

# Guide de démarrage rapide des scripts personnalisés Unity Express

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Aperçu](#)

[Utilisez le Script Editor de Cisco Unity Express](#)

[Créez un script de base](#)

[Méthodes d'intégration](#)

[Échantillon Callflow](#)

[Exemple de script](#)

[Téléchargez un script au Cisco Unity Express](#)

[Dépannez](#)

[Annexe](#)

[Annexe 1 - Systèmes invite](#)

[Annexe 2 - Remettez à l'état initial les suivis 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 d'autres détails sur le Cisco Unity Express, référez-vous au [guide de Script Editor de Cisco Unity Express](#). Ce document est destiné pour les administrateurs qui ont besoin d'un guide de base afin de commencer à utiliser Cisco Unity Express Editor et à mettre à jour des scripts dans le produit de Cisco Unity Express.

**Remarque:** Le support de script personnalisé n'est pas disponible par le support technique de Cisco régulier. Pour des questions, les commentaires, et l'aide avec un script personnalisé, envoient un email à [ask-cue-editor@external.cisco.com](mailto:ask-cue-editor@external.cisco.com).

## [Conditions préalables](#)

### [Conditions requises](#)

Le lecteur doit être au courant de la gestion et de la configuration de Cisco Unity Express par l'interface de ligne de commande (CLI) ou le GUI.

Le Script Editor de Cisco Unity Express (CUEEditor2.1.1.exe) et l'exemple de script que ce document utilise (CUE-AA-S6-AASample.zip) sont fournis par le [centre de téléchargement logiciel du Cisco Unity Express 2.1.1](#).

## Composants utilisés

Les informations dans ce document sont basées sur le Cisco Unity Express Version 2.1.1. L'exemple contient les étapes qui sont seulement disponibles dans le Cisco Unity Express 2.1.1. Bien que, plusieurs des principes soient identiques pour des versions antérieures. Des étapes qui sont spécifiques à 2.1.1 sont précisées explicitement dans ce document.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## Conventions

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## Aperçu

Les bateaux de base de produit de Cisco Unity Express avec une partie de messagerie vocale et une réception automatique de base (dans le Cisco Unity Express 2.1.1 et plus tard, là sont deux scripts de réception automatique). Beaucoup de clients trouvent la réception automatique standard trop limitée. Le Cisco Unity Express a un scripteur qui permet la création des scripts personnalisés. Ces scripts personnalisés peuvent être chargés dans le Cisco Unity Express et remplacer (ou fonctionner en plus de) la réception automatique régulière.

**Avertissement :** L'exemple de script affiché ici est pour l'illustration. Il peut être modifié de quelque façon. Le but de ce document est d'illustrer le processus de conception et réalisation d'un script personnalisé. On ne le destine pas pour fournir un script de finition de production. Cependant, si vous choisissez d'utiliser ce script, au minimum, enregistrez chacune des demandes. Vous pouvez écouter les fichiers audios reliés afin d'entendre comment ils retentissent. Mais utilisez la [gestion par l'intermédiaire du système de gestion du téléphone \(AVT\) /Greetings](#).

**Remarque:** À ce moment, il n'est pas possible d'obtenir une copie des scripts système réels que le Cisco Unity Express utilise. Ces scripts ont des étapes que le scripteur régulier ne prend en charge pas. Les scripts système en date du Cisco Unity Express 2.1.1 sont aa.aef, voicebrowser.aef, setmwi.aef, promptmgmt.aef, checkaltgreet.aef, xfermailbox.aef, et aasimple.aef.

Le scripteur est un outil qui te permet pour créer des scripts sur un PC séparé du Cisco Unity Express. Une fois que créés, les scripts sont téléchargés et configurés sur le Cisco Unity Express.

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

- **Étape** — La brique de base pour la création de script. Chaque étape est l'unité exécutable la plus fondamentale, telle que « si » déclaration, « Goto », et ainsi de suite.
- **Script** — Un ou plusieurs étapes qui sont exécutées dans l'ordre. Un script est un fichier avec une extension .aef.
- **Variable** — Ce sont des variables dans un script. Les variables peuvent être de différents types, tels que l'entier, booléen, chaîne, et ainsi de suite.
- **Paramètre** — C'est une propriété d'une variable de sorte que la variable soit exposée à l'administrateur par l'interface web de Cisco Unity Express. Par exemple, si vous avez une variable d'OperatorExtension que vous voulez assigner la valeur 1000 par défaut, il doit parfois être changé à 2000. Dans ce cas, la variable est exposée par l'interface web de sorte que le script entier ne doive pas être chargé dans le Cisco Unity Express de nouveau chaque fois que la valeur doit être changée.
- **Demande** — Un fichier .wav qui peut être lu. Il est téléchargé manuellement dans le système de Cisco Unity Express ou enregistré par la gestion par l'intermédiaire du système du téléphone (AVT) (avant la version 2.1.1 l'AVT se sont appelés le système de gestion de messages d'accueil (GMS)). Toutes les demandes d'utilisateur qui sont téléchargées sont placées dans le même répertoire. Ils sont visibles dans le GUI par la commande de menu de **messaging vocal > de demandes** ou dans le CLI par l'**exposition le ccn incite la** commande. Dans un script, des demandes d'utilisateur sont spécifiées comme P [<promptname>]. Des systèmes invite sont pré-enregistrés et peuvent être utilisés. Ils sont spécifiés comme fournisseur de services [<promptname>]. [L'annexe 1](#) répertorie les systèmes invite disponibles.
- **Application** — C'est le script avec tous les demandes et paramètres complétés. Par défaut, le Cisco Unity Express se transporte avec la messagerie vocale, le système AVT, et une application simple de réception automatique qui tous sont configurés quand vous exécutez l'assistant d'initialisation à la fin d'un installer.
- **Déclencheur** — Le déclencheur indique à Cisco Unity Express qu'une application particulière doit être exécutée. Par exemple, quand vous composez 1000, le système téléphonique (Cisco CallManager ou Cisco CallManager Express) conduit l'appel au Cisco Unity Express. Quand le Cisco Unity Express voit qu'un appel est placé au numéro 1000, il recherche un déclencheur pour cette extension. Les lancements respectifs d'application puis. En d'autres termes, il sait si un appel aux 1000 besoins d'aller à la messagerie vocale, à une réception automatique, ou à autre chose. Vous pouvez avoir de plusieurs déclencheurs à la même application. Il y a un certain nombre de déclencheurs qui sont ajoutés par défaut, tel que des déclencheurs à la messagerie vocale, à l'AVT, et à la réception automatique en boîte. Quand vous comprenez la terminologie, elle facilite l'outil d'administration web de Cisco Unity Express pour comprendre.

De la page Web de Cisco Unity Express, l'option de réception automatique sous la catégorie de messagerie vocale se rapporte aux applications comme définies en terminologie. Chaque entrée contient le numéro associé avec l'application, le script qu'elle met en référence, tous les paramètres qui sont associés avec le script, si l'application est activée, et le nombre de ports associés.

La gestion des appels contient actuellement seulement les nombres associés pour la messagerie vocale, la réception automatique intégrée, et le système AVT. Dans le Cisco Unity Express 2.1 et plus tard, sous des **nombres de gestion > d'émission à ligne ouverte** il y a un affichage supplémentaire tous les déclencheurs, des deux JTAPI (pour le Cisco CallManager) et SIP (pour Cisco CallManager Express ou le mode SRST). Actuellement, la seule manière de voir s'il y a de plusieurs déclencheurs configurés que le point à la même application est avec la commande

émission de **déclencheur de ccn d'exposition** dans le CLI.

L'option de demandes se rapporte aux fichiers audios (.wav) qui existent sur le système.

La sélection de scripts se rapporte aux fichiers .aef que vous pouvez enregistrer sur le système. Actuellement, vous pouvez télécharger jusqu'à quatre scripts en plus des scripts système.

Du CLI, ces informations sont facilement disponibles utilisant les commandes de **ccn d'exposition**. 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 d'**application de ccn d'exposition** affiche le nom d'application, le script qu'il se rapporte, s'il est activé et combien de simultanés appelle il peut manipuler, avec tous les paramètres de script avec leurs valeurs.

Le **ccn d'exposition** incite des listes de commandes toutes les commandes, leur langage, taille et date modifiée.

Les informations de **déclencheur de ccn d'exposition** donnent le nombre (« nom ») aussi bien que l'application qu'il appelle, l'état (activer/), et le nombre de sessions.

## [Utilisez le Script Editor de Cisco Unity Express](#)

Utilisez le [Script Editor](#) (clients [enregistrés](#) seulement) pour créer un nouveau script. Il doit être installé sur un PC distinct.

Au même emplacement sont plusieurs exemples de script. Ces scripts aident à apprendre comment faire des exécutions spécifiques dans un script. Cisco vous recommande d'installer l'éditeur et téléchargent quelques exemples de script afin de se familiariser avec la manière que l'éditeur fonctionne.

Une fois que les lancements de Script Editor, vous peuvent ouvrir un script suivant les indications de ce graphique :

Le volet de palette contient des répertoires avec chaque étape, classée par catégorie basé sur leur principale fonction. De là, l'entrave fait un pas plus d'au volet de conception afin de créer un script. Dans le volet variable, vous pouvez créer et éditer des variables. Après que vous fassiez glisser une étape au volet de conception, vous pouvez cliquer avec le bouton droit sur une étape, sélectionner **Properties**, et personnaliser l'étape et les variables assignées à l'étape. Avant que vous téléchargiez un script au Cisco Unity Express, vous devez toujours le valider d'abord par les **outils > validez** l'option. Tous affichages de sortie erreurs puis dans le volet de debug.

Une des choses les plus utiles à se souvenir quand vous utilisez l'éditeur est la fonction d'aide. Vous pouvez y accéder directement par le menu Help sur la barre d'outils, ou quand vous sélectionnez une propriété particulière d'étape.

## Créez un script de base

Vous pouvez employer le menu Help afin de se familiariser avec certaines des étapes. Dans le Script Editor, faites glisser quelques étapes de la palette au volet de conception et cliquez avec le bouton droit sur un et sélectionnez **Properties**. Cliquez sur alors en fonction le **bouton aide** et obtenez l'aide sur cette étape particulière.

Ce document décrit la création d'un exemple de script. Le document discute également certains des problèmes courants que les gens essaient de résoudre quand ils créent une réception automatique faite sur commande. Le but ici est de créer un script qui répond à un appel, vérifie pour voir s'il y a un message d'accueil de remplaçant de secours et si le temps en cours est des heures de travail régulières ou pas. S'il est fermé, lisez un message d'accueil fait sur commande et envoyez l'appel à l'opérateur. S'il est ouvert, jouez un menu qui permet à l'appelant pour composer le cadran-par-nom de 1 par, 2 pour la cadran-par-extension ou 0 pour un opérateur. L'option de cadran-par-extension devrait seulement transférer l'appel vers une plage spécifiée, aucun nombre que vous pouvez introduire.

**Avertissement :** L'exemple que ce document l'utilise est à des fins d'illustration seulement. Tandis que vous êtes libre pour utiliser Qu'est ce que parties vous aimez, il ne s'assortit pas probablement exactement avec la situation de votre société. Particulièrement dans les zones du traitement des erreurs, ce script est utile pas beaucoup.

**Remarque:** Hormis les exemples de script chez [www.cisco.com](http://www.cisco.com), le CD de Cisco Unity Express contient un script semblable en tant que celui utilisé ici, appelé `aa_sample1.aef`. Il est très semblable au militaire de carrière, expédiant la réception automatique (avant 2.1). Si vous prévoyez d'utiliser le script en boîte comme point commençant, utilisez ce script au lieu du fichier `aa.aef` qui est trouvé quand vous installez quelques versions du Script Editor.

## Méthodes d'intégration

Le Cisco Unity Express prend en charge actuellement l'intégration au Cisco CallManager ou au Cisco CallManager Express. Bien que le script lui-même ne change pas, la méthode d'intégration peut avoir comme conséquence quelques légères différences. Quand vous exécutez une étape de réorientation, qui transfère logiquement l'appel du Cisco Unity Express vers une extension externe, cette méthode d'intégration est essentielle. Une fois intégrée avec Cisco CallManager Express, qui est identique pour une intégration de Cisco CallManager actuellement en mode SRST, la signalisation est faite par le SIP sous forme de message BYE/Also. Ce que ce le moyen est qu'une réorientation arrête l'appel au Cisco Unity Express et indique le routeur du Cisco CallManager Express/SRST connecter l'appelant à l'extension spécifiée dans « également : » partie du message BYE. Si cette extension est non valide ou autrement inaccessible, l'appelant peut être lâché. C'est parce qu'il n'y a aucune manière pour que le Cisco Unity Express reprenne cet appel et d'est annoncé que le transfert échoue. C'est un transfert sans suivi vrai. Si les redirect to d'appel des non enregistrés ou occupé ou l'extension qui ne répondent pas, les règles de transfert d'appels régulières qui appliquent à ce nombre de répertoire sont en vigueur. Le Cisco Unity Express n'est plus une partie de l'appel. Ceci signifie également que quand vous script réorientez des étapes, actuellement il n'y a pas beaucoup d'un avantage quand vous ajoutez le code pour traiter occupé/non valide/infructueux réorienté, puisque la réorientation elle-même est

équivalente à un débranchement. Il n'y a aucun cas de panne possible.

Pour des intégrations de Cisco CallManager, le Cisco Unity Express communique par l'interface de programmation de téléphonie de Javas (JTAPI). Ceci permet occupé/non valide/infructueux réorientent des étapes pour se produire. , Ajoutez par conséquent le code au script afin de manipuler ces situations. Quand tous les Cisco CallManagers deviennent inaccessibles et le système tombe dans le mode SRST, ceci utilise la signalisation de SIP discutée plus tôt dans ce document. Par conséquent, toutes les limites décrites dans ce document s'appliquent.

## Échantillon Callflow

Avant que vous commenciez par un script de production, il est impératif de tracer l'écoulement complet d'appel. Pour cet exemple supposez que toutes les extensions sont dans la plage 200-299 (le point important est qu'elles tout le début avec 2). Pour cet exemple, vous avez quelque chose telle que cette procédure :

1. Répondez à l'appel.
2. Vérifiez s'il y a un message d'accueil alternatif. S'il y a, lisez-le.
3. Vérifiez si c'est des vacances. Si oui, lisez des voeux de vacances et poursuivez au menu principal.
4. Vérifiez s'il a lieu pendant ou après des heures de travail et lisez (ouvrez-vous/fermé) le message d'accueil respectif. Poursuivez au menu principal.
5. Le menu principal doit lire une demande et permettre à des utilisateurs pour écrire une extension à tout moment s'ils la connaissent. On leur permet pour écrire 1 s'ils veulent à la consultation un utilisateur dans le répertoire, ou 0 s'ils veulent atteindre l'opérateur. Puisque tout le début d'extensions d'utilisateur valide avec 2, écrivant 2 branchements à la section de cadran-par-extension.
6. Les utilisateurs et le transfert de consultations de cadran-par-nom leur. L'utilisateur est transféré vers l'opérateur si zéro est écrit.
7. De cadran-par-extension d'option les tentatives immédiatement de collecter deux chiffres supplémentaires. Si moins sont entrés, ils retournent au menu principal. Si deux sont collectés, il tente à la consultation l'utilisateur. L'utilisateur est transféré vers cette extension si les procédures utilisateur, autrement l'utilisateur va au menu principal.
8. Le transfert à la fonction d'opérateur transfère l'appel. S'il échoue pour quelque raison, il retourne au menu principal.
9. Un ordre supplémentaire de traitement des erreurs pour différents problèmes (tels que de plusieurs transferts infructueux) qui indique l'utilisateur essayer de nouveau plus tard et raccroche.

Beaucoup peut être fait pour le traitement des erreurs. Pour une vraie implémentation, il aide probablement à tracer l'écoulement complet d'appel (ceci inclut le traitement des erreurs, et chaque interaction utilisateur ou choix) dans un organigramme.

## Exemple de script

### Début

Écrivez le script dans le Script Editor.

Tous les scripts commencent par une étape de début. La première chose à faire est de recevoir



l'appel. Après cela, lisez une demande (DP[500]) qui est une 0.5 seconde pause. Dans certaines situations, le cut-through sonore au réseau téléphonique public commuté (PSTN) ne se produit pas rapidement. Jeu quelque chose avant et « vrai » message d'accueil afin d'éviter tout découpage perçu au message d'accueil. Est allumée ensuite une étape de Goto d'exception. Ce que ceci fait est recherchent n'importe quelle exception de type « UndefinedPromptException » au moment de l'exécution du script. Cette exception particulière se produit quand les essais d'étape pour jouer une demande qui n'existe pas. Ceci se produit seulement si l'étape qui joue la demande fait continuer « la propriété sur PromptErrors » réglée à non. Si réglé à l'oui, aucune demande n'est jouée et aucune exception n'est jetée. L'utilisation de ceci sur l'étape de Goto d'exception veille que si une demande essentielle qui est administrable par l'administrateur (tel que le menu principal), est effacée, vous pouvez s'embrancher à une section où vous pouvez s'embrancher à l'étiquette « désolée » afin de dire l'utilisateur d'essayer de retour plus tard. Après cela, appelez un subflow, checkAltGreet.aef. C'est un script système, qui est toujours là. Il vérifie l'existence du fichier AltGreeting.wav. S'il existe, il le joue. S'il ne fait pas, le subflow retourne. C'est pratique puisque vous pouvez l'employer en même temps que le GMS (maintenant appelé le système AVT) afin de jouer un message d'accueil de secours d'une certaine sorte avant le message régulier. Pour plus d'informations sur cette caractéristique, référez-vous à [Configuring and utilisant le système de gestion de message d'accueil et le message d'accueil alternatif de secours pour Unity Express](#). Il n'y a aucune manière de dire si la salutation alternative joue réellement (ou vérifie l'existence d'un fichier dans le référentiel local). Vous ne devez pas utiliser ceci dans le début d'un script. Basé sur votre application, vous pouvez employer la fonctionnalité de message d'accueil de remplaçant de secours afin d'enregistrer des demandes de vacances, par exemple. Au lieu d'appeler ce subflow, appelez-le seulement si c'est des vacances. Ceci aide les administrateurs du site à enregistrer des voeux de vacances par l'interface utilisateur de téléphonie sans nécessité d'accéder au système par un navigateur ou le CLI (pour renommer des demandes ou pour les assigner aux variables de script).

Décidez quelle demande d'ouverture à la lire. Si c'est des vacances, jouez les voeux de vacances. S'il a lieu pendant les heures de travail (qui sont configurables par des programmes dans l'administrateur Web de Cisco Unity Express), alors jouez le message d'accueil d'ouverture (qui peut être quelque chose aussi simple en tant que « mercis d'appeler Cisco Systems »), jeu d'autre le message d'accueil fermé. Indépendamment desquels vous jouez, vous allez également alors directement au menu principal.

**Remarque:** « Sont les vacances » et les étapes « de heures de travail » sont spécifiques à 2.1.1 et plus tard. Pour des versions antérieures de Cisco Unity Express, la seule option est d'utiliser l'étape de « heure ». Ceci ne te permet pas pour lire différents messages d'accueil basés sur le jour ou la date. Il est basé seulement sur le temps.

## [Menu principal](#)

Écrivez la section de menu principal. Puisqu'une grande partie du traitement des erreurs envoie plus tard l'appelant de nouveau au menu principal, un compteur est ajouté. Les contre-« tentatives » sont comparées à « MaxRetries », une autre variable qui est configurable par l'utilisateur (3 par défaut). Si ce compteur est atteint, faites des excuses et virez l'appelant sur l'opérateur (« désolé vous avez des problèmes. Tenez s'il vous plaît pour un operator.").

L'étape de menu lit une demande de base (« si vous savez que l'extension de votre interlocuteur la composent s'il vous plaît n'importe quand. Pour la presse 1 de charme-par-nom, pour une presse 0 d'opérateur, pour répéter ces options, presse 9."). Dans ce cas, puisque vous jouez votre propre demande enregistrée, vous changez le paramètre maximum de relances sur cette étape à 0. Ceci t'est fait partout ont une option de faire ceci. La raison est que si elle est quelque chose

autre que 0, vous entendez que le système invite « sont toujours vous là ? » chaque fois que l'échéance temporisateur est atteinte et elle s'embranchement au branchement de délai d'attente. Le branchement infructueux est atteint quand quelqu'un appuie sur n'importe quoi autre que 1, 2, 9, ou 0. dans ce cas, vous jouent une demande « que l'extension écrite est non valide » et la baisse hors de l'étape où vous incrémentez les tentatives parent et retournent au menu principal.

**Remarque:** Pour s'assurer que l'audio arrête dès que vous écrivez un chiffre sur des étapes de medias (telles que le menu ou lisez la demande), vous doit signer le gisement d'**insertion** dans l'onglet prompt de l'étape. Le champ interruptible (sur l'onglet Général) est utilisé pour d'autres interruptions d'étape qui s'appliquent actuellement pas applicable au Cisco Unity Express.

Une seule section de traitement des erreurs est ajoutée. La section « désolée » lit un message (« nous ne pouvons pas transférer l'appel à ce moment. Essayez s'il vous plaît de nouveau plus tard. Goodbye. »). Les débranchements d'appel puis.

Vous pouvez toujours utiliser un message d'accueil de système au lieu de ces variables personnalisées. Les systèmes invite ont toujours la notation de fournisseur de services [] tandis que les demandes d'utilisateur utilisent la notation P []. Vous ne pouvez pas assigner un système invite à une variable prompt quand vous le définissez (dans le volet variable puisque telles sont par définition des valeurs d'utilisateur). Une fois que défini, dans le script vous pouvez assigner un système invite à une variable par l'étape de positionnement (ou l'un des créez les étapes prompts).

## Cadran de nom

Une manière de faire le cadran de nom est de télécharger l'échantillon (S4\_DialByName) provenant de Cisco.com, le charge car un script distinct dans le Cisco Unity Express et ajoute alors une étape de Subflow d'appel pour l'appeler. Dans ce cas l'étape est mise directement dans le script et est discutée en détail. L'étape de NameToUser te permet seulement pour configurer une de ses demandes (« orthographiez le nom de famille de la personne que vous voulez appeler, suivi du prénom pour la presse 7 de la lettre Q, et pour la presse de la lettre Z 9"). Il n'est pas possible d'éliminer tous les systèmes invite. Afin de les réduire, vous le premier besoin de désactiver la touche d'annulation (par défaut \*). S'il est présent, le système ajoute un message « pour recommencer, appuyer sur l'étoile » à la fin de la demande enregistrée. En second lieu, le maximum de modification relance à zéro, de sorte que « essayez s'il vous plaît de nouveau » et « soyez vous là » des messages ne puissent pas se produire toujours. Maintenez toujours le nombre d'essais avec une contre- variable distincte que vous ajoutez. Il reste un système invite qui ne peut pas être éliminé. C'est la demande pour de plusieurs correspondances (« plus d'un nom a été trouvé. Choisissez parmi le suivant... "). Le nom à l'étape d'utilisateur finit toujours comme réussi, le délai d'attente, infructueux, ou l'opérateur (si sélectionné dans les propriétés d'étape). Excepté le branchement réussi et d'opérateur, vous relâchez hors de l'étape de menu. Ceci signifie que vous incrémentez les tentatives parez et retournez au menu principal. Basé sur la situation, vous pouvez avoir un compteur distinct de relance et ajouter une étape pour retourner à l'étiquette de DialByName de sorte qu'elle répète l'action qui est déjà sélectionnée.

**Remarque:** Un défaut dans la version 2.1.1 que la considération exigée est l'ID de bogue Cisco [CSCeg81385](#) (clients [enregistrés](#) seulement). Quand des relances maximum sur l'étape de NameToUser est placées à zéro ou à un, vous devez utiliser un caractère d'arrêt. Sans caractère d'arrêt, il ne va pas au branchement « réussi » de cette étape. Au lieu de cela il va toujours au « délai d'attente », même s'il y a une correspondance.

Si l'utilisateur n'écrit pas une extension qui peut être tracée à un nom, le branchement infructueux



vérifie un nouveau compteur (SpellByNameAttempts) contre MaxRetries. Supposez que le même nombre maximal de relances pour écrire le menu principal doit s'appliquer à la fonction de charme-par-nom. Vous pouvez ajouter une variable distincte si ceux-ci doivent être différents.

Une étape réussie de NameToUser signifie qu'au moins un chiffre est écrit et le nom de famille d'un seul utilisateur comme défini dans le Cisco Unity Express est sélectionné (le branchement de délai d'attente est seulement atteint si aucun chiffre du tout n'est écrit). Dans la commande à être sûre que l'extension est seule, vérifiez que vous pouvez tracer l'extension de nouveau à l'utilisateur. Ceci peut être accompli par l'étape d'ExtensionToUser. Cette étape est ajoutée dans le Cisco Unity Express 2.1.1. Précédemment, il n'y avait aucune manière, donnée une extension, de s'assurer qu'elle a tracé à un utilisateur réel dans le système de messagerie vocale. Le meilleur que vous pourriez faire était d'installer quelques variables pour comparer contre, comme en plus d'une étape « si (TransferExt < MinExtension) || (TransferExt > MaxExtension)... » ou quelque chose semblable.

Voici que vous voyez le script après que vous tracez l'extension de nouveau à l'utilisateur. S'il ne lit pas alors une demande que l'extension est non valide. Habituellement il n'y a aucune extension pour quelque raison. Dans le cas de Cisco CallManager Express, ceci peut signifier que vous raccrochez si vous essayez de transférer l'appel. Ceci peut entraîner la légère confusion puisque l'appelant a déjà sélectionné un utilisateur. Cependant, c'est clairement déjà une condition d'erreurs que vous voulez connaître environ de toute façon. Si la cartographie de l'extension à un utilisateur est réussie, le premier contrôle pour voir s'ils ont un nom enregistré (la variable de spokenName est complété quand vous consultation cet utilisateur). Placez alors la demande variable à **appeler** plus le spokenName. Autrement, placez-le à **appeler l'extension** plus l'extension écrite de l'utilisateur. Vous pouvez également définir le nom (jeu S [nom] dans la demande). Il n'y a aucun nom enregistré à jouer. N'importe quelle orthographe de système, telle que définir un nom ou une extension, est toujours faite dans la Voix du système et ne peut pas être personnalisée. L'étape implicite de confirmation joue la demande que vous avez juste créée. Si l'utilisateur ne tape rien, il avance et écrit « oui » le branchement et réoriente l'appel. S'il y a un problème, dans le cas de l'intégration de Cisco CallManager, elle joue un message d'accueil et l'appelant est renvoyé au menu principal. Si l'appelant écrit quelque chose, générez une autre demande. Cette fois commencent la demande par le spokenName, ou une version écrite du nom d'utilisateur (créez l'étape prompt conditionnelle). Suivez ceci par une pause et une demande « si c'est le nom de la personne que vous appelez, presse 1". Maintenant l'utilisateur peut appuyer sur le 1 par oui, 2, \*, ou attendre. Le système l'un ou l'autre les envoie de nouveau à la section de charme-par-nom ou à l'opérateur. Ceci dépend de combien de fois ils sont passées par le système.

### [Cadran par l'extension](#)

Le cadran par la section d'extension a déjà le premier chiffre ("2" dans cet exemple) collecté. Placez le prefixDigit variable à **2**. Vous pouvez facilement changer ceci, basé sur le principal chiffre (ou s'il y a plus que celui principal chiffre possible). Aucun principal chiffre ne peut être identique qu'une autre option dans le menu (tel que 1, que vous utilisez pour le cadran-par-nom dans cet exemple). La chaîne de chiffre d'obtenir collecte également un nombre fixe de chiffres (deux dans ce cas). Par conséquent, le nombre total de chiffres doit être réparé. Si les extensions valides commencent par 2 ou 3, alors vous pouvez ajouter « placez le prefixDigit = étape de "3" » à l'option 3 de l'étape de menu suivie d'une étape de Goto DialByExtension.

Puisque vous avez les extensions de trois chiffres dans cet environnement, le paramètre de longueur d'entrée d'étape de chaîne de chiffre d'obtenir est 2. Il n'y a aucun caractère de terminaison et aucune touche d'annulation. Le nombre maximal de relances est 0. Si un utilisateur n'écrit pas une extension valide, le script incrémente le nombre de tentatives et retourne au menu

principal. L'étape ne lit aucun audio (la demande est P []) puisque l'utilisateur est déjà au milieu des chiffres de composition.

Quand deux chiffres sont avec succès collectés, ajoutez le prefixDigit au début ("2" dans ce cas) qui est déjà composé (qui te permet pour s'embrancher de l'étape de menu). Le TransferExt qui résulte alors des passages à l'extension à l'étape d'utilisateur, qui vérifie si c'est un utilisateur valide. S'il est, alors créez une demande avec définie l'extension. Dans le jeu implicite d'étape de confirmation la demande « appelle extension » a suivi par l'extension. La consultation l'utilisateur et jouent le nom parlé, comme vous avez fait dans le DialByName. Cependant, dans ce cas ce n'est pas nécessaire. Si aucun chiffre n'est composé, supposez que c'est l'extension que l'appelant veut atteindre, et réoriente l'appel. Autrement cycle de nouveau au menu principal. Le but de l'étape de la demande DP[250] de jeu est de purger n'importe quelle autre entrée, parce que l'étape de chaîne de chiffre d'obtenir n'a aucun caractère d'arrêt configuré. L'idée est que vous voulez que ceci fonctionne si quelqu'un compose l'extension "200" ou "200#". Si # est entré tandis que l'étape implicite de confirmation fonctionne, alors elle va à l'aucune partie de cette étape. Dans certains cas, il n'est probablement pas nécessaire d'avoir une étape implicite de confirmation du tout, mais de réorienter plutôt l'appel immédiatement. Pour l'illustration, ces étapes sont laissées dedans.

### [Transfert d'opérateur](#)

Lisez un « transfert d'appels » prompt le réorientent alors à l'extension configurée d'opérateur. S'il échoue pour quelque raison, vous avez un compteur distinct que vous incrémentez et retournez au menu principal. La raison pour ceci est que vous voulez avoir une certaine méthode pour donner à l'opérateur un certain temps pour obtenir outre du téléphone ou du re-registre de sorte que vous puissiez avec succès essayer de transférer vers eux de nouveau. Après qu'un certain nombre de tentatives (dur-codées à 3 dans cet exemple), vous s'embranchent à l'étiquette « désolée », qui relâche l'appel.

### [Traitement des erreurs avancé](#)

Il y a beaucoup d'endroits où les scripts peuvent échouer. Un de plus d'espaces communs dans un script bien conçu est quand les demandes d'utilisateur manquent. Quand les tentatives de système de lire une demande qui n'existe pas, un certain nombre de choses peuvent se produire. D'abord, si l'étape qui joue la demande a « pour continuer sur des erreurs prompts » réglées à oui. Il passe à l'étape suivante et ne joue pas la demande. Si continuez sur des erreurs prompts est placé à l'aucun, une exception est générée. Le plus souvent, c'est le « UndefinedPromptException », ainsi il signifie que la demande n'est pas trouvée ou n'est jamais configuré. Quand vous avez en fonction une étape de Goto d'exception, elle peut te permettre d'attraper cette exception et de s'embrancher à une section d'un code qui peut jouer un message système, ou d'une forme différente d'audio à la place. Puisqu'en fonction l'étape de Goto d'exception s'applique à tout l'audio qui joue dans le script entier (vous ne peut pas définir un distinct sur l'étape de Goto d'exception pour chaque demande vous jeu), la meilleure manière de manipuler ceci est de placer code d'erreur variable juste avant que vous jouiez la demande. De cette façon, vous pouvez s'embrancher quelque part, prendre une mesure basée sur ce code d'erreur, et probablement puis s'embrancher de retour.

Par exemple, vous pouvez modifier le script avec ceci :

Vous avez ajouté l'étiquette de « Maxretries ». Ceci est utilisé pour arriver de retour au même lieu. Placez un ErrorCode variable à **Maxretries**. Dans l'étape de demande de jeu, assurez-vous que continuez sur des erreurs prompts est placé à **non**. Ajoutez maintenant une section qui peut

manipuler ceci. Voyez si l'ErrorCode apparie « Maxretries », alors placent le SorryProblems variable (que vous avez tenté de jouer plus tôt) au message système **désolé vous ont le problème. Restez s'il vous plaît sur la ligne et certains seront avec vous sous peu.** Allez alors à l'étiquette de Maxretries de sorte qu'elle puisse jouer.

La dernière étape est de changer en fonction le message de Goto d'exception pour rechercher l'erreur d'UndefinedPromptException :

Cette étape peut être n'importe où dans le script. Cependant, il est habituellement mis à ou près du début. Il y a un certain nombre d'étapes qui peuvent lire des demandes. Les systèmes invite sont répertoriés au bas de ce document.

Cette méthode peut être utilisée au lieu d'appeler le subflow checkAltGreet.aef. Vous devez lire le fichier AltGreeting.wav. S'il y a une erreur, manipulez-la. L'avantage est que vous connaissez quand il fait ou ne joue pas.

Bien que moins communs, ce sont les autres exceptions promptes :

- UndefinedPromptException
- PromptException
- UndefinedPromptGenerator
- InvalidPromptArgumentException
- UnsupportedPromptExpression

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

Il n'est pas actuellement possible de réorienter l'appel directement dans une boîte vocale. Il doit être envoyé à un nombre qui en avant à la messagerie vocale. Ceci peut être un nombre factice dans Cisco CallManager Express (ou même dans le Cisco CallManager) qui est configuré au positionnement call forward all au numéro pilote de messagerie vocale. Employez les configurations de nombre du Cisco Unity Express number/e.164 afin d'apparier ce nombre à un abonné ou à un groupe. [Transférez un appelant directement dans une boîte aux lettres d'Unity Express](#) explique comment ceci peut être fait.

### [Variables de script](#)

Vous devez créer toutes les variables utilisées dans un script avant que vous puissiez les utiliser. Si vous utilisez un et puis l'effacement postérieur il, les **outils** choisis > **valident** dans le Script Editor afin de le trouver. Pour cet exemple de script, ce sont les variables qui sont utilisées. Notez le nom, type, la valeur (qui est la valeur de la variable au commencement) et attribut. L'attribut de paramètre signifie que la variable est exposée à l'administrateur par la page Web. Quelque chose que vous voulez que l'administrateur de Cisco Unity Express personnalise les besoins d'avoir l'attribut de paramètre.

Si vous ne voulez pas enregistrer toutes ces invites personnalisées, vous pouvez éditer chacun et décocher l'attribut de **paramètre** de sorte qu'il ne soit pas exposé par l'intermédiaire de la page Web. Afin de s'assurer toujours les travaux de système, ajoutez plusieurs étapes de positionnement au début du script. Par exemple, placez « appeler » au « fournisseur de services [AA/AACalling] ». Non toutes les demandes affichées ici ont les scripts système correspondants. Dans la plupart des cas, il semble raisonnable de les enregistrer vous-même. Si les demandes ne changent pas, vous pouvez les enregistrer une fois, vous assurez qu'elles existent sur le système

de Cisco Unity Express avec un nom du fichier donné, et puis retirent l'attribut de paramètre de la variable. De cette façon, la demande lit un nom du fichier fixe. Mais, la variable n'est pas exposée par l'intermédiaire de l'interface web où un administrateur peut accidentellement la changer.

## Téléchargez un script au Cisco Unity Express

Habituellement, la première étape pour ajouter une nouvelle application et un script au Cisco Unity Express est de télécharger les demandes. C'est important parce que les demandes doivent exister sur le système afin que vous puissiez pour les configurer dans les paramètres de script. Autrement, vous devez laisser le blanc de paramètres, télécharger les demandes, et puis retourner et éditer les paramètres de script. Pour des informations supplémentaires sur la façon enregistrer et télécharger des demandes, référez-vous à la [gestion par l'intermédiaire du système de gestion du téléphone \(AVT\) /Greetings](#).

Une fois que vous faites enregistrer votre script et êtes prêt à l'essayer, vous le premier besoin de le télécharger au Cisco Unity Express. Le moyen le plus simple est d'utiliser le GUI, puisque cela n'exige pas d'un ftp server de charger les fichiers script de.

Procédez comme suit :

1. Ouvrez une session au GUI de Cisco Unity Express avec un compte qui a des droites d'administration.
2. **Messagerie vocale > réception automatique** choisies.
3. Cliquez sur **Add**.
4. À côté « du script propre automatisé sélectionné », cliquez sur le bouton de **téléchargement**.
5. Cliquez sur **Browse**.
6. Trouvez le fichier script, sélectionnez-le et l'appuyez sur **CORRECT**.
7. Si vous voulez changer le nom du fichier de destination, faites ainsi. Autrement **téléchargement de presse**.
8. Écrivez un nom d'application. Ceci peut être quelque chose en minuscules et est utilisé pour se rapporter à l'application associée avec ce fichier script.
9. Cliquez sur **Next** (Suivant).
10. La page affiche maintenant toutes les variables pour le script qui sont marquées comme paramètres. Ceci signifie qu'ils peuvent être configurés à cette page. Si vous n'avez pas téléchargé les fichiers prompts encore, alors vous pouvez passer par ce premier et puis compléter les informations plus tard.
11. Cliquez sur Next quand vous finissez d'écrire des paramètres de script.
12. Assurez-vous que l'application est marqué en tant qu'**activé**, configurez le nombre maximal d'appels simultanés (le par défaut est le nombre maximal laissé par le permis) et introduisez un nombre d'émission à ligne ouverte. C'est le nombre qui s'appelle pour lequel cette application particulière doit être appelée (déclencheur). Ceci peut être votre nombre principal de réception automatique, ou provisoire. Vous pouvez également ignorer ceci et le remplir dedans plus tard. Sans nombre d'émission à ligne ouverte, vous ne pouvez pas tester ou lancer un script. **Remarque:** Pour des intégrations de Cisco CallManager avec le support SRST, vous devez introduire de plusieurs nombres d'émission à ligne ouverte. Avant le Cisco Unity Express 2.1.1, ces informations ont dû être écrites par le CLI. Dans le Cisco Unity Express 2.1, de plusieurs nombres d'émission à ligne ouverte (déclencheurs) peuvent être introduits de l'écran de **nombres de gestion > d'émission à ligne ouverte**.
13. Clic **terminé**.

Une fois que vous téléchargez le script et configurez l'application, vous devez vous assurer que vous pouvez conduire des appels à l'application. Ceci signifie que Cisco CallManager Express ou le Cisco CallManager doit être configuré. Avec le CallManager Express, ceci implique la configuration d'un cadran-pair qui indique le Cisco Unity Express qui apparie le modèle (nombre de déclencheur ou le nombre d'émission à ligne ouverte) configuré dans le Cisco Unity Express. Dans le Cisco CallManager, un point de routage CTI avec ce nombre doit être ajouté. N'oubliez pas que vous devez associer ce point de routage CTI avec l'utilisateur JTAPI des logins de ce Cisco Unity Express As. Cette configuration peut probablement exiger une réinitialisation de Cisco Unity Express. Une fois que vous ajoutez le point d'acheminement et l'associez avec l'utilisateur, vous pouvez changer le script autant que vous voulez sans nécessité de redémarrer.

Quand vous mettez à jour un script, si aucun paramètre variable n'est ajouté ou est changé, vous pouvez entrer dans l'éditeur de Cisco Unity Express, cliquez sur la **messagerie vocale > les scripts**, téléchargez, et sélectionnez le script. Il est remplacé avec tous autres paramètres configurés qui demeurent les mêmes.

Quand vous créez des demandes, la chose la plus facile à faire est d'utiliser le système AVT. Le fichier le nomme enregistre font inclure des horodatages des messages de date/heure. Mais dans l'unité de Cisco exprimez 2.1.1 et plus tard, il est possible d'utiliser le **ccn CLI renomment la commande prompt**. Précédemment, vous avez dû télécharger la demande, re-téléchargement il avec un nouveau nom, puis supprimez le fichier de l'original .wav. 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>
```

## Dépannez

Quand vous dépannez un script, vous devez appeler dedans et tester de divers scénarios, les deux entrée régulière d'utilisateur, aussi bien que délai d'attente et conditions d'erreurs anticipées.

Avant que vous téléchargez un script, accédez à toujours les **outils > valident** l'entrée dans le Script Editor et s'assurent que vous recevez le message réussi « par validation ». Sinon, tout l'affichage d'erreurs dans le volet de débogage. Vous pouvez les double-cliquer et il t'affiche où la condition d'erreurs existe dans le script.

Dans la plupart des exemples, les suivis par défaut sont plus que suffisants pour dépannage des problèmes. Afin de simplifier ceci, laissez les suivis de par défaut en fonction et utiliser la capacité de filtrage afin d'afficher de ce que vous avez besoin.

Avant n'importe quel test, émettez la **commande trace claire**. Ceci efface le tampon mémoire des messages de suivi de sorte que la quantité d'informations filtrée inclue seulement les données après que la commande soit émise.

Pour l'exécution de script, l'élément le plus important à filtrer en fonction est EXECUTING\_STEP. Ceci affiche toutes les étapes pendant qu'ils sont exécutés. Exemple :

```
cue-3745-44a> clear trace cue-3745-44a> 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 de sortie certaines des informations redondantes sont enlevées au milieu et ont cultivé le début de l'horodatage des messages de date/heure. Le plus souvent, la plupart d'informations importantes sont à l'extrémité de chaque ligne.

**Remarque:** Plusieurs de ces lignes de sortie sont rapportées à une deuxième et troisième ligne due aux raisons spatiales.

```
5:33.956 Step id=529,Step Class=com.cisco.wfframework.steps.core.StepComment, Step Description=
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=
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)
```

Maintenant vous pouvez voir chaque étape pendant qu'elle exécute. Vous ne pouvez pas voir comment les expressions évaluent, ni pouvez vous voir n'importe quel utilisateur entrer. Si le script devant vous, vous peut suivre où le script s'embrancher, avec la connaissance des chiffres vous êtes entré (dans ce cas, l'utilisateur se connecte, compose une extension, 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
```



De cette sortie vous voyez que l'extension 201 est composée.

Si les versions du CUE et du scripteur de CUE sont les deux différentes, et vous essayez de télécharger un script aa par l'intermédiaire du GUI ou du CLI, le téléchargement échoue avec le message d'erreur. Si les deux versions sont différentes, alors vous devez recréer le script utilisant l'éditeur de CUE de la version qui est identiques que votre CUE. Ou, améliorez le CUE à la version du scripteur de CUE et exécutez le script.

## Annexe

### Annexe 1 - Systèmes invite

Cette table prouve à des systèmes invite que vous pouvez utiliser.

**Remarque:** Vous ne pouvez pas réutiliser toutes les demandes que vous entendez dans l'application de messagerie vocale de Cisco Unity Express dans un script personnalisé. Certaines des demandes sont seulement disponibles par l'intermédiaire du script de messagerie vocale que vous ne pouvez pas personnaliser par l'intermédiaire du Script Editor de Cisco Unity Express.

Système invite	Audio enregistré
fournisseur de services [AA/AAWelcome]	« Accueil au préposé automatisé »
fournisseur de services [AA/AAMainMenu]	« Pour introduire le numéro de téléphone de la personne que vous essayez d'atteindre la presse 1, pour écrire le nom de la personne vous essayez d'atteindre la presse 2, pour virer sur la presse d'opérateur 0" »
fournisseur de services [AA/AAEnterExtn]	« Introduisez s'il vous plaît le numéro de téléphone et appuyez sur # clé »
fournisseur de services [AA/AACallingExtn]	« Appelant l'extension »
fournisseur de services [AA/AAPhoneReach]	« Le numéro de téléphone que vous essayez d'atteindre »
fournisseur de services [AA/AAOutOfServicePhone]	« Est actuellement hors service »
fournisseur de services [AA/AANameDial]	« Orthographiez le nom de famille de la personne que vous voulez appeler suivi du prénom. Pour la lettre Q, presse 7, pour la presse Z 9" »

fournisseur de services [AA/AACalling]	« Appellant »
fournisseur de services [AA/AASorry]	« Désolé vous avez le problème. Restez s'il vous plaît sur la ligne et certains seront avec vous sous peu. »
SP[AA/AAWant 2Call]	« Si c'est le nom de la personne que vous appelez, la presse 1, pour recommencer, appuyer sur * »
fournisseur de services [AA/still_there]	« Êtes toujours vous là ? »

## [Annexe 2 - Remettez à l'état initial les suivis par défaut](#)

Afin de remettre à l'état initial le système (Cisco Unity Express 2.1.x et plus tôt) pour transférer des suivis par le CLI, d'abord désactivez tous les suivis utilisant l'**aucun suivi toute** la commande, puis collez ceci dans le CLI :

```

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 debug
trace ccn managercontact debug
trace ccn stepcall debug
trace ccn stepmedia debug
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 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 debug  
trace ccn managercontact debug  
trace ccn stepcall debug  
trace ccn stepmedia debug  
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](#)