
Élément capturé	~ 1-2 minutes	~ 1-2 minutes	Automatique	~ 1-2 minutes

## Définitions de réglage de précision

### Par défaut

Les facteurs de l'algorithme WalkMe prend en compte de nombreux aspects d'un élément pour évaluer son existence. La description de l'élément complet peut être trouvée en cliquant sur un élément capturé dans la console.

### Ignorer le texte

Si la capture contenait du texte, l'algorithme WalkMe ignorera ce texte lorsque vous essayez de trouver l'élément capturé. S'il y a un élément (par exemple, un bouton) qui change le texte ou s'il essaie de capturer un tableau/une liste avec le texte dynamique. S'il n'y a pas de valeur de texte dans le code capturé, Ignorer le texte n'aura aucun effet : « texte » : « [ici] »

```
{  
  "uiKitType": "UITabBarButton",  
  "frame": "{(227, 730), (71, 48)}",  
  "isInteractable": true,  
  "screen": "Glip.TabBarController\\Glip.RCUIPopGestureNavigationController\\Glip.GlipUIRecentChatViewController\\Glip.GlipUIRecentChatAllTableViewCellController",  
  "isInput": 0,  
  "windowBounds": "{(0, 0), (375, 812)}",  
  "type": "UITabBarButton",  
  "propertyName": "",  
  "xpath": "UIWindow()\\UILayoutContainerView(0)\\UITabBar(1)\\UITabBarButton(4)",  
  "accessibilityInfo": {  
    "accessibilityLabel": "",  
    "accessibilityHint": "",  
    "accessibilityValue": "",  
    "text": "Contacts",  
    "containerVC": "Glip.TabBarController",  
    "imageFingerprint": 7551607122347454464,  
    "referenceVC": "Glip.RCUIPopGestureNavigationController",  
    "sdkVersion": "2.3.0",  
    "uiControlArray": [{"target": "UITabBar", "actions": [{"_buttonDown": "_buttonUp", "_buttonCancel": "_sendAction:withEvent:"}]}]}  
}
```

## Faire correspondre le texte

S'il y a eu un texte capturé (comme vu dans l'image ci-dessus), Faire correspondre le texte peut réduire l'algorithme pour se concentrer sur le texte. Regex peut être utilisé, mais n'est pas nécessaire. Remarque : cela ne s'applique qu'aux éléments natifs (et non hybrides).

## Identifier par texte

Permet de se concentrer sur le texte capturé dans un élément hybride. Remarque : si le texte capturé est séparé en plusieurs chaînes, WalkMe ne s'identifiera que par la première chaîne.

## Ignorer l'écran

Certaines valeurs d'écran sont soit dynamiques, soit pas assez descriptives. Ce paramètre nous permet d'ignorer la valeur de l'écran dans ces cas. Remarque : pour les WalkThrus, seule la première étape prend l'écran en considération, donc si la valeur de l'écran de la première étape est dynamique, vous devez ajouter l'écran Ignorer. Toutes les étapes ultérieures ne tiendront pas compte des valeurs d'écran.

## Ignorer le nom de la classe

Android permet aux éléments d'avoir des noms de classe. Ceci est écrit en tant que « class\_name » dans la capture. Lorsque l'élément que vous essayez de sélectionner a une classe dynamique, il est important d'ignorer le nom de la classe. Remarque : cela ne s'applique qu'à Android

```
{  
  "screen": "HomeActivity#HomeMeetingsPageFragment",  
  "activity": "com.glip.ui.home.HomeActivity",  
  "isInput": false,  
  "indexOfListInPath": 0,  
  "viewPositionInList": -1,  
  "text": "Message",  
  "isOnDialog": false,  
  "sdk_ver": "2.5.3",  
  "version": "2",  
  "id": -1,  
  "accessibilityId": "Message",  
  "class_name": "com.google.android.material.bottomnavigation.BottomNavigationView",  
  "index_on_parent": 0,  
  "xpath": {"nodes": [{"id": -1, "className": "android.widget.LinearLayout", "index": 0}, {"id": -1, "className": "android.widget.FrameLayout", "index": 0}, {"id": 2131296368, "className": "androidx.appcompat.widget.FitWindowsFrameLayout", "index": 0}, {"id": 16908290, "className": "androidx.appcompat.widget.ContentFrameLayout", "index": 0}]}  
}
```

## Ignorer l'ID

Comme `class_name`, les ID sont uniques à Android. Toutes les captures d'élément n'auront pas d'ID, mais si l'ID n'est pas unique, il pourrait être utile de l'ignorer pendant le dépannage.

```
{ "screen": "HomeActivity#HomeGroupsPageFragment", "activity": "com.glip.ui.home.HomeActivity", "isInput": false, "indexOfListInPath": 0, "viewPositionInList": -1, "text": "Message", "isOnDialog": false, "sdk_ver": "2.5.3", "version": 2, "id": -1, "accessibilityId": "Message", "class_name": "com.google.android.material.bottomnavigation.BottomNavigationView", "index_on_parent": 0, "xpath": { "nodes": [ { "id": -1, "className": "android.widget.LinearLayout", "index": 0 } ] }
```

## Ignorer l'ID de l'accessibilité

WalkMe est en mesure de se concentrer et d'identifier des éléments en fonction de divers facteurs décrits dans cet article. Certains développeurs d'application codent des éléments utilisant les ID d'accessibilité et des identifiants uniques indépendamment de toutes les autres caractéristiques dynamiques, telles que les images, le texte, la langue, etc. Si un ID d'accessibilité est disponible pour un élément, WalkMe se concentrera dessus. Ce paramètre peut être utilisé pour ignorer l'ID d'accessibilité si besoin.

## Trouver par Hiérarchie (xpath)

Par défaut, WalkMe se concentre fortement sur Hiérarchie dans l'algorithme. Hiérarchie regarde le « chemin » de l'élément qui est sélectionné. Par exemple, Page > Liste > Rangée 1 > Cellule 1. Dans la console, c'est le [x-path]. Il existe un problème connu où il peut y avoir différentes hiérarchies entre iOS 13 et supérieur, et 12 et inférieur. Si possible, testez un appareil avec iOS 13 et un appareil avec iOS 12. Cela sera de moins en moins fréquent lorsque les gens migrent vers iOS 13+.

## Trouver par type

Par rapport au Web, c'est la « Classe d'élément ». Par exemple, chaque élément de liste ou chaque bouton Enregistrer qui se trouve à l'écran aura probablement la même classe. Ceci est [type] dans la console. Le cas d'utilisation le plus courant pour l'utilisation de Type est d'éliminer le problème de hiérarchie iOS12/iOS13 précisé plus haut.

## Vérifier la visibilité (ne s'applique qu'aux étapes iOS SWT)

Parfois, les écrans de l'application sont développés en couches et certains éléments, bien que cachés sous la couche supérieure, existent toujours techniquement. WalkMe est toujours en mesure de trouver ces éléments qui mènent à des étapes pointant vers des éléments cachés. Pour résoudre ce problème, activez ce paramètre de sorte que WalkMe ne trouve ces éléments que lorsqu'ils sont réellement visibles.

