

Guide de démarrage rapide des scripts personnalisés Unity Express

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Aperçu](#)

[Utiliser Cisco Unity Express Script Editor](#)

[Créer un script de base](#)

[Méthodes d'intégration](#)

[Exemple de flux d'appels](#)

[Exemple de script](#)

[Télécharger un script vers Cisco Unity Express](#)

[Dépannage](#)

[Annexe](#)

[Annexe 1 - Instructions système](#)

[Annexe 2 - Réinitialiser les traces par défaut](#)

[Informations connexes](#)

[Introduction](#)

Ce document fournit des informations au sujet du système Cisco Unity Express (CUE). Il offre plus particulièrement une introduction pour la création d'une application de réception automatique personnalisée dans le produit.

Pour plus d'informations sur Cisco Unity Express, reportez-vous au [Guide de Cisco Unity Express Script Editor](#). Ce document est destiné aux administrateurs qui ont besoin d'un guide de base pour commencer à utiliser Cisco Unity Express Editor et à gérer les scripts dans le produit Cisco Unity Express.

Remarque : la prise en charge des scripts personnalisés n'est pas disponible via le support technique Cisco standard. Pour toute question, commentaire et aide sur un script personnalisé, envoyez un e-mail à ask-cue-editor@external.cisco.com.

[Conditions préalables](#)

[Conditions requises](#)

Le lecteur doit connaître l'administration et la configuration de Cisco Unity Express via l'interface de ligne de commande (CLI) ou l'interface utilisateur graphique.

Cisco Unity Express Script Editor (CUEEeditor2.1.1.exe) et l'exemple de script utilisé dans ce document (CUE-AA-S6-AASample.zip) sont disponibles à partir du [centre de téléchargement du logiciel Cisco Unity Express 2.1.1](#).

Components Used

Les informations de ce document sont basées sur Cisco Unity Express version 2.1.1. L'exemple contient des étapes disponibles uniquement dans Cisco Unity Express 2.1.1. Bien que plusieurs principes soient les mêmes pour les versions antérieures. Les étapes spécifiques à 2.1.1 sont indiquées explicitement dans ce document.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

Aperçu

Le produit Cisco Unity Express de base est livré avec un élément de messagerie vocale et un standard automatique de base (dans Cisco Unity Express 2.1.1 et versions ultérieures, il existe deux scripts de standard automatique). De nombreux clients trouvent le standard de réception automatique trop limité. Cisco Unity Express dispose d'un éditeur de script qui permet la création de scripts personnalisés. Ces scripts personnalisés peuvent être chargés dans Cisco Unity Express et remplacer (ou travailler en plus) le standard automatique standard.

Avertissement : L'exemple de script présenté ici est utilisé à des fins d'illustration. Il peut être modifié de quelque manière que ce soit. L'objectif de ce document est d'illustrer le processus de conception et de mise en oeuvre d'un script personnalisé. Il n'est pas destiné à fournir un script de production terminé. Cependant, si vous choisissez d'utiliser ce script, enregistrez au minimum chacune des invites. Vous pouvez écouter les fichiers audio joints afin d'entendre leur son. Mais utilisez l'[Administration via le système de gestion des messages d'accueil/téléphone](#).

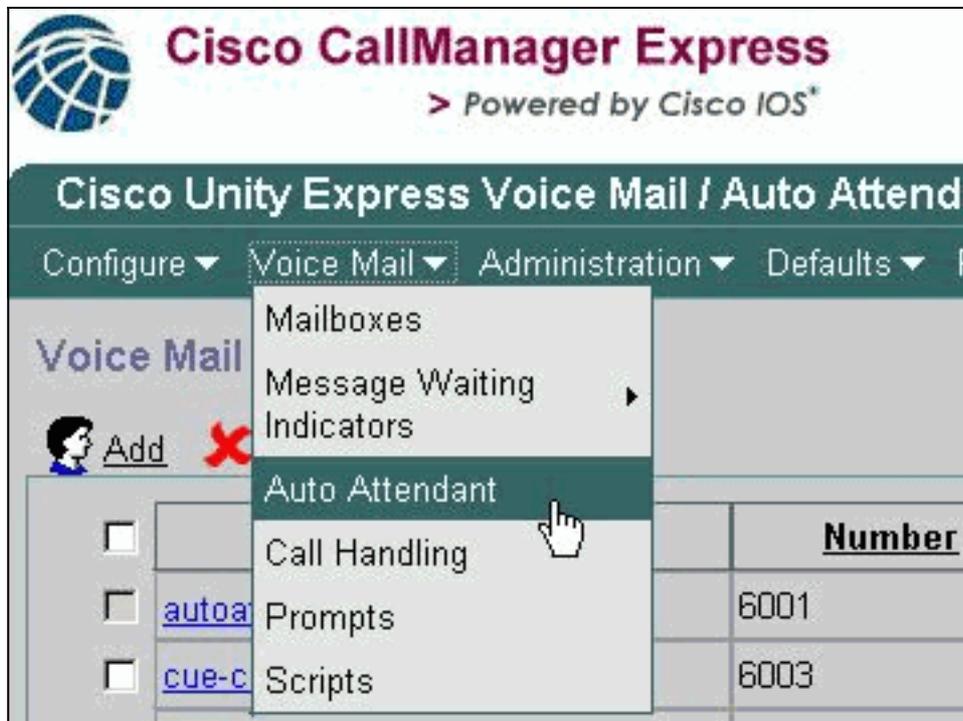
Remarque : Pour le moment, il n'est pas possible d'obtenir une copie des scripts système utilisés par Cisco Unity Express. Ces scripts comportent des étapes que l'éditeur de script normal ne prend pas en charge. Les scripts système de Cisco Unity Express 2.1.1 sont aa.aef, voicebrowser.aef, setmwi.aef, promptgmt.aef, checkaltgreet.aef, xfermailbox.aef et aasimple.aef.

L'éditeur de script est un outil qui vous permet de créer des scripts sur un PC distinct de Cisco Unity Express. Une fois créés, les scripts sont téléchargés et configurés sur Cisco Unity Express.

Cette liste explique une terminologie de base afin de vous aider à comprendre les concepts de ce document.

- **Étape :** bloc de base pour la création de script. Chaque étape est l'unité exécutable la plus basique, telle qu'une instruction « if », une instruction « Goto », etc.

- **Script** : une ou plusieurs étapes exécutées dans l'ordre. Un script est un fichier avec une extension .aef.
- **Variable** : il s'agit de variables dans un script. Les variables peuvent être de types différents, tels qu'Integer, Boolean, String, etc.
- **Paramètre** : propriété d'une variable de sorte que la variable soit exposée à l'administrateur via l'interface Web de Cisco Unity Express. Par exemple, si vous avez une variable OperatorExtension que vous voulez attribuer par défaut à la valeur 1000, elle doit parfois être changée en 2000. Dans ce cas, la variable est exposée via l'interface Web de sorte que l'intégralité du script n'ait pas besoin d'être chargée à nouveau dans Cisco Unity Express chaque fois que la valeur doit être modifiée.
- **Invite** : fichier .wav pouvant être lu. Il est soit téléchargé manuellement dans le système Cisco Unity Express, soit enregistré via le système d'administration via le téléphone (AVT) (avant la version 2.1.1, AVT était appelé système de gestion des messages d'accueil (GMS)). Toutes les invites utilisateur téléchargées sont placées dans le même répertoire. Ils sont visibles dans l'interface graphique utilisateur via l'option de menu **Messagerie vocale > Invite** ou dans l'interface de ligne de commande via la commande **show ccn Invite**. Dans un script, les invites utilisateur sont spécifiées en tant que P[<promptname>]. Les invites système sont préenregistrées et peuvent être utilisées. Ils sont spécifiés en tant que SP[<promptname>]. [L'annexe 1](#) répertorie les invites système disponibles.
- **Application** : script contenant toutes les invites et tous les paramètres renseignés. Par défaut, Cisco Unity Express est livré avec la messagerie vocale, le système AVT et une application de réception automatique simple, toutes configurées lorsque vous exécutez l'Assistant d'initialisation à la fin d'une installation.
- **Trigger** : le déclencheur indique à Cisco Unity Express qu'une application particulière doit être exécutée. Par exemple, lorsque vous composez le 1000, le système téléphonique (Cisco CallManager ou Cisco CallManager Express) achemine l'appel vers Cisco Unity Express. Lorsque Cisco Unity Express détecte qu'un appel est placé sur le numéro 1000, il recherche un déclencheur pour ce poste. L'application correspondante est alors lancée. En d'autres termes, il sait si un appel vers 1000 doit passer par la messagerie vocale, la réception automatique ou autre. Vous pouvez avoir plusieurs déclencheurs pour la même application. Un certain nombre de déclencheurs sont ajoutés par défaut, tels que les déclencheurs de la messagerie vocale, AVT et du standard automatique en conserve. Lorsque vous comprenez la terminologie, l'outil d'administration Web de Cisco Unity Express est plus facile à



comprendre.

Sur la page Web de Cisco Unity Express, l'option Réception automatique sous la catégorie Messagerie vocale fait référence aux applications définies dans la terminologie. Chaque entrée contient le numéro associé à l'application, le script auquel elle fait référence, tous les paramètres associés au script, que l'application soit activée ou non et le nombre de ports associés.

La gestion des appels ne contient actuellement que les numéros associés pour la messagerie vocale, la réception automatique intégrée et le système AVT. Dans Cisco Unity Express 2.1 et versions ultérieures, sous **Administration > Call-in Numbers**, tous les déclencheurs sont affichés, à la fois en JTAPI (pour Cisco CallManager) et en SIP (pour Cisco CallManager Express ou SRST). Actuellement, la seule façon de voir s'il y a plusieurs déclencheurs configurés qui pointent vers la même application est d'utiliser la commande **show ccn trigger** émise dans l'interface de ligne de commande.

L'option Invite fait référence aux fichiers audio (.wav) qui existent sur le système.

La sélection Scripts fait référence aux fichiers .aef que vous pouvez stocker sur le système. Actuellement, vous pouvez télécharger jusqu'à quatre scripts en plus des scripts système.

À partir de l'interface de ligne de commande, ces informations sont facilement disponibles à l'aide des commandes **show ccn**. Cette sortie représente des parties de chaque sortie de commande :

```
cue-3745-44a> show ccn application
Name: customaa
Description: customaa
Script: customaa.aef
ID number: 4
Enabled: yes
Maximum number of sessions: 8
OperatorExtension: 205
MainMenu: MainMenu.wav
ClosedGreeting: ClosedGreeting.wav
InvalidExt: InvalidExt.wav
MaxRetries: 3
MaxExtension: 205
SorryGoodbye: SorryGoodbye.wav
```

```
EnterExtension:          EnterExtension.wav
namePrompt:             namePrompt.wav
MinExtension:           200
...
```

```
cue-3745-44a> show ccn prompts
```

```
Name:                   MainMenu.wav
Language:               de_DE
Last Modified Date:    Wed Dec 22 03:34:57 GMT+00:00 2004
Length in Bytes:       121978
Name:                   EnterExtension.wav
Language:               de_DE
Last Modified Date:    Wed Dec 22 03:34:26 GMT+00:00 2004
Length in Bytes:       21338
Name:                   SorryGoodbye.wav
Language:               de_DE
Last Modified Date:    Wed Dec 22 03:34:02 GMT+00:00 2004
Length in Bytes:       17658
Name:                   ClosedGreeting.wav
Language:               de_DE
Last Modified Date:    Wed Dec 22 03:33:00 GMT+00:00 2004
Length in Bytes:       86138
Name:                   InvalidExt.wav
Language:               de_DE
Last Modified Date:    Wed Dec 22 03:47:07 GMT+00:00 2004
Length in Bytes:       29818
Name:                   namePrompt.wav
Language:               de_DE
Last Modified Date:    Wed Dec 22 03:40:28 GMT+00:00 2004
Length in Bytes:       22618
...
```

```
cue-3745-44a> show ccn trigger
```

```
Name:                   18955
Type:                   SIP
Application:            customaa
Locale:                 systemDefault
Idle Timeout:           10000
Enabled:                yes
Maximum number of sessions: 8
...
```

La sortie **show ccn application** affiche le nom de l'application, le script auquel il fait référence, s'il est activé ou non et le nombre d'appels simultanés qu'il peut traiter, ainsi que tous les paramètres de script avec leurs valeurs.

La commande **show ccn invite** répertorie toutes les commandes, leur langue, leur taille et leur date de modification.

Les informations **show ccn trigger** indiquent le numéro (« Nom ») ainsi que l'application qu'il appelle, l'état (activé/désactivé) et le nombre de sessions.

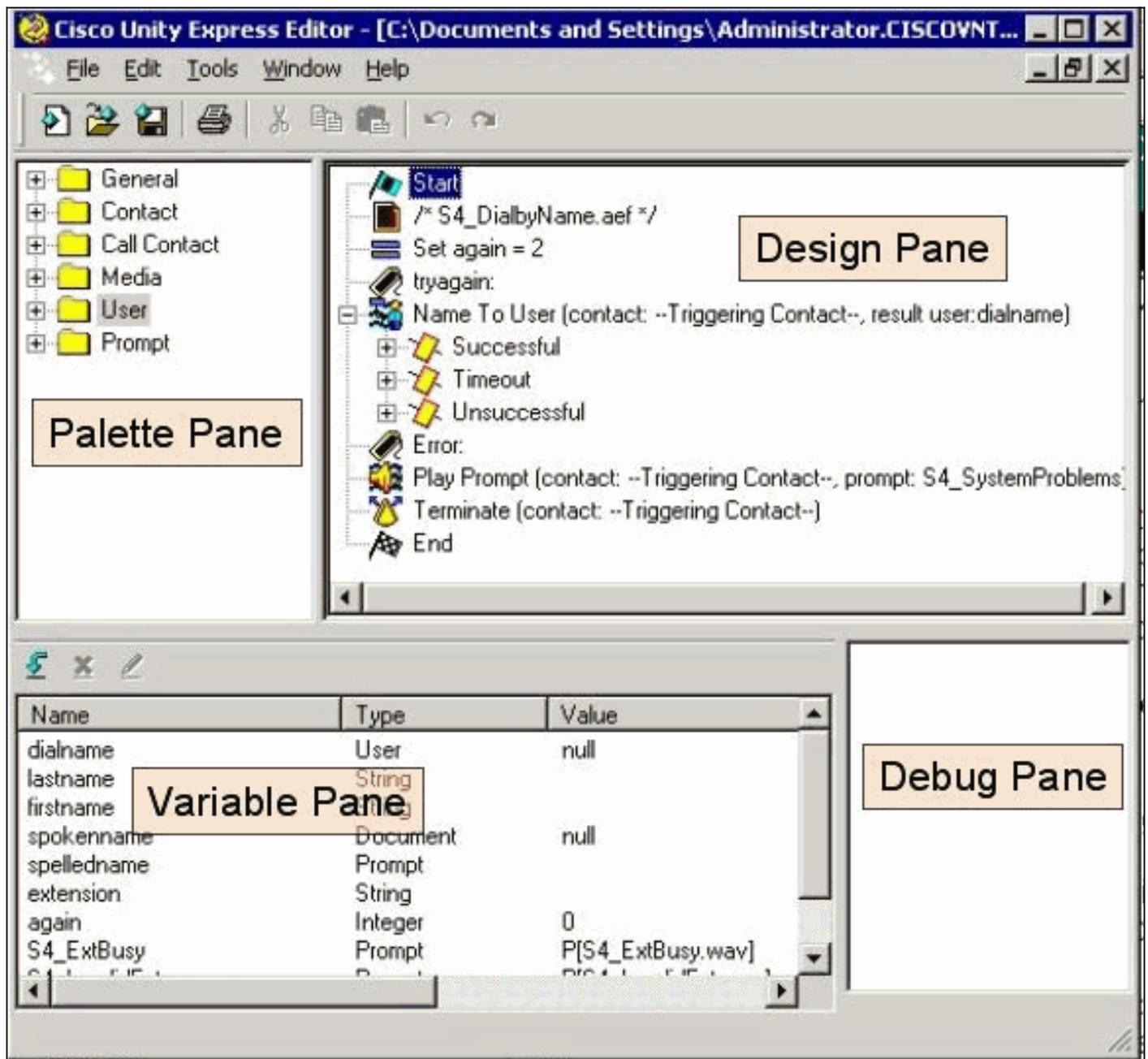
[Utiliser Cisco Unity Express Script Editor](#)

Utilisez [Script Editor](#) (clients [enregistrés](#) uniquement) pour créer un nouveau script. Il doit être installé sur un autre ordinateur.

Plusieurs exemples de scripts se trouvent au même emplacement. Ces scripts permettent d'apprendre à effectuer des opérations spécifiques dans un script. Cisco vous recommande d'installer l'éditeur et de télécharger quelques exemples de scripts afin de vous familiariser avec le

fonctionnement de l'éditeur.

Une fois Script Editor lancé, vous pouvez ouvrir un script comme illustré dans ce graphique :



Le volet Palette contient des dossiers avec chaque étape, classés en fonction de leur fonction principale. À partir de là, faites glisser les étapes vers le volet Création afin de créer un script. Dans le volet Variable, vous pouvez créer et modifier des variables. Après avoir fait glisser une étape vers le volet Création, vous pouvez cliquer avec le bouton droit sur une étape, sélectionner **Propriétés** et personnaliser l'étape et les variables affectées à l'étape. Avant de télécharger un script vers Cisco Unity Express, vous devez toujours le valider d'abord via l'option **Outils > Valider**. Toute sortie d'erreur s'affiche ensuite dans le volet Débogage.

L'une des choses les plus utiles à retenir lorsque vous utilisez l'éditeur est la fonction Aide. Vous pouvez y accéder directement via le menu Aide de la barre d'outils ou lorsque vous sélectionnez une propriété d'étape particulière.

[Créer un script de base](#)

Vous pouvez utiliser le menu Aide pour vous familiariser avec certaines des étapes. Dans Script Editor, faites glisser quelques étapes de la palette vers le volet de conception, puis cliquez avec le bouton droit de la souris sur l'une d'elles et sélectionnez **Propriétés**. Cliquez ensuite sur le bouton **Aide** et obtenez de l'aide sur cette étape particulière.

Ce document décrit la création d'un exemple de script. Le document traite également de certains des problèmes courants que les personnes tentent de résoudre lorsqu'elles créent un standard automatique personnalisé. L'objectif ici est de créer un script qui répond à un appel, vérifie s'il existe un message d'accueil secondaire d'urgence et si l'heure actuelle est des heures de bureau régulières ou non. S'il est fermé, lisez un message d'accueil personnalisé et envoyez l'appel à l'opérateur. S'il est ouvert, lisez un menu qui permet à l'appelant de composer le numéro 1 pour composer le numéro par nom, le numéro 2 pour composer le numéro par poste ou le numéro 0 pour un opérateur. L'option de numérotation par poste ne doit transférer l'appel que vers une plage spécifiée, et non vers un numéro que vous pouvez saisir.

Avertissement : L'exemple utilisé par ce document est à titre d'illustration uniquement. Bien que vous soyez libre d'utiliser les portions que vous voulez, elles ne correspondent probablement pas exactement à la situation de votre entreprise. Surtout dans les domaines de la gestion des erreurs, ce script n'est pas très utile.

Remarque : Outre les exemples de scripts disponibles sur www.cisco.com, le CD Cisco Unity Express contient un script similaire à celui utilisé ici, appelé `aa_sample1.aef`. Il est très similaire au standard de l'Auto Attendant (avant 2.1). Si vous prévoyez d'utiliser le script en conserve comme point de départ, utilisez ce script au lieu du fichier `aa.aef` qui se trouve lors de l'installation de certaines versions de Script Editor.

Méthodes d'intégration

Cisco Unity Express prend actuellement en charge l'intégration à Cisco CallManager ou Cisco CallManager Express. Bien que le script lui-même ne change pas, la méthode d'intégration peut entraîner de légères différences. Lorsque vous exécutez une étape Redirect, qui transfère logiquement l'appel de Cisco Unity Express vers un poste externe, cette méthode d'intégration est essentielle. Lorsqu'elle est intégrée à Cisco CallManager Express, ce qui est identique pour une intégration Cisco CallManager actuellement en mode SRST, la signalisation est effectuée via SIP sous la forme d'un message BYE/Egalement. Cela signifie qu'une redirection raccroche l'appel à Cisco Unity Express et indique au routeur Cisco CallManager Express/SRST de connecter l'appelant au poste spécifié dans la partie « Egalement : » du message BYE. Si ce poste n'est pas valide ou est inaccessible, l'appelant peut être supprimé. En effet, Cisco Unity Express ne peut pas récupérer cet appel et être averti que le transfert échoue. C'est un vrai transfert aveugle. Si l'appel est redirigé vers un numéro de répertoire non enregistré ou occupé ou un numéro de poste qui ne répond pas, les règles de renvoi d'appels standard qui s'appliquent à ce numéro de répertoire sont en vigueur. Cisco Unity Express ne fait plus partie de l'appel. Cela signifie également que lorsque vous script les étapes de redirection, actuellement il n'y a pas beaucoup d'avantage lorsque vous ajoutez du code pour traiter les redirections occupées/non valides/non réussies, puisque la redirection elle-même est équivalente à une déconnexion. Il n'y a pas de cas d'échec possible.

Pour les intégrations de Cisco CallManager, Cisco Unity Express communique via l'interface JTAPI (Java Telephony Application Programming Interface). Cela permet d'effectuer des étapes de redirection occupées/non valides/non réussies. Par conséquent, ajoutez du code au script afin de gérer ces situations. Lorsque tous les Cisco CallManager deviennent inaccessibles et que le système passe en mode SRST, cela utilise la signalisation SIP décrite précédemment dans ce

document. Par conséquent, toutes les limitations décrites dans ce document s'appliquent.

Exemple de flux d'appels

Avant de commencer avec un script de production, il est impératif de mapper l'ensemble du flux d'appels. Dans cet exemple, supposez que toutes les extensions sont comprises entre 200 et 299 (le point important est qu'elles commencent toutes par 2). Pour cet exemple, vous avez une procédure comme celle-ci :

1. Répondez à l'appel.
2. Vérifiez s'il existe un autre message d'accueil. S'il y en a, jouez-le.
3. Vérifiez si c'est un jour férié. Si c'est le cas, lisez un message d'accueil et passez au menu principal.
4. Vérifiez qu'il est en cours ou en dehors des heures d'ouverture et lisez le message d'accueil correspondant (ouvert/fermé). Passez au menu principal.
5. Le menu principal doit lire une invite et permettre aux utilisateurs d'entrer un poste à tout moment s'ils le connaissent. Ils sont autorisés à entrer 1 s'ils veulent rechercher un utilisateur dans le répertoire, ou 0 s'ils veulent atteindre l'opérateur. Puisque toutes les extensions utilisateur valides commencent par 2, entrez 2 branches dans la section dial-by-extension.
6. Le nom de numérotation recherche les utilisateurs et les transfère vers eux. L'utilisateur est transféré à l'opérateur si zéro est saisi.
7. L'option dial-by-extension tente immédiatement de collecter deux chiffres supplémentaires. Si moins sont entrés, il retourne au menu principal. Si deux sont collectés, il tente de rechercher l'utilisateur. L'utilisateur est transféré vers ce poste si l'utilisateur quitte, sinon il accède au menu principal.
8. Le transfert vers la fonction opérateur transfère l'appel. S'il échoue pour une raison quelconque, il retourne au menu principal.
9. Une séquence de gestion des erreurs supplémentaire pour divers problèmes (tels que plusieurs transferts infructueux) qui demande à l'utilisateur de réessayer ultérieurement et de raccrocher.

Beaucoup peut être fait pour la gestion des erreurs. Pour une mise en oeuvre réelle, il est probablement utile de cartographier l'ensemble du flux d'appels (ceci inclut la gestion des erreurs et chaque interaction ou choix utilisateur) dans un organigramme.

Exemple de script

Démarrer

Entrez le script dans Script Editor.



Tous les scripts commencent par une étape Start. La première chose à faire est d'accepter l'appel.

Après cela, lancez une invite (DP[500]) qui est une pause de 0,5 seconde. Dans certains cas, la transmission audio directe au réseau téléphonique public commuté (RTPC) ne se produit pas rapidement. Jouez quelque chose avant et un message d'accueil « réel » afin d'éviter toute coupure perçue au message d'accueil. L'étape suivante est On Exception Goto. Ce que cela fait, c'est rechercher toute exception de type « UndefinedPromptException » au moment de l'exécution du script. Cette exception particulière se produit lorsque l'étape tente de lire une invite qui n'existe pas. Cela se produit uniquement si la propriété Continuer sur PromptErrors est définie sur Non pour l'étape qui lit l'invite. Si cette valeur est Oui, aucune invite n'est lue et aucune exception n'est levée. L'utilisation de cette étape Aller à l'exception garantit que si une invite critique administrable par l'administrateur (par exemple le menu principal) est effacée, vous pouvez vous diriger vers une section où vous pouvez vous connecter à l'étiquette « Désolé » afin de demander à l'utilisateur de réessayer plus tard. Ensuite, appelez un sous-flux, checkAltGreet.aef. C'est un script système, qui est toujours là. Il vérifie l'existence du fichier AltGreeting.wav. S'il existe, il le lit. Si ce n'est pas le cas, le sous-flux retourne. C'est pratique car vous pouvez l'utiliser en conjonction avec le GMS (maintenant appelé système AVT) afin de jouer un message d'urgence d'une sorte avant le message normal. Pour plus d'informations sur cette fonctionnalité, référez-vous à [Configuration et utilisation du système de gestion des messages d'accueil et du message d'accueil secondaire d'urgence pour Unity Express](#). Il n'y a aucun moyen de savoir si le message d'accueil secondaire est réellement lu (ou de vérifier l'existence d'un fichier dans le référentiel local). Vous n'avez pas besoin d'utiliser ceci au début d'un script. En fonction de votre application, vous pouvez utiliser la fonctionnalité de message d'accueil secondaire d'urgence afin d'enregistrer des invites de jours fériés, par exemple. Au lieu d'appeler ce sous-flux, appelez-le uniquement s'il s'agit d'un jour férié. Cela permet aux administrateurs du site d'enregistrer les messages d'accueil de jours fériés via l'interface utilisateur de téléphonie sans avoir besoin d'accéder au système via un navigateur ou l'interface de ligne de commande (pour renommer les invites ou les affecter à des variables de script).



Choisissez l'invite d'ouverture à lire. S'il s'agit d'un jour férié, lisez le message d'accueil des jours fériés. Si c'est pendant les heures d'ouverture (configurables par le biais de plannings dans l'administrateur Web de Cisco Unity Express), lisez le message d'accueil d'ouverture (qui peut être aussi simple que « Merci d'avoir appelé Cisco Systems »), sinon lisez le message d'accueil fermé. Quel que soit celui que vous jouez, vous allez également directement au menu principal.

Remarque : Les étapes « Est un jour férié » et « Heures d'ouverture » sont spécifiques à la version 2.1.1 et aux versions ultérieures. Pour les versions antérieures de Cisco Unity Express, la seule option est d'utiliser l'étape « Heure du jour ». Cela ne vous permet pas de lire des messages d'accueil différents en fonction du jour ou de la date. Il est basé uniquement sur l'heure.

Menu principal



Saisissez la section Menu principal. Comme une grande partie de la gestion des erreurs envoie ultérieurement l'appelant au menu principal, un compteur est ajouté. Le compteur « tentatives » est comparé à « MaxRetries », une autre variable configurable par l'utilisateur (3 par défaut). Si ce compteur est atteint, présentez vos excuses et transférez l'appelant à l'opérateur (« Désolé que vous ayez des problèmes. Veuillez patienter pour un opérateur. »).

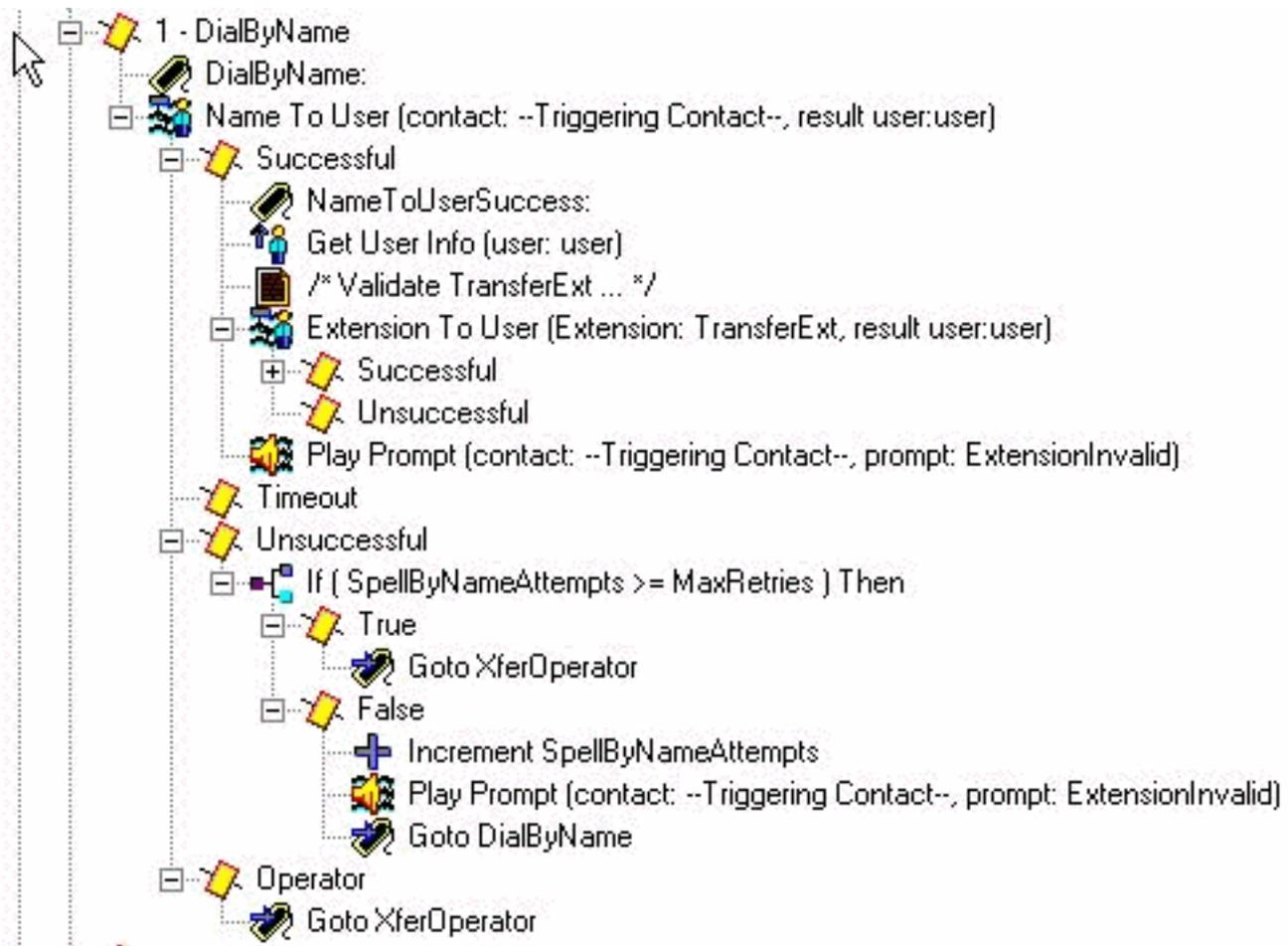
L'étape Menu émet une invite de base (« Si vous connaissez le poste de votre interlocuteur, veuillez le composer à tout moment. Pour spell-by-name appuyez sur 1, pour un opérateur appuyez sur 0, pour répéter ces options, appuyez sur 9. »). Dans ce cas, puisque vous jouez votre propre invite enregistrée, vous modifiez le paramètre Maximum Retries de cette étape sur 0. Cela se fait partout où vous avez la possibilité de le faire. La raison est que si c'est autre que 0, vous entendez l'invite système « êtes-vous toujours là ? » chaque fois que le compteur d'expiration est atteint et qu'il se branche à la branche Délai d'expiration. La branche Échec est atteinte lorsque quelqu'un appuie sur une valeur autre que 1, 2, 9 ou 0. Dans ce cas, vous lisez une invite « L'extension entrée n'est pas valide » et quittez l'étape où vous incrémentez le compteur de tentatives et revenez au menu principal.

Remarque : Afin de vous assurer que le son s'arrête dès que vous entrez un chiffre sur les étapes du support (par exemple Menu ou Invite de lecture), vous devez cocher le champ **Intervenir** dans l'onglet Invite de l'étape. Le champ Interruptible (sous l'onglet Général) est utilisé pour d'autres interruptions d'étape qui ne sont pas actuellement applicables à Cisco Unity Express.

Une seule section de gestion des erreurs est ajoutée. La section « Désolé » lit un message (« Nous ne pouvons pas transférer l'appel pour le moment. Veuillez réessayer ultérieurement. Au revoir. »). L'appel se déconnecte ensuite.

Vous pouvez toujours utiliser un message d'accueil système au lieu de ces variables personnalisées. Les invites système ont toujours la notation SP[] tandis que les invites utilisateur utilisent la notation P[]. Vous ne pouvez pas affecter une invite système à une variable d'invite lorsque vous la définissez (dans le volet des variables, car il s'agit de valeurs utilisateur par définition). Une fois défini, dans le script, vous pouvez affecter une invite système à une variable à l'aide de l'étape Set (ou de l'une des étapes Create Prompt).

Composer par nom



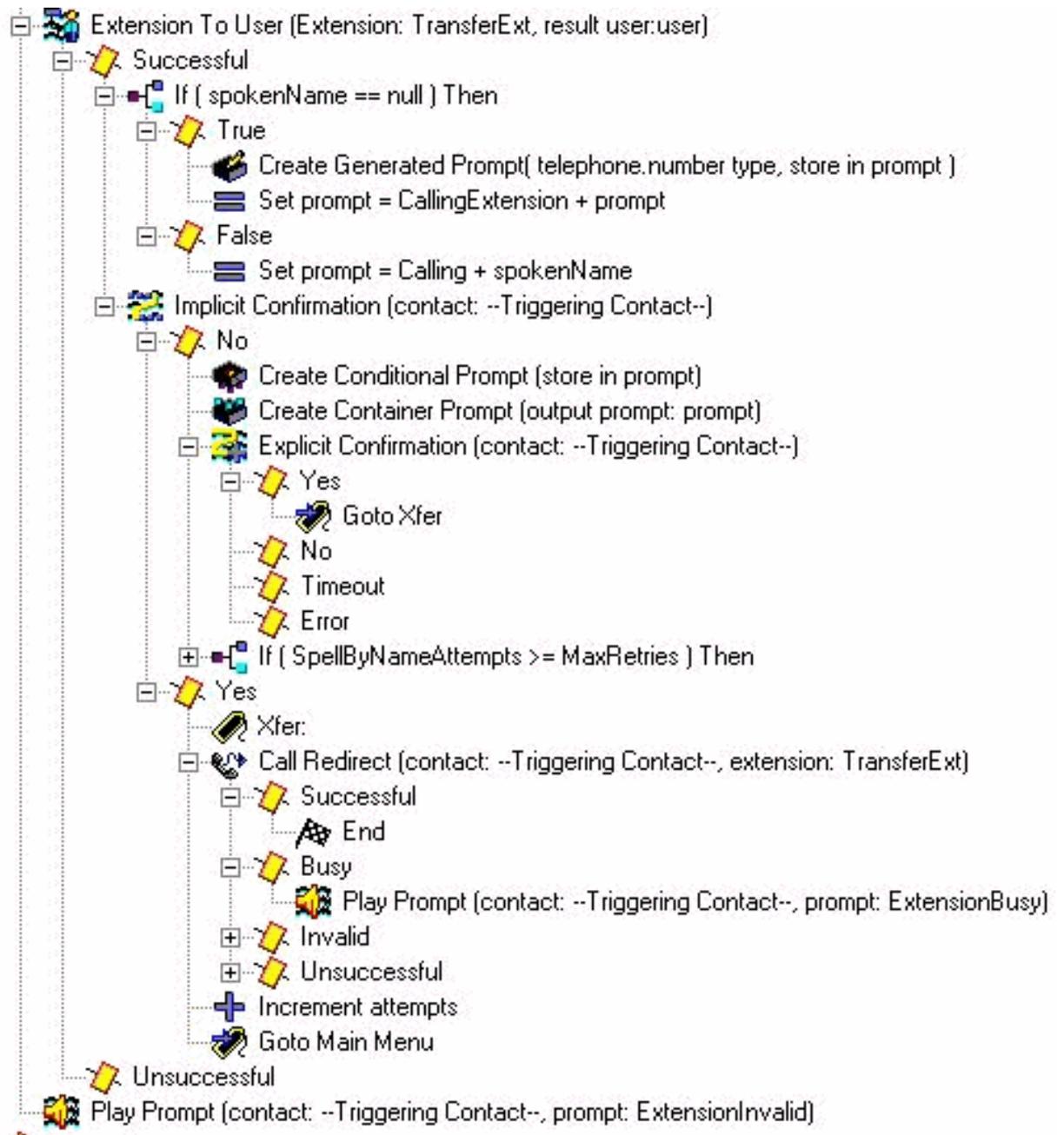
Une façon de procéder consiste à télécharger l'exemple (S4_DialByName) à partir de Cisco.com, à le charger en tant que script distinct dans Cisco Unity Express, puis à ajouter une étape de Subflow d'appel pour l'appeler. Dans ce cas, l'étape est placée directement dans le script et est traitée en détail. L'étape NameToUser vous permet uniquement de configurer l'une de ses invites (« Épelez le nom de la personne que vous voulez appeler, suivi du prénom de la lettre Q appuyez 7 et de la lettre Z appuyez 9 »). Il n'est pas possible d'éliminer toutes les invites système. Afin de les minimiser, vous devez d'abord désactiver la touche Annuler (par défaut *). S'il est présent, le système ajoute un message « pour recommencer, appuyez sur étoile » à la fin de l'invite enregistrée. Ensuite, modifiez Maximum Retries à zéro, de sorte que les messages « Veuillez réessayer » et « Êtes-vous toujours là » ne puissent pas se produire. Suivez toujours le nombre d'essais avec une variable de compteur distincte que vous ajoutez. Il reste une invite système qui ne peut pas être supprimée. Il s'agit de l'invite de plusieurs correspondances (« Plusieurs noms ont été trouvés. Sélectionnez l'une des options suivantes... »). L'étape Nom de l'utilisateur se termine toujours par Succès, Délai d'attente, Échec ou Opérateur (si sélectionné dans les propriétés de l'étape). À l'exception de la branche Succès et opérateur, vous quittez l'étape Menu. Cela signifie que vous incrémentez le compteur de tentatives et revenez au menu principal. En fonction de la situation, vous pouvez disposer d'un compteur de nouvelle tentative distinct et ajouter une étape pour revenir à l'étiquette DialByName afin qu'elle répète l'action déjà

sélectionnée.

Remarque : Un défaut de la version 2.1.1 qui nécessite une attention particulière est l'ID de bogue Cisco [CSCeg81385](#) (clients [enregistrés](#) uniquement). Lorsque le nombre maximal de tentatives de l'étape NameToUser est défini sur zéro ou un, vous devez utiliser un caractère de fin. Sans le caractère de terminaison, il ne va pas à la branche « Succès » de cette étape. Au lieu de cela, il va toujours à « Timeout », même s'il y a une correspondance.

Si l'utilisateur n'entre pas de poste pouvant être mappé à un nom, la branche Échec vérifie un nouveau compteur (SpellByNameAttempts) par rapport à MaxRetries. Supposez que le même nombre maximal de tentatives pour entrer dans le menu principal doit s'appliquer à la fonction spell-by-name. Vous pouvez ajouter une variable distincte si elle doit être différente.

Une étape NameToUser réussie signifie qu'au moins un chiffre est entré et que le nom d'un utilisateur unique tel que défini dans Cisco Unity Express est sélectionné (la branche Timeout n'est atteinte que si aucun chiffre n'est entré). Afin de vous assurer que le poste est unique, vérifiez que vous pouvez mapper le poste à l'utilisateur. Cela peut être réalisé par l'étape ExtensionToUser. Cette étape est ajoutée dans Cisco Unity Express 2.1.1. Auparavant, il n'y avait aucun moyen, avec une extension, de s'assurer qu'il était mappé à un utilisateur réel dans le système de messagerie vocale. Le mieux que vous puissiez faire était de configurer certaines variables à comparer, par exemple avec l'ajout d'une étape « If (TransferExt < MinExtension) || (TransferExt > MaxExtension)... » ou quelque chose de similaire.

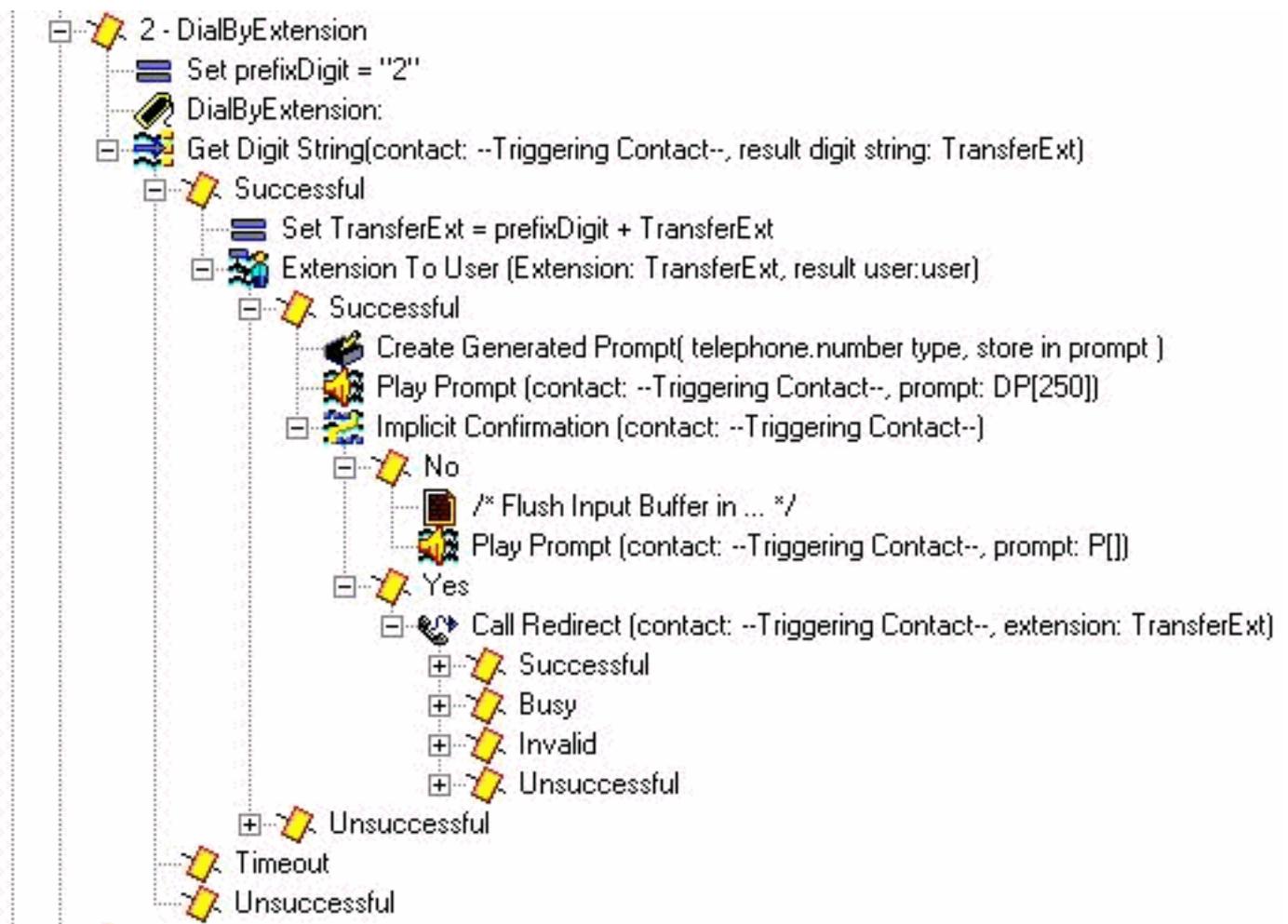


Ici vous voyez le script après avoir mappé le poste à l'utilisateur. S'il ne le fait pas, alors il lance une invite indiquant que le poste n'est pas valide. En général, il n'y a pas de prolongation pour une raison quelconque. Dans le cas de Cisco CallManager Express, cela peut signifier que vous raccrochez si vous essayez de transférer l'appel. Cela peut créer une légère confusion puisque l'appelant a déjà sélectionné un utilisateur. Cependant, c'est clairement déjà une condition d'erreur que vous voulez savoir de toute façon. Si le mappage du poste à un utilisateur réussit, vérifiez d'abord s'il a un nom enregistré (la variable `voiceName` est remplie lorsque vous recherchez cet utilisateur). Définissez ensuite la variable d'invite sur **Calling** plus le nom vocal. Sinon, définissez-la sur **Poste appelant** plus le poste orthographié de l'utilisateur. Vous pouvez également spécifier le nom (lire `S[name]` dans l'invite). Aucun nom enregistré à lire. L'orthographe d'un système, telle que l'orthographe d'un nom ou d'une extension, est toujours effectuée dans la voix du système et ne peut pas être personnalisée. L'étape de confirmation implicite lit l'invite que vous venez de créer. Si l'utilisateur ne tape rien, il entre dans la branche Oui et redirige l'appel. En cas de problème, dans le cas de l'intégration de Cisco CallManager, il diffuse un message d'accueil et l'appelant est renvoyé au menu principal. Si l'appelant entre quelque chose, générez une autre invite. Cette fois, commencez l'invite par le nom vocal ou une version orthographiée du nom de l'utilisateur (étape Créer une invite conditionnelle). Suivez cette procédure par une pause et une invite « si c'est le nom de la personne que vous appelez, appuyez sur 1 ». Maintenant, l'utilisateur

peut appuyer sur 1 pour yes, 2, * ou attendre. Le système les renvoie soit à la section spell-by-name, soit à l'opérateur. Cela dépend du nombre de fois qu'ils sont passés par le système.

Composer par poste

Le premier chiffre (« 2 » dans cet exemple) est déjà collecté dans la section dial by extension. Définissez la variable prefixDigit sur 2. Vous pouvez facilement changer ceci, en fonction du chiffre de début (ou s'il y a plus d'un chiffre de début possible). Aucun chiffre de début ne peut être identique à une autre option du menu (par exemple 1, que vous utilisez pour composer un numéro par nom dans cet exemple). La chaîne Get Digit collecte également un nombre fixe de chiffres (deux dans ce cas). Par conséquent, le nombre total de chiffres doit être fixé. Si les postes valides commencent par 2 ou 3, vous pouvez ajouter une étape « Set prefixDigit = « 3 » » à l'option 3 de l'étape Menu suivie d'une étape Goto DialByExtension.

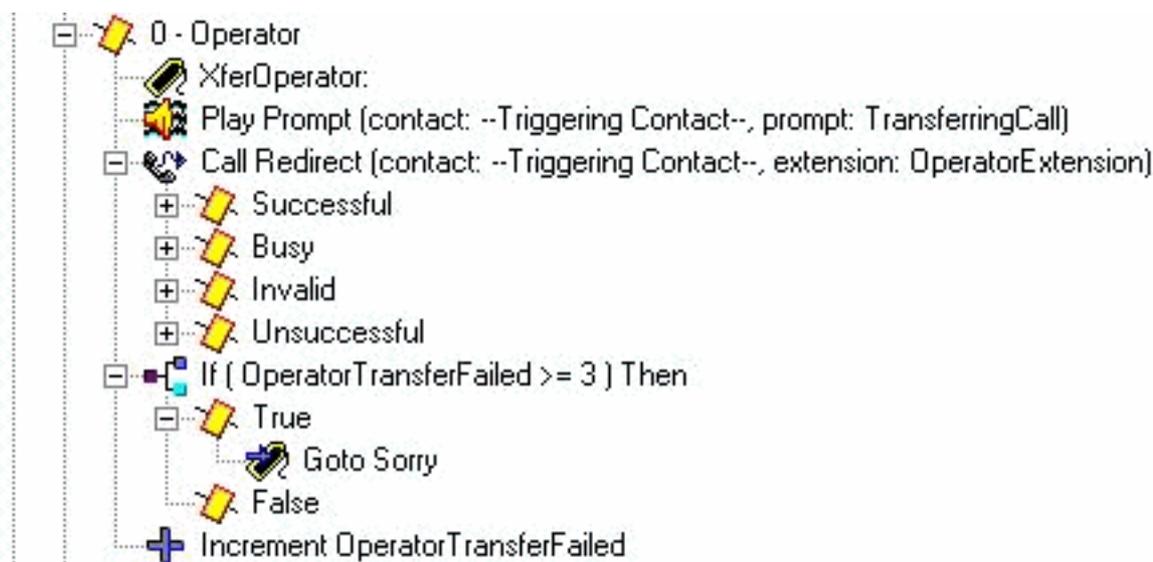


Puisque vous avez des extensions à trois chiffres dans cet environnement, le paramètre Get Digit String step Input length est égal à 2. Il n'y a aucun caractère de fin et aucune clé d'annulation. Le nombre maximal de tentatives est 0. Si un utilisateur n'entre pas de poste valide, le script incrémente le nombre de tentatives et retourne au menu principal. L'étape ne lit aucun son (l'invite est P[]) car l'utilisateur est déjà au milieu des chiffres de composition.

Lorsque deux chiffres sont correctement collectés, ajoutez le préfixe prefixDigit (« 2 » dans ce cas) qui est déjà composé (ce qui vous permet de vous connecter à partir de l'étape Menu). Le TransferExt résultant passe ensuite à l'étape Extension To User, qui vérifie s'il s'agit d'un utilisateur valide. Si c'est le cas, créez une invite avec le poste spécifié. Dans l'étape de confirmation implicite, lisez l'invite « poste appelant » suivie du poste. Recherchez l'utilisateur et lisez le nom vocal, comme vous l'avez fait dans DialByName. Toutefois, dans ce cas, il n'est pas

nécessaire. Si aucun chiffre n'est composé, supposez qu'il s'agit du numéro de poste que l'appelant veut joindre et redirigez l'appel. Sinon, revenez au menu principal. L'étape Play Prompt DP[250] a pour but de purger toute autre entrée, car l'étape Get Digit String ne comporte aucun caractère de terminaison configuré. L'idée est que vous voulez que cela fonctionne si quelqu'un compose le numéro de poste « 200 » ou « 200# ». Si le # est saisi pendant l'exécution de l'étape de confirmation implicite, il passe à la partie Non de cette étape. Dans certains cas, il n'est probablement pas nécessaire d'avoir une étape de confirmation implicite, mais plutôt de rediriger l'appel immédiatement. À titre d'illustration, ces étapes sont laissées en place.

Transfert opérateur

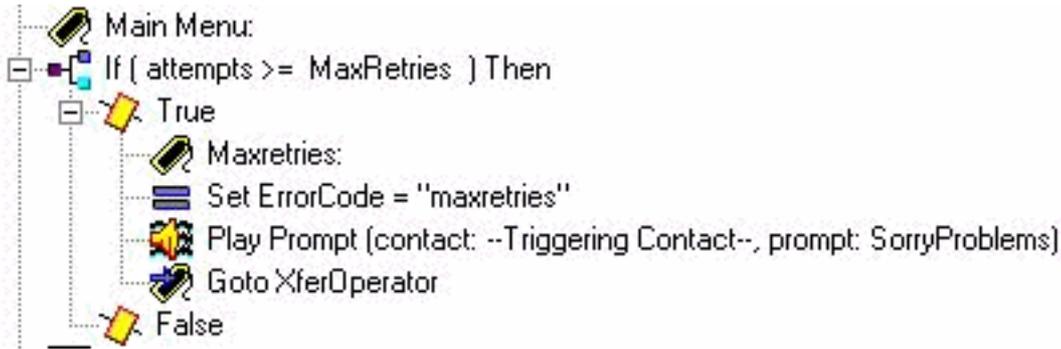


Lancez une invite « Transfert d'appel », puis redirigez-la vers le poste de l'opérateur configuré. Si elle échoue pour une raison quelconque, vous avez un compteur séparé que vous incrémentez et revenez au menu principal. La raison en est que vous voulez avoir une méthode pour donner à l'opérateur un certain temps pour décrocher du téléphone ou se réenregistrer afin que vous puissiez essayer de les transférer à nouveau. Après un certain nombre de tentatives (codées en dur à 3 dans cet exemple), vous passez à l'étiquette « Désolé », qui abandonne l'appel.

Gestion avancée des erreurs

Il existe de nombreux endroits où les scripts peuvent échouer. L'une des zones les plus communes d'un script bien conçu est lorsque les invites utilisateur sont manquantes. Lorsque le système tente de lire une invite qui n'existe pas, un certain nombre de choses peuvent se produire. Tout d'abord, si l'étape qui lit l'invite a la valeur « Continuer sur les erreurs d'invite » sur Oui. Il passe à l'étape suivante et ne lit pas l'invite. Si la valeur Non est affectée aux erreurs de poursuite à l'invite, une exception est générée. La plupart du temps, il s'agit de l'« UndefinedPromptException », ce qui signifie que l'invite est introuvable ou n'est jamais configurée. Lorsque vous avez une étape On Exception Goto, elle peut vous permettre d'attraper cette exception et de vous connecter à une section de code qui peut soit lire un message système, soit utiliser une autre forme audio à la place. Puisque l'étape On Exception Goto s'applique à tous les fichiers audio qui sont lus dans l'ensemble du script (vous ne pouvez pas définir une étape On Exception Goto distincte pour chaque invite que vous lisez), la meilleure façon de gérer cela est de définir une variable de code d'erreur immédiatement avant de lire l'invite. De cette façon, vous pouvez vous brancher quelque part, prendre une action en fonction de ce code d'erreur, et peut-être revenir en arrière.

Par exemple, vous pouvez modifier le script avec ceci :



Vous avez ajouté le label « Maxretries ». C'est utilisé pour retourner au même endroit. Définissez une variable ErrorCode sur **Maxretries**. À l'étape Lecture de l'invite, assurez-vous que la valeur **Non** est **affectée** aux erreurs de la suite de l'invite. Ajoutez à présent une section qui peut gérer ce problème. Vérifiez si le ErrorCode correspond à « Maxretries », puis définissez la variable DésoléProblems (que vous avez tenté de lire précédemment) sur le message système **Désolé d'avoir des problèmes. Veuillez rester en ligne et certains seront bientôt avec vous**. Ensuite, accédez à l'étiquette Maxretries pour qu'il puisse jouer.



La dernière étape consiste à modifier le message On Exception Goto pour rechercher l'erreur UndefinedPromptException :



Cette étape peut être n'importe où dans le script. Cependant, il est généralement placé au début ou près de celui-ci. Plusieurs étapes permettent de lire des invites. Les invites système sont répertoriées au bas de ce document.

Cette méthode peut être utilisée au lieu d'appeler le sous-flux checkAltGreet.aef. Vous devez lire le fichier AltGreeting.wav. En cas d'erreur, gérez-la. L'avantage est que vous savez quand il joue ou non.

Bien que moins courantes, les autres exceptions d'invite sont les suivantes :

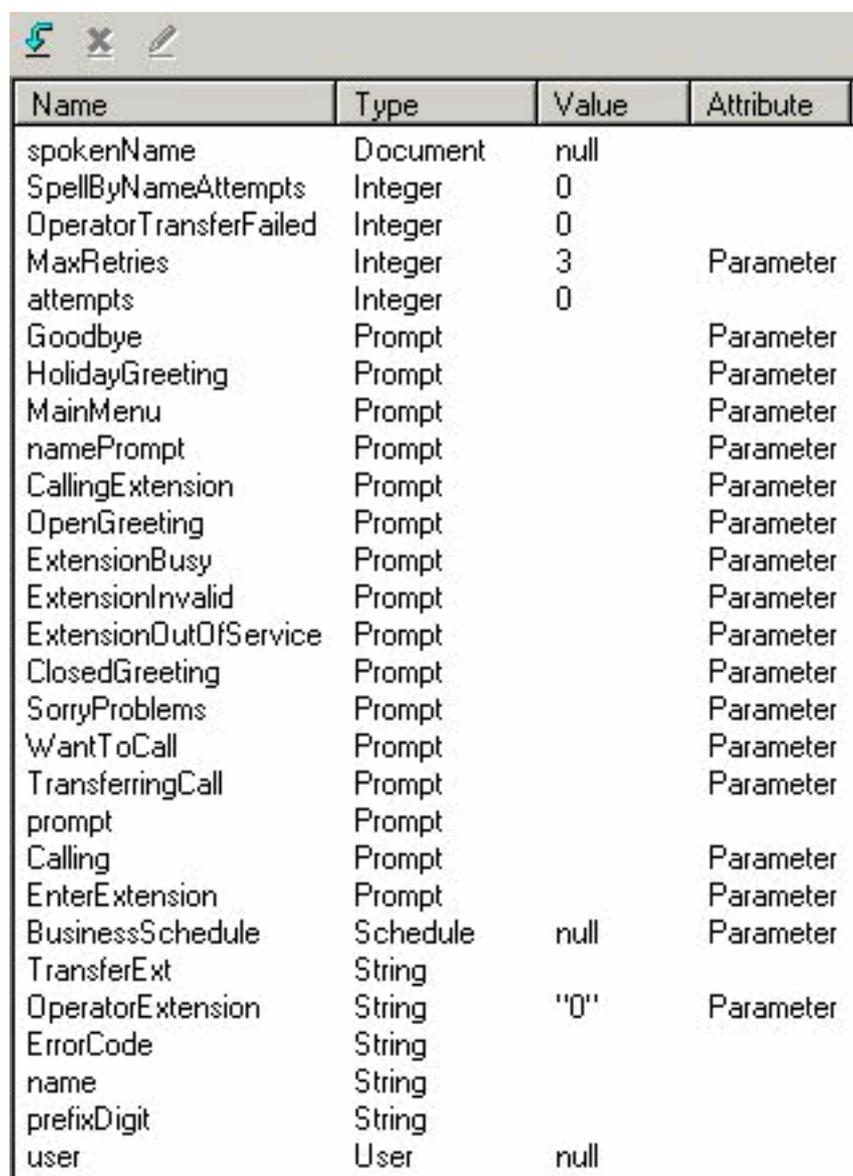
- ExceptionInviteNonDéfinie
- PromptException
- GénérateurInviteNonDéfini
- InvalidPromptArgumentException
- ExpressionInviteNonPriseEnCharge

[Transférer un appel directement vers la messagerie vocale](#)

Il est actuellement impossible de rediriger l'appel directement vers une boîte vocale. Il doit être envoyé à un numéro qui transmet à la messagerie vocale. Il peut s'agir d'un numéro fictif dans Cisco CallManager Express (ou même dans Cisco CallManager) configuré pour renvoyer tous les appels définis sur le numéro pilote de la messagerie vocale. Utilisez les paramètres de numéro Cisco Unity Express/e.164 afin de faire correspondre ce numéro à un abonné ou à un groupe. [Transférer un appelant directement dans une boîte aux lettres Unity Express](#) explique comment procéder.

[Variables de script](#)

Vous devez créer toutes les variables utilisées dans un script avant de pouvoir les utiliser. Si vous en utilisez un et que vous le supprimez ultérieurement, sélectionnez **Outils > Valider** dans Script Editor afin de le trouver. Pour cet exemple de script, il s'agit des variables utilisées. Notez le nom, le type, la valeur (qui est initialement la valeur de la variable) et l'attribut. L'attribut Paramètre signifie que la variable est exposée à l'administrateur via la page Web. Tout ce que vous voulez que l'administrateur Cisco Unity Express personnalise doit avoir l'attribut Parameter.



Name	Type	Value	Attribute
spokenName	Document	null	
SpellByNameAttempts	Integer	0	
OperatorTransferFailed	Integer	0	
MaxRetries	Integer	3	Parameter
attempts	Integer	0	
Goodbye	Prompt		Parameter
HolidayGreeting	Prompt		Parameter
MainMenu	Prompt		Parameter
namePrompt	Prompt		Parameter
CallingExtension	Prompt		Parameter
OpenGreeting	Prompt		Parameter
ExtensionBusy	Prompt		Parameter
ExtensionInvalid	Prompt		Parameter
ExtensionOutOfService	Prompt		Parameter
ClosedGreeting	Prompt		Parameter
SorryProblems	Prompt		Parameter
WantToCall	Prompt		Parameter
TransferringCall	Prompt		Parameter
prompt	Prompt		
Calling	Prompt		Parameter
EnterExtension	Prompt		Parameter
BusinessSchedule	Schedule	null	Parameter
TransferExt	String		
OperatorExtension	String	"0"	Parameter
ErrorCode	String		
name	String		
prefixDigit	String		
user	User	null	

Si vous ne voulez pas enregistrer toutes ces invites personnalisées, vous pouvez les modifier et décocher l'attribut **Paramètre** afin qu'il ne soit pas affiché via la page Web. Afin de s'assurer que le système fonctionne toujours, ajoutez plusieurs étapes Set au début du script. Par exemple, définissez 'Calling' sur 'SP[AA/AACalling]'. Toutes les invites affichées ici ne comportent pas de

scripts système correspondants. Dans la plupart des cas, il est logique de les enregistrer vous-même. Si les invites ne changent pas, vous pouvez les enregistrer une fois, vous assurer qu'elles existent sur Cisco Unity Express System avec un nom de fichier donné, puis supprimer l'attribut Parameter de la variable. Ainsi, l'invite lit un nom de fichier fixe. Mais la variable n'est pas exposée via l'interface Web où un administrateur peut la modifier accidentellement.

[Télécharger un script vers Cisco Unity Express](#)

Généralement, la première étape pour ajouter une nouvelle application et un nouveau script à Cisco Unity Express consiste à télécharger les invites. Ceci est important car les invites doivent exister sur le système pour que vous puissiez les configurer dans les paramètres du script. Sinon, vous devez laisser les paramètres vides, télécharger les invites, puis revenir en arrière et modifier les paramètres du script. Pour plus d'informations sur l'enregistrement et le téléchargement des invites, reportez-vous à [Administration via le système de gestion des messages d'accueil/téléphone \(AVT\)](#).

Une fois votre script enregistré et prêt à l'essayer, vous devez d'abord le télécharger dans Cisco Unity Express. Le moyen le plus simple est d'utiliser l'interface utilisateur graphique, car cela ne nécessite pas de serveur FTP pour charger les fichiers de script à partir de.

Procédez comme suit :

1. Connectez-vous à l'interface utilisateur graphique de Cisco Unity Express avec un compte disposant de droits d'administration.
2. Sélectionnez **Voice Mail > Auto Attendant**.
3. Cliquez sur **Add**.
4. En regard du script de réception automatique sélectionné, cliquez sur le bouton **Télécharger**.
5. Cliquez sur **Browse**.
6. Recherchez le fichier de script, sélectionnez-le et appuyez sur **OK**.
7. Si vous voulez modifier le nom du fichier de destination, faites-le. Sinon, appuyez sur **Télécharger**.
8. Saisissez un nom d'application. Il peut s'agir de n'importe quoi en minuscules et sert à faire référence à l'application associée à ce fichier de script.
9. Cliquez sur **Next** (Suivant).
10. La page affiche maintenant toutes les variables du script marquées comme paramètres. Cela signifie qu'ils peuvent être configurés sur cette page. Si vous n'avez pas encore téléchargé les fichiers d'invite, vous pouvez d'abord passer en revue cette section et ensuite remplir les informations plus tard.
11. Cliquez sur **Suivant** lorsque vous avez terminé de saisir les paramètres du script.
12. Assurez-vous que l'application est marquée comme **Activé**, configurez le nombre maximal d'appels simultanés (par défaut, le nombre maximal autorisé par la licence) et entrez un numéro d'appel. Il s'agit du numéro appelé pour lequel cette application particulière doit être appelée (déclencheur). Il peut s'agir de votre numéro de réception automatique principal ou temporaire. Vous pouvez également ignorer ceci et le remplir plus tard. Sans numéro d'appel, vous ne pouvez pas tester ou lancer un script. **Remarque** : pour les intégrations de Cisco CallManager avec prise en charge SRST, vous devez saisir plusieurs numéros d'appel. Avant Cisco Unity Express 2.1.1, ces informations devaient être entrées via l'interface de ligne de commande. Dans Cisco Unity Express 2.1, plusieurs numéros d'appel (déclencheurs) peuvent être saisis à partir de l'écran **Administration > Call-in Numbers**.

13. Cliquez sur **Terminé**.

Une fois que vous avez téléchargé le script et configuré l'application, vous devez vous assurer que vous pouvez acheminer les appels vers l'application. Cela signifie que Cisco CallManager Express ou Cisco CallManager doivent être configurés. Avec CallManager Express, ceci implique la configuration d'un terminal de numérotation dial-peer qui pointe vers Cisco Unity Express qui correspond au modèle (numéro de déclenchement ou numéro d'appel) configuré dans Cisco Unity Express. Dans Cisco CallManager, un point de routage CTI portant ce numéro doit être ajouté. N'oubliez pas que vous devez associer ce point de routage CTI à l'utilisateur JTAPI auquel Cisco Unity Express se connecte en tant que. Cette configuration peut nécessiter un redémarrage de Cisco Unity Express. Une fois le point de routage ajouté et associé à l'utilisateur, vous pouvez modifier le script autant que vous le souhaitez sans avoir à redémarrer.

Lorsque vous mettez à jour un script, si aucun paramètre de variable n'est ajouté ou modifié, vous pouvez accéder à l'éditeur Cisco Unity Express, cliquer sur **Messagerie vocale > Scripts**, Télécharger et sélectionner le script. Il est remplacé par tous les autres paramètres configurés qui restent identiques.

Lorsque vous créez des invites, la chose la plus facile à faire est d'utiliser le système AVT. Les noms de fichiers qu'il enregistre sont accompagnés de timbres de date/heure. Mais dans Cisco Unit Express 2.1.1 et versions ultérieures, il est possible d'utiliser la commande CLI **ccn rename prompt**. Auparavant, vous deviez télécharger l'invite, la recharger avec un nouveau nom, puis supprimer le fichier .wav d'origine. Exemple :

```
cue-3745-44a> show ccn prompts
```

```
...
Name:                UserPrompt_01032005170038.wav
Language:            en_US
Last Modified Date:  Mon Jan 03 17:00:38 EST 2005
Length in Bytes:     35098
```

```
cue-3745-44a> ccn rename prompt UserPrompt_01032005170038.wav
```

```
MyPrompt.wav language en_US
```

```
Warning! Any existing ccn applications that use this prompt will
stop working until they are manually updated with the new prompt name.
```

```
Continue? (y/n) y
```

```
cue-3745-44a>
```

Dépannage

Lorsque vous dépannez un script, vous devez appeler et tester différents scénarios, à la fois une entrée utilisateur régulière, ainsi que le délai d'attente et les conditions d'erreur anticipées.

Avant de télécharger un script, accédez toujours à l'entrée **Outils > Valider** dans Script Editor et assurez-vous d'avoir le message « Validation réussie ». Si ce n'est pas le cas, toutes les erreurs s'affichent dans le volet de débogage. Vous pouvez double-cliquer dessus et cela vous indique où se trouve la condition d'erreur dans le script.

Dans la plupart des cas, les traces par défaut sont plus que suffisantes pour le dépannage des problèmes. Pour simplifier cela, laissez les traces par défaut et utilisez la fonctionnalité de filtrage afin d'afficher ce dont vous avez besoin.

Avant tout test, émettez la commande **clear trace**. Cela efface la mémoire tampon des messages de suivi de sorte que la quantité d'informations filtrées ne comprenne que les données après

l'exécution de la commande.

Pour l'exécution du script, l'élément le plus important à filtrer est EXECUTING_STEP. Affiche toutes les étapes lors de leur exécution. Exemple :

```
cue-3745-44a> clear trace
```

```
cue-3745-44a> /*** NOW THE TEST CALL IS MADE ***/
```

```
cue-3745-44a> show trace buffer long | include EXECUTING_STEP
```

```
3119 12/28 17:05:33.955 ACCN APMG 0 EXECUTING_STEP:Executing a step:
Application=App[name=customaa,type=Cisco Script Application,id=4,
desc=customaa,enabled=true,max=8,valid=true,optional=
[cfgVars=[Lcom.cisco.wfapi.util.WFNameValuePair;@14efaa66,script=customaa21.aef]],
Task id=17,000,000,007,Step id=0,Step Class=com.cisco.wfframework.steps.core.
StepStart,Step Description=Start
...
```

Dans cet exemple, certaines informations redondantes sont supprimées au milieu et rognées au début de l'horodatage. La plupart du temps, les informations les plus importantes se trouvent à la fin de chaque ligne.

Note : Bon nombre de ces lignes de sortie sont ramenées à une deuxième et une troisième ligne pour des raisons spatiales.

```
5:33.956 Step id=529,Step Class=com.cisco.wfframework.steps.core.StepComment,
Step Description=/* Assumptions: User extensions ... */
5:33.957 Step id=3,Step Class=com.cisco.wf.steps.ivr.AcceptStep,Step
Description=Accept (contact: --Triggering Contact--)
5:33.977 Step id=129,Step Class=com.cisco.wf.steps.ivr.OutputStep,Step
Description=Play Prompt (contact: --Triggering Contact--, prompt: DP[500])
5:34.461 Step id=2,190,Step Class=com.cisco.wfframework.steps.core
.StepOnExceptionGoto,Step Description=On Exception(UndefinedPromptException)
Goto Error
5:34.463 Step id=1,649,Step Class=com.cisco.wfframework.steps.core
.StepComment,Step Description=/* Check for Alternate Greeting. ... */
5:34.464 Step id=1,650,Step Class=com.cisco.wfframework.steps.core
.StepCallSubflow,Step Description=Call Subflow -- checkAltGreet.aef
5:34.467 Step id=0,Step Class=com.cisco.wfframework.steps.core.StepStart,
Step Description=Start
5:34.468 Step id=78,Step Class=com.cisco.wf.steps.ivr.GetContactInfoStep,
Step Description=Get Contact Info (contact: --Triggering Contact--)
5:34.469 Step id=79,Step Class=com.cisco.wfframework.steps.core
.StepCreateHost,Step Description=aType = new com.cisco
.aesop.AltGreetType(language)
5:34.473 Step id=56,Step Class=com.cisco.wfframework.steps.core
.StepHost,Step Description=pExist = aType.isEnabled()
5:34.477 Step id=5,Step Class=com.cisco.wfframework.steps.core
.StepIf,Step Description=If ( pExist == true
5:34.478 Step id=3,Step Class=com.cisco.wfframework.steps.core
.StepEnd,Step Description=End
5:34.480 Step id=510,Step Class=com.cisco.wf.steps.ivr.HolidayStep,
Step Description=Is Holiday (Date: --Current Date--)
5:34.487 Step id=512,Step Class=com.cisco.wf.steps.ivr.BusinessHoursStep,
Step Description=Business Hours (date: --Current Date--,
time: --Current Time--, schedule: BusinessSchedule)
5:34.527 Step id=1,659,Step Class=com.cisco.wfframework.steps.core
.StepLabel,Step Description=Office Open:
5:34.529 Step id=1,660,Step Class=com.cisco.wf.steps.ivr.OutputStep,
```

```

Step Description=Play Prompt (contact:
--Triggering Contact--, prompt: OpenGreeting)
5:35.722 Step id=1,669,Step Class=com.cisco.wfframework.steps.core
.StepLabel,Step Description=Main Menu:
5:35.723 Step id=732,Step Class=com.cisco.wfframework.steps.core
.StepIf,Step Description=If ( attempts >= MaxRetries ) Then
5:35.724 Step id=2,195,Step Class=com.cisco.wfframework.steps.core
.StepAssign,Step Description=Set ErrorCode = "mainmenu"
5:35.726 Step id=2,259,Step Class=com.cisco.wf.steps.ivr.MenuStep,
Step Description=Menu (contact: --Triggering Contact--, prompt: MainMenu)
5:35.730 Step id=2,294,Step Class=com.cisco.wf.steps.ivr.ParseInputStep,
Step Description=Get Digit String(contact:
--Triggering Contact--, result digit string: TransferExt)
5:36.197 Step id=2,295,Step Class=com.cisco.wfframework.steps.core
.StepAssign,Step Description=Set TransferExt = "2" + TransferExt
5:36.198 Step id=2,296,Step Class=com.cisco.wf.steps.ivr.ExtensionToAddressStep,
Step Description=Extension To User (Extension: TransferExt, result user:user)
5:36.214 Step id=2,297,Step Class=com.cisco.prompt.steps.CreateGeneratedPromptStep,
Step Description=Create Generated Prompt( telephone.number type, store in prompt )
5:36.229 Step id=2,298,Step Class=com.cisco.wf.steps.ivr.ImplicitConfirmStep,
Step Description=Implicit Confirmation (contact: --Triggering Contact--)
5:41.971 Step id=2,301,Step Class=com.cisco.wf.steps.ivr.RedirectStep,
Step Description=Call Redirect
(contact: --Triggering Contact--, extension: TransferExt)

```

Vous pouvez maintenant voir chaque étape lors de son exécution. Vous ne pouvez pas voir comment les expressions évaluent, ni voir d'entrée utilisateur. Si le script devant vous, vous pouvez suivre où le script se branche, avec la connaissance des chiffres que vous avez entrés (dans ce cas, l'utilisateur compose, compose un numéro de poste et est transféré).

Utilisez un filtre afin de voir les chiffres.

```

cue-3745-44a> show trace buffer long | include "process digit"
3119 12/28 17:05:35.728 ACCN CMTS 0 process digit 2 tag=2
3119 12/28 17:05:35.943 ACCN CMTS 0 process digit 0
3119 12/28 17:05:36.195 ACCN CMTS 0 process digit 1

```

Ce résultat indique que le poste 201 est composé.

Si les versions de CUE et de l'éditeur de script CUE sont toutes deux différentes et que vous essayez de télécharger un script AA via une interface utilisateur graphique ou CLI, le téléchargement échoue avec le message d'erreur `Upload fail`. Si les deux versions sont différentes, vous devez recréer le script à l'aide de l'éditeur CUE de la version identique à votre CUE. Vous pouvez également mettre à niveau le CUE vers la version de l'éditeur de script CUE et exécuter le script.

[Annexe](#)

[Annexe 1 - Instructions système](#)

Ce tableau affiche les invites système que vous pouvez utiliser.

Remarque : Vous ne pouvez pas réutiliser toutes les invites que vous entendez dans l'application de messagerie vocale Cisco Unity Express dans un script personnalisé. Certaines invites ne sont disponibles que via le script de messagerie vocale que vous ne pouvez pas personnaliser via Cisco Unity Express Script Editor.

Invite système	Audio enregistré
SP[AA/AAWelcome]	Bienvenue dans le service de réception automatique
SP[AA/AAMain Menu]	« Pour entrer le numéro de téléphone de la personne que vous essayez d'atteindre, appuyez sur 1. Pour entrer le nom de la personne que vous essayez d'atteindre, appuyez sur 2. Pour transférer vers l'opérateur, appuyez sur 0. »
SP[AA/AASystemEnter Extn]	« Entrez le numéro de téléphone et appuyez sur la touche #. »
SP[AA/AACalling Extn]	« Poste appelant »
SP[AA/AASystemPhone Reach]	« Numéro de téléphone que vous essayez d'atteindre »
SP[AA/AASystemOut of Service Phone]	« Est actuellement hors service »
SP[AA/AASystemName Dial]	« Épelez le nom de la personne que vous voulez appeler, suivi du prénom. Pour la lettre Q, appuyez sur 7, pour Z appuyez sur 9 »
SP[AA/AACalling]	« Appel »
SP[AA/AASystemSorry]	« Désolé, vous avez des problèmes. S'il vous plaît, restez en ligne et certains seront bientôt avec vous. »
SP[AA/AASystemWant 2Call]	« Si c'est le nom de la personne que vous appelez, appuyez sur 1 pour recommencer, appuyez sur *»
SP[AA/morte_1 à]	« Y es-tu toujours ? »

[Annexe 2 - Réinitialiser les traces par défaut](#)

Afin de réinitialiser le système (Cisco Unity Express 2.1.x et versions antérieures) aux traces par défaut via l'interface de ligne de commande, désactivez d'abord toutes les traces à l'aide de la commande **no trace all**, puis collez ceci dans l'interface de ligne de commande :

```

trace ccn engine debug
trace ccn libldap debug
trace ccn subsystemappl debug
trace ccn managerappl debug
trace ccn managerchannel debug
trace ccn subsystemjtapi debug
trace ccn subsystemsip debug
trace ccn stacksip debug
trace ccn subsystemhttp debug
trace ccn vbrowsercore debug
trace ccn subsystemcmt debug

```

trace ccn libmedia dbug
trace ccn managercontact dbug
trace ccn stepcall dbug
trace ccn stepmedia dbug
trace config-ccn sip-subsystem debug
trace config-ccn jtapi-subsystem debug
trace config-ccn sip-trigger debug
trace config-ccn jtapi-trigger debug
trace config-ccn http-trigger debug
trace config-ccn group debug
trace config-ccn application debug
trace config-ccn script debug
trace config-ccn prompt debug
trace config-ccn miscellaneous debug
trace voicemail database query
trace voicemail database results
trace voicemail database transaction
trace voicemail database connection
trace voicemail database execute
trace voicemail mailbox login
trace voicemail mailbox logout
trace voicemail mailbox send
trace voicemail mailbox save
trace voicemail mailbox receive
trace voicemail mailbox delete
trace voicemail message create
trace voicemail message dec
trace voicemail message delete
trace voicemail message get
trace voicemail message inc
trace webinterface initwizard inittrace ccn engine dbug
trace ccn libldap dbug
trace ccn subsystemappl dbug
trace ccn managerappl dbug
trace ccn managerchannel dbug
trace ccn subsystemjtapi dbug
trace ccn subsystemsip dbug
trace ccn stacksip dbug
trace ccn subsystemhttp dbug
trace ccn vbrowsercore dbug
trace ccn subsystemcmt dbug
trace ccn libmedia dbug
trace ccn managercontact dbug
trace ccn stepcall dbug
trace ccn stepmedia dbug
trace config-ccn sip-subsystem debug
trace config-ccn jtapi-subsystem debug
trace config-ccn sip-trigger debug
trace config-ccn jtapi-trigger debug
trace config-ccn http-trigger debug
trace config-ccn group debug
trace config-ccn application debug
trace config-ccn script debug
trace config-ccn prompt debug
trace config-ccn miscellaneous debug
trace voicemail database query
trace voicemail database results
trace voicemail database transaction
trace voicemail database connection
trace voicemail database execute
trace voicemail mailbox login
trace voicemail mailbox logout
trace voicemail mailbox send
trace voicemail mailbox save

```
trace voicemail mailbox receive
trace voicemail mailbox delete
trace voicemail message create
trace voicemail message dec
trace voicemail message delete
trace voicemail message get
trace voicemail message inc
trace webinterface initwizard init
```

Informations connexes

- [Assistance technique concernant la technologie vocale](#)
- [Assistance concernant les produits vocaux et de communications unifiées](#)
- [Dépannage des problèmes de téléphonie IP Cisco](#)
- [Support technique - Cisco Systems](#)