Hub One

Guide Triggers WireLess TelNet

Triggers TelNet

Intrudoction

Un trigger est une procédure qui est lancée quand certaines conditions sont vérifiées pour exécuter certaines actions. Il est associé à un ensemble de conditions qui doivent être vérifiées pour qu'il soit exécuté. Ces contions peuvent être, par exemple, la présence d'un texte donné à une position donnée de l'écran, ou la position du curseur.

Les triggers étaient originalement utilisés avec un module "voice" est formés de trois étapes :

- Vérification des conditions de déclenchement (section "match");
- Si elles sont vérifiées (eventuellement) prononcer quelque chose (section "TTS");
- Puis (éventuellement) attendre une saisie utilisateur par voix ou scanner (section "ASR").

Chacune de ces 3 parties peut exécuter des actions supplémentaires qui ne sont pas directement liées à la voice, comme, par exemple, exécuter un script, formatter une saisie utilisateur, etc.

Quand les triggers ne sont pas utilisés avec la voice, il n'est plus possible de prononcer ou reconnaître quelque chose mais toutes les autres fonctionnalités qui ne sont pas directement liées à la voice restent disponibles.

Pour assurer la compatibilité ascendante, les fonctionalités sont configurées au même endroit où elles le seraient si l'on utilisait la voice.

Match section [VOICETRIGGER_XX]

On=<Yes/No> (default : No)

Permet d'activer/désactiver le trigger.

Type_X= (default : A=)

Type de comparaison: A Alphabétique, N Numérique.Egal, différent, superieur, inférieur, supérieur ou égal, inférieur ou égal.

Row_X=<entier> (default : 0)

Ligne sur laquelle le texte sera cherché. 0 (zéro) signifie toutes les lignes. Si la ligne et la colonne sont tous les deux mis à zéro, la condition n'est pas testée.

Col_X=<entier> (default : 0)

Colonne sur laquelle le texte sera cherché. 0 (zéro) signifie toutes les colonnes. Si la ligne et la colonne sont tous les deux mis à zéro, la condition n'est pas testée.

Match_X=<texte> (default : "")

Texte à chercher sur la position Row_X / Col_X. Pour que les espaces finales soient considérés, le texte doit être délimité par des guillemets "". Si vous ne saisissez aucun texte, la condition n'est pas évaluée (VRAI).

CursorAtRow=<entier> (default : 1)

Ligne de la position d'une zone de saisie. Domaine : [1;24[. Si X et Y sont tous les deux mis à zéro, le trigger pourra être déclenché quelque soit la position du curseur.

CursorAtCol=<entier> (default : 1)

Colonne de la position d'une zone de saisie Domaine : [1;24[. Si X et Y sont tous les deux mis à zéro, le trigger pourra être déclenché quelque soit la position du curseur.

Previous= (default : "")

Formule pour tester le trigger précédant et valider le courant.Egal(=) ou différent (!). Combiner avec AND (&) et OR (|).

HideScreen=<Yes/No> (default : No)

Yes=Ne pas montrer le contenu de l'écran quand ce trigger est déclanché.

HideColor=<entier> (default : 2)

Couleur du masque à appliquer sur l'écran pendant la durée d'application du trigger.

HideArea= (default : "")

Texte à chercher sur l'écran. Insérez une ou plus étiquettes {r,c,l} ici. Par exemple : "{15,5,3}{10,8,10}".

Scanner=<num> (default : 0)

signifie que ce trigger ne change pas le pilotage standard du scanner. . 0 : laisser tel quel, 1 : verrouiller, 2 déverouiller.

LockKbd=<num> (default : 0)

Pilotage du verrouillage du clavier par le trigger. 0 : laisser tel quel, 1 : verrouiller, 2 déverouiller.

Previous= (default : "")

Formule pour tester le trigger précédant et valider le courant. Egal(=) ou différent (!). Combiner avec AND (&) et OR (|).'=5' Valide si précédant est 5. '!5' Valide si précédant n'est pas 5.'!2&!3&!4' Le trigger précédant ne doit pas être ni 2 ni 3 ni 4.'=5|=7' le trigger précédant doit être 5 ou 7.

DoIf=<expression> (default : 1)

Expression qui doit être évaluée à true pour que le trigger puisse être exécuté. Les parenthèses sont obligatoires. Variables : #LFI (dernière saisie formatée), #LRI (dernière saisie brute), #N (où N est un numéro de variable VoiXtreme). Types de données : entier, réel, texte. Operateurs : +, -, *, / et % (modulo), ^ (exposant), & (et), | (ou), << (concaténer), <, <=, >, >=, !=, =. L'opérateur << retournera toujours une valeur de type texte, les autres retourneront une valeur numérique. Toute valeur non nulle ou texte non vide sera évalué à true. Voir la documentation du Telnet pour plus de détails. Pour avoir le comportement de l'opérateur "not", faire "0 & expression". Et pour le "-" unaire, faire "0 - expression".L'opérateur << attend deux paramètres de type string, les paramètres de type autre que string sont convertis en string : "10 << 20" retourne '1020'. Tous les autres opérateurs attendent des valeurs numériques, les valeurs non numériques sont converties : "10 + '20.5'" retourne 30.5, "'3.14' + 'abc'" échoue.Les opérandes droits des opérateurs / et % doivent être des valeurs évaluées non nulles. % ne marche qu'avec des nombres entiers.

AutoTrigger= <Yes/No> (default : No)

Yes = Enverra automatiquement le terminator après les TTS, s'il y en a. Les auto-triggers étaient originellement définis en utilisant la grammaire "$" dans la section ASR. Cette ancienne syntaxe est toujours supporté pour la compatibilité ascendante mais est obsolète.

AutoTriggerDelay=<entier> (default : 0)

Temps d'attente avant que le terminator ne soit envoyé. En milisecondes. Le délai pour les auto-triggers était originellement défini après la grammaire "$" dans la section ASR. Cette ancienne syntaxe est toujours supporté pour la compatibilité ascendante mais est obsolète.

OnInit=<fichier> (default : "")

Chemin du fichier script à exécuter au début de lexécution du trigger, après que les conditions aient été testées. Raccourcis : $WTN = dossier d'installation du Telnet (comme par exemple /Application/WTn52/), $FLASH = dossier de mémoire persistante (comme par exemple : /Application/).

Text To Speech section [VOICETRIGGER_TTS_XX]

On=<Yes/No> (default : No)
Active/ désactive les actions TTS dans ce trigger.

OnTts=<fichier> (default : "")

Chemin du fichier script à exécuter avant le traitement de la partieTTS.

Raccourcis :

- $WTN = dossier d'installation du Telnet (comme par exemple /Application/WTn52/)
- $FLASH = dossier de mémoire persistante (comme par exemple : /Application/).

Text To Speech section [VOICETRIGGER_ASR_XX]

On=<Yes/No> (default : No)

Active / désactive les actions ASR pour le trigger.

LenMin=<digit> (default : 2)

Longueur minimum de saisie vocale attendue. Les saisies inférieures à cette valeur ne seront pas acceptées et le système demandera une nouvelle saisie.

LenMax=<digit> (default : 4)

Longueur maximum de saisie vocale attendue. Les saisies supérieures à cette valeur ne seront pas acceptées et le système demandera une nouvelle saisie.

CancelDo=<Yes/No> (default : Yes)

Permet à l'utilisateur d'annuler la saisie de données et envoi la séquence d'annulation à l'hôte.

Cancel=<scan-code> (default : 0123)

Scan code envoyé à l'hôte quand l'utilisateur annule la saisie. Valeur par défaut, 0123 = CURSOR-UP (voir TN52_CURSOR_KEYS).

FormatKw= (default : "")

Chaine de format de saisie de Kbd Keyword FFF1. Peut contenir des données fixes et variables p/ex 999{2,15,3}. S'il est vide aucun formatage n'est fait. Le texte variable est signalé par trois valeurs numériques qui indiquent sa position par la ligne, la colonne, et sa longueur {r,c,l}.

OnAsr=<fichier> (default : "")

Chemin du fichier script à exécuter après avoir reçu une saisie utilisateur (par scanner donc lorsqu'il n'y a pas de voice).

Raccourcis :

- $WTN = dossier d'installation du Telnet (comme par exemple /Application/WTn52/).
- $FLASH = dossier de mémoire persistante (comme par exemple : /Application/).

Scripting et variables

Il existe un ensemble de variables contenant du texte et permettant de conserver des informations d'un trigger à l'autre. Ces variables peuvent être utilisées dans certaines propriétés décrites ci-dessus avec la syntaxe : « {#N} » où « N » est un numéro identifiant la variable.

Ces variables peuvent être récupérées et initialisées dans les scripts à l'aide des fonctions TrgGetVar() et TrgSetVar() (voir la documentation des fonctions BASIC).

Pour pouvoir utiliser un script, il faut le définir des un trigger

- Dans la section « match » si l'on souhaite l'exécuter dès le lancement d'un trigger (voir propriété « OnInit »).
- Dans la section TTS (voir propriété « OnTts » qui ne présente aucune différence avec « OnInit » lorsque l'on utilise pas de voice)
- Dans la section ASR si l'on souhaite l'exécuté après une saisie utilisteur à laquelle on voudrait par exemple accéder (voir propriété « OnAsr »).