

# Restrictions de référentiel UCCX

## Contenu

[Introduction](#)

[Informations générales](#)

[Problème](#)

[Solution](#)

[Taille de données en cours de référentiel d'évaluation](#)

[Déterminez le nombre de lignes dans des Tableaux de répertoire de référentiel](#)

[Déterminez le nombre de pages utilisées par des Tableaux de répertoire de référentiel](#)

[Déterminez le nombre de pages utilisées par des Tableaux de fichier d'élément de référentiel](#)

[Calculez](#)

[Demandes](#)

## Introduction

Ce document décrit une question produite quand vous améliorez des systèmes du Cisco Unified Contact Center Express (UCCX) aux versions 8 et ultérieures et un grand nombre d'éléments de référentiel sont téléchargés au système, ou quand vous tentez de télécharger un grand nombre d'éléments de référentiel au système dans les versions 8 et ultérieures.

## [Informations générales](#)

Les versions 7.x et ultérieures UCCX utilisent Microsoft SQL (MSSQL) comme engine de base de données. MSSQL ne distingue pas, en termes de stockage de données, entre les données de différents types. Quand il enregistre des données dans une base de données 3-GB, MSSQL enregistre toutes les données, indépendamment du type, dans un bloc 3-GB.

En revanche, Informix, l'engine de base de données utilisée dans des versions 8.0 et ultérieures UCCX, différencie entre les données de différents types quand il l'enregistre sur le disque. Des données typiques de base de données (telles que des chaînes, des caractères, et des entiers.) sont enregistrées dans un bloc de disque consacré à la base de données, tandis que de grandes données binaires de l'objet (BLOB), si en existe dans les enregistrements de table de base de données, sont enregistrées dans une partie indépendante du disque, appelée un sbspace. Un sbspace est une unité logique composée d'un ou plusieurs blocs de disque qui enregistrent des données BLOB. Informix enregistre des données traditionnelles et des données BLOB séparément afin d'augmenter la représentation des données BLOB de lecture et d'écriture à partir et derrière la base de données et derrière le disque. Quand on crée une base de données qui contient des données BLOB, l'administrateur doit spécifier la taille des blocs de disque pour la base de données (afin d'enregistrer des données traditionnelles) et la taille du sbspace séparément.

Pour des mécanismes de stockage de données, MSSQL place toutes les données dans une position simple de la taille N, alors qu'Informix divise la mémoire de ces données en deux positions : une position pour les informations contextuelles sur des données BLOB de la taille X, et une position différente pour le BLOB s'objecte de la taille Y.

Dans UCCX, l'administrateur a l'option de télécharger les éléments de référentiel qui se composent des demandes, des documents, des grammaires, et des scripts. Le contenu de ces éléments est enregistré dans les tables correspondantes de base de données comme données BLOB et informations contextuelles sur elles, telles que le nom du fichier, le répertoire, le temps de Last modified, l'utilisateur de Last modified, la longueur, et la somme de contrôle.

Des éléments de référentiel sont enregistrés dans la base de données UCCX **db\_cra\_repository**. Dans les versions 7.x et antérieures UCCX qui utilisent MSSQL, le **db\_cra\_repository** est 3 Go dans la taille et contient les informations contextuelles et BLOB. Dans les versions 8.0 et ultérieures UCCX qui utilisent Informix, le bloc de mémoire de données relié au **db\_cra\_repository** est 10.2 Mo dans la taille, et stocke seulement les informations contextuelles sur les éléments de référentiel. Le contenu des éléments de référentiel est enregistré dans le format BLOB dans un sbspace appelé l'**uccx\_sbspace**. Dans des versions 8.0 et ultérieures UCCX, l'**uccx\_sbspace** est 3 Go dans la taille.

La sortie du **disque de dbserver d'uccx d'exposition** sur un serveur de la version 8.0+ UCCX, indique la distinction entre ces deux datastores :

Selon le mélange de données dans la base de données MSSQL, il est possible à la taille des données BLOB enregistrées dans la base de données MSSQL pour dépasser la taille définie du sbspace dans Informix quand le transfert ou la mise à jour est tenté. De même, il est possible que les informations contextuelles sur les données BLOB enregistrées dans la base de données MSSQL dépassent la taille administratif-spécifiée pour cela des données dans le bloc de base de données Informix.

Quand ceci se produit, la mise à jour ou le transfert de la version 7.x UCCX à la version 8.x UCCX échoue, parce que le **db\_cra\_repository** ou l'**uccx\_sbspace** n'est pas assez grand pour faciliter les mêmes informations qui ont été stockées dans MSSQL. C'est typiquement une question dans un système UCCX qui contient un grand nombre de demandes. La demande contextuelle et les données BLOB doivent partager le **db\_cra\_repository** et l'**uccx\_sbspace** avec des documents, des grammaires, et des scripts, mais ces autres types de référentiel sont en général petits dans la taille et numérotent.

Comme exemple, considérez un système de version 7.x UCCX avec des dizaines de milliers de demandes, chacun avec seulement quelques secondes d'audio. Dans la version 7.x UCCX qui l'utilise MSSQL, le contenu et les informations contextuelles rapides est enregistré dans le même bloc 3-GB. Car il y a beaucoup de demandes de petite taille, la base de données pourrait enregistrer 50 Mo des informations contextuelles sur les demandes, mais seulement 2 Go des données BLOB qui représentent l'audio des demandes. Par conséquent, les demandes dans le référentiel occupent un peu plus de 2 Go du positionnement de limite 3-GB sur la création de base de données.

Quand vous tentez de migrer ce système vers la version 8.x et l'Informix UCCX, le transfert échoue parce que les 50 Mo des informations contextuelles dépasse la limite du Mo 10.2 du **db\_cra\_repostoriory** quoique les 2 Go des adaptations satisfaites rapides bien sous la limite de l'**uccx\_sbspace**.

Inversement, considérez un système de version 7.x UCCX avec moins, mais encore beaucoup, de longues demandes. Avec moins de demandes mais de plus grande taille, le rapport du contenu prompt aux informations contextuelles est différent. Dans la version 7.x UCCX et le MSSQL, le contenu rapide pourrait prendre 2.8 Go du Mo de l'information **db\_cra\_repository** et de contextuelle 3. Ce système améliore avec succès, comme 3 adaptations de Mo dans les adaptations **db\_cra\_repository** et de 2.8 de Go dans l'**uccx\_sbspace** alloué.

Typiquement, quand vous tentez de migrer vers des versions 8.x et ultérieures UCCX, les données contextuelles au sujet des demandes téléchargées au système de versions 7.x ou antérieures UCCX dépassent la limite de taille de **db\_cra\_repository** avant que le contenu rapide dépasse la limite de taille de l'**uccx\_sbspace**. Supplémentaire, l'espace libre vrai disponible pour les éléments personnalisés de référentiel est 6.9 Mo, car la configuration par défaut consomme 3.4 Mo du **db\_cra\_repository**.

## Problème

Quand vous tentez de télécharger de nouveaux éléments de référentiel (documents, grammaires, demandes, scripts) à un système UCCX qui exécute des versions 8 ou ultérieures, vous recevez ce message d'erreur :

```
The files uploaded are not valid or not structured
according to languages. Please check the help
documentation for more details.
```

Le transfert des versions 7.0(2) et antérieures UCCX aux versions 8.0 et ultérieures change l'engine de système d'exploitation et de base de données sur laquelle l'application fonctionne. L'engine de base de données utilisée dans des versions 8.0 et ultérieures UCCX enregistre des données différemment que celle des versions 7.x et ultérieures UCCX. Ceci a des implications pour le transfert d'UCCX, comme bases de données qui contiennent de grands ensembles de données dans la version 7.x UCCX ne pourraient pas migrer correctement à la version 8.x UCCX.

## Solution

Avant que vous migriez vers la version 8.x UCCX, estimez la quantité de **db\_cra\_repository** et d'**uccx\_sbspace** requis afin d'enregistrer les éléments en cours de référentiel dans le système de version 7.x UCCX, pour inclure n'importe quelle croissance future.

### Taille de données en cours de référentiel d'évaluation

Afin de commencer, déterminez le nombre de lignes dans chacune des tables de référentiel qui tiennent les informations sur des éléments et des répertoires de référentiel.

### Déterminez le nombre de lignes dans des Tableaux de répertoire de référentiel

Utilisez le SQL Query Analyzer de Microsoft afin d'enregistrer le nombre de lignes des tables de répertoire de référentiel avec ces commandes :

- **SÉLECTIONNEZ LE COMPTE (\*) du documentsfoldertbl**

- SÉLECTIONNEZ LE COMPTE (\*) du grammarsfoldertbl
- SÉLECTIONNEZ LE COMPTE (\*) du promptsfoldertbl
- SÉLECTIONNEZ LE COMPTE (\*) du scriptsfoldertbl

### Déterminez le nombre de pages utilisées par des Tableaux de répertoire de référentiel

Informix explique la taille-sur-disque en termes de pages. Déterminez le nombre de pages qui est occupé par le contenu des tables de répertoire de référentiel avec cette formule, et substituez le **nombre de lignes** aux comptes obtenus à partir des commandes précédent-mentionnées. Calculez cette formule pour chaque table, et ajoutez le nombre de pages. Il n'est pas possible de déterminer exactement le nombre de pages si le nombre de lignes de chaque table est ajouté d'abord, et alors le résultat de formule est calculé.

- # page le documentsfoldertbl = le **nombre de lignes dans le documentsfoldertbl**/(2020/(180 + 4))
- # page le grammarsfoldertbl = le **nombre de lignes dans le grammarsfoldertbl**/(2020/(180 + 4))
- # page le promptsfoldertbl = le **nombre de lignes dans le promptsfoldertbl**/(2020/(180 + 4))
- # page le scriptsfoldertbl = le **nombre de lignes dans le scriptsfoldertbl**/(2020/(180 + 4))

# page le documentsfoldertbl + # grammarsfoldertbl de pages + # promptsfoldertbl de pages + # scriptsfoldertbl de pages = **nombre total de pages pour des Tableaux de répertoire**

### Déterminez le nombre de pages utilisées par des Tableaux de fichier d'élément de référentiel

Terminez-vous le même calcul afin de déterminer le nombre total de pages pour les tables de fichier qui contiennent les éléments réels de référentiel. Sélectionnez ces commandes avec le SQL Query Analyzer de Microsoft :

- SÉLECTIONNEZ LE COMPTE (\*) du documentsfiletbl
- SÉLECTIONNEZ LE COMPTE (\*) du grammarsfiletbl
- SÉLECTIONNEZ LE COMPTE (\*) du promptsfiletbl
- SÉLECTIONNEZ LE COMPTE (\*) du scriptsfiletbl

Déterminez le nombre de pages qui sont occupées par le contenu des tables de fichier de référentiel avec cette formule, et substituez le **nombre de lignes** avec les comptes obtenus à partir des commandes précédent-mentionnées. Calculez la formule pour chaque table, et ajoutez le nombre de pages.

- # page le documentsfiletbl = le **nombre de lignes dans le documentsfiletbl**/(2020/(229 + 4))
- # page le grammarsfiletbl = le **nombre de lignes dans le grammarsfiletbl**/(2020/(229 + 4))
- # page le promptsfiletbl = le **nombre de lignes dans le promptsfiletbl**/(2020/(229 + 4))
- # page le scriptsfiletbl = le **nombre de lignes dans le scriptsfiletbl**/(2020/(229 + 4))

# page le documentsfiletbl + # grammarsfiletbl de pages + # promptsfiletbl de pages + # scriptsfiletbl de pages = **nombre total de pages pour des Tableaux de fichier**

### Calculez

Exécutez ces calculs afin de terminer l'évaluation en cours de taille de données de référentiel :

1. Déterminez le nombre total de pages nécessaires afin d'enregistrer le référentiel en cours dans Informix avec :

**Nombre total de pages = de nombre total de pages pour les Tableaux de fichier + le nombre total de pages pour des Tableaux de répertoire**

2. Déterminez tout le espace, dans le Mo, que les pages consomment :

**Nombre total de pages X 2 = taille totale dans le Mo**

Si les calculs prouvent que les informations contextuelles sur les éléments et les dossiers de référentiel actuellement téléchargés dans la version 7.x UCCX dépassent 3.4 Mo, alors elles sont recommandées au refactor la conception d'élément de référentiel. Bien que l'espace libre disponible pour les informations contextuelles sur des éléments de référentiel dans **db\_cra\_repository** soit 6.9 Mo, il est recommandé de laisser 50% disponible pour la croissance future. Les évaluations de croissance et l'espace occupé maximal permis est calculées par déploiement, basé sur des facteurs de croissance prévus.

## Demandes

Puisque les demandes sont typiquement le plus grand consommateur de l'espace de référentiel, des méthodes utilisées afin de réduire le nombre de demandes dans le référentiel sont discutées dans le reste de cet article.

Si les demandes actuellement téléchargées dans le référentiel de version 7.x UCCX occupent une partie significative de l'espace de stockage, du refactor la conception rapide, de la mémoire, et de la récupération globaux de référentiel avant que vous migriez vers la version 8.x UCCX. Quand vous tentez au refactor la conception prompte, considérez ces options :

- Réduisez le nombre de répertoires prompts. Suivant les indications des calculs précédents, chaque répertoire prompt occupe une ligne dans la base de données de référentiel, et consomme donc l'espace contre la limite **db\_cra\_repository**. Si vous réduisez le nombre de répertoires, vous pouvez l'espace libre pour des fichiers.
- Le partage incite pour des mots ou expression communs à travers des applications/scripts afin de réduire le nombre de demandes requises dans le système.
- Utilisez système-a fourni des demandes pour des vulgarisations parmi des demandes, telles que des nombres et des devises.
- Enregistrez les demandes sur un web server distinct au sein de l'entreprise, et récupérez et lisez les demandes nécessaires sur demande avec la Voix XML (VXML).

VXML est utilisé afin de récupérer et lire des demandes sur demande d'un emplacement de hors fonction-case. Si vous enregistrez un grand nombre de demandes sur un web server distinct, vous pouvez :

- Éliminez les problèmes de transfert précédemment mentionnés, comme des demandes ne sont plus enregistrées dans le **db\_cra\_repository**.
- Fournissez un plus grands accès et facilité de Gestion.
- Réduisez la commutateur-version, la mise à jour, le transfert, et le temps de sauvegarde.

Bien que beaucoup d'options existent afin d'équiper la personnalisation de la réponse vocale interactive (RVI) dans VXML, le script UCCX et l'application VXML utilisée afin de récupérer une demande d'un web server de hors fonction-case et la lire à l'appelant est utilisé comme base du

développement ultérieur. Tout comme l'autre script fait sur commande dans UCCX, les scripts fournis dans cette section sont offerts comme un guide et ne sont pas pris en charge par le centre d'assistance technique Cisco (TAC).

Remarque: Une application VXML est appelée d'un script UCCX avec l'étape de **navigateur de Voix**. Plus d'informations sur l'étape de **navigateur de Voix** sont fournies des [guides de programmation de Cisco Unified Contact Center Express](#).

L'étape de **navigateur de Voix** de The Voice consomme un document VXML. Ce document doit être créé suite à l'étape de **document URL de création**, et doit être hébergé sur un web server externe à UCCX. Bien que l'application VXML soit écrite afin de recevoir l'appelant entré par la fréquence multi de double tonalité (DTMF), cette application est conçue pour jouer seulement une demande qui est hors fonction-case hébergée. Cependant, il peut être développé afin d'inclure la fonctionnalité supplémentaire. On le suppose que le reste du script UCCX, avant que l'étape de **navigateur de Voix** soit appelée, a la logique requise afin de déterminer quelle demande est jouée, et un positionnement de variable de chaîne au nom du fichier prompt.

Puisque le document VXML est statique, mais la demande lue par elle est dynamique, un langage de script de côté serveur est utilisé afin de créer le document VXML. Ceci peut être n'importe quel langage de script de côté serveur qui a la capacité de placer l'en-tête de type de contenu de la réponse de **demande GET XML**. Dans cet exemple, le PHP est utilisé.

La page PHP est écrite afin de recevoir un paramètre URL dans une **demande GET** qui représente le nom de demande d'audio qui est lu. La page PHP concatène le modèle VXML avec le nom du fichier prompt passé dans les paramètres URL de **demande GET** afin de former le document complet VXML. Il place alors l'en-tête de type de contenu de la réponse au XML, et place le corps de la réponse pour être le contenu VXML.

```
<?php
$wav_filename = $_GET['wav'];

$xml_string = '<?xml version="1.0"?>
<vxml xmlns="http://www.w3.org/2001/vxml" version="2.0">
  <form>
    <block>
      <prompt bargein="true">
        <audio src="http://<Servername or IP Address>/
          <Path>/'. $wav_filename. '.wav" />
      </prompt>
    </block>
  </form>
</vxml>';
header('Content-type: text/xml');
echo $xml_string;
?>
```

Afin de produire un document bien formé VXML, la page PHP d'exemple doit être accédée à avec une **demande GET** qui contient un **wav de** paramètre et une valeur de chaîne, avec la supposition que la page PHP d'exemple est nommée **generatevxml.php** :

**http://<Servername or IP Address>/path/generatevxml.php?wav=MenuPrompt**

Assurez-vous que le **MenuPrompt.wav** est dans l'emplacement sur le web server externe spécifié dans le modèle VXML contenu dans la page PHP.

Dans le script UCCX, employez l'étape de **document URL de création** afin d'effectuer une **demande GET de generatevxml.php** concaténant l'URL de base du **<Servername ou de l'adresse**

**IP >/path/generatevxml.php de http:// ? le wav= avec le nom du fichier prompt a dérivé de la logique précédente de script, et place le résultat dans une variable de document.**

Créez une étape de **navigateur de Voix** qui consomme la variable de document.

Quand ce script s'appelle, fourni le **generatevxml.php** et MenuPrompt.wavare **accessible** sur le web server d'UCCX, le MenuPrompt.wavPrompt **le lit à l'appelant**.

Quand des applications VXML sont utilisées afin d'enregistrer la hors fonction-case de demandes, de sorte qu'elles soient accédées à seulement quand nécessaire dans la commande les lisent à l'appelant, elle tient compte d'une plus grandes efficacité, gestionnabilité, et utilité. C'est une question pour la considération si un système de version 7.x UCCX est amélioré à un système de version 8.x UCCX, et le nombre de demandes sont tel que le contenu des informations contextuelles est plus grand que **db\_cra\_repository** ou **uccx\_sbpace**.