

# Beperkingen van UCCX-indeling

## Inhoud

[Inleiding](#)

[Achtergrondinformatie](#)

[Probleem](#)

[Oplossing](#)

[Schatting van de grootte van de actuele registratie](#)

[Bepaal het aantal rijen in de tabellen van de reservemap](#)

[Bepaal het aantal pagina's dat wordt gebruikt in de tabellen voor reservemappen](#)

[Bepaal het aantal pagina's dat wordt gebruikt in de tabellen voor bestandsindelingen van de optie berekenen](#)

[Vragen](#)

## Inleiding

Dit document beschrijft een probleem dat u hebt ondervonden wanneer u de Cisco Unified Contact Center Express (UCCX)-systemen naar versies 8 en later upgrades uitvoert en een groot aantal reserveonderdelen naar het systeem wordt geüpload of wanneer u in versies 8 en later een groot aantal reserveonderdelen naar het systeem probeert te uploaden.

## Achtergrondinformatie

UCCX versies 7.x en gebruikt later Microsoft SQL (MSSQL) als de databases. MSSQL maakt in termen van gegevensopslag geen onderscheid tussen gegevens van verschillende typen.

Wanneer het gegevens in een 3 GB gegevensbestand opslaat, slaat MSSQL alle gegevens, ongeacht het type, op in één 3 GB blok.

Daarentegen maakt Informix, de databases die bij UCCX versies 8.0 en hoger worden gebruikt, een onderscheid tussen gegevens van verschillende typen wanneer deze op de schijf worden opgeslagen. De typische gegevens van de gegevensbank (zoals Kiekjes, Tekens, en Integers.) worden opgeslagen in een schijfbrok gewijd aan de gegevensbank, terwijl de gegevens van de BLOB (Binary Large Object), als om het even welke in de gegevens van de gegevensbank lijst bestaat, in een afzonderlijk gedeelte van de schijf opgeslagen wordt, die een sbspace wordt genoemd. Een sbspace is een logische eenheid die bestaat uit een of meer schijfeenheden die BLOB gegevens opslaan. Informix slaat traditionele gegevens en BLOB gegevens afzonderlijk op om de prestaties van het lezen en schrijven van BLOB gegevens van en naar de database en schijf te verbeteren. Wanneer een database wordt gecreëerd die BLOB gegevens bevat, moet de beheerder de grootte van de diskgroepen voor de database (om traditionele gegevens op te slaan) en de grootte van de ruimte afzonderlijk specificeren.

Voor gegevensopslagmechanismen plaatst MSSQL alle gegevens in één emmer van grootte N,

terwijl Informix de opslag van deze gegevens in twee emmers verdeelt: één emmer voor de contextuele informatie over BLOB-gegevens van grootte X, en een ander emmer voor de BLOB-objecten zelf van grootte Y.

In UCCX heeft de beheerder de optie om positieartikelen te uploaden die uit claims, documenten, grammatica's en scripts bestaan. De inhoud van deze items wordt opgeslagen in de corresponderende database-tabellen als BLOB-gegevens en contextuele informatie over deze items, zoals bestandsnaam, map, laatst aangepaste tijd, laatste aangepaste gebruiker, lengte en checksum.

Bewaarposten worden opgeslagen in de UCCX database **db\_cra\_cache**. In UCCX versies 7.x en eerder die MSSQL gebruiken, is **db\_cra\_cache** 3 GB in grootte en bevat zowel de contextuele als de BLOB-informatie. In UCCX versies 8.0 en hoger die Informix gebruiken, is de opslagkit die aan de **db\_cra\_opslagplaats** is gekoppeld, 10,2 MB groot en slaat alleen de contextuele informatie over de positieartikelen op. De inhoud van de positieobjecten wordt in BLOB formaat opgeslagen in een ruimte genaamd **uccx\_sbospace**. In UCCX versies 8.0 en hoger is de **uccx\_sbospace** 3 GB groot.

De uitvoer van de **CD-serverschijf** van **show** op een UCCX versie 8.0+ server onthult het onderscheid tussen deze twee datastores:

```
admin:show uccx dbserver disk
```

SNO.	DATABASE NAME	TOTAL SIZE (MB)	USED SIZE (MB)	FREE SIZE (MB)	PERCENT FREE
1	rootdbs	358.4	59.5	298.9	83%
2	log_dbs	317.4	307.3	10.1	3%
3	db_cra	512.0	17.3	494.7	96%
4	db_hist	24508.6	6661.2	17847.2	80%
5	db_cra_repository	10.2	3.4	6.9	67%
6	db_trascat	512.0	3.3	508.7	99%
7	temp_uccx	1572.9	0.1	1572.7	99%
8	uccx_sbospace	3145.7	2988.1	157.6	5%
9	uccx_er	204.0	0.1	204.7	99%
10	uccx_ersb	1572.9	1494.1	78.8	5%
11	sadmin	102.4	4.3	98.1	95%

Afhankelijk van de mix van gegevens in de MSSQL database is het mogelijk dat de grootte van de BLOB gegevens die opgeslagen zijn in de MSSQL database de gedefinieerde grootte van de sbospace in Informix overschrijdt wanneer migratie of upgrade wordt geprobeerd. Op dezelfde manier is het mogelijk dat de contextuele informatie over de BLOB gegevens die opgeslagen zijn in de MSSQL database de administratief gespecificeerde grootte voor die gegevens in de Informix database chunk overschrijdt.

Wanneer dit gebeurt, zal upgrade of migratie van UCCX versie 7.x naar UCCX Versie 8.x mislukt, omdat ofwel de **db\_cra\_cache** of de **uccx\_sbospace** niet groot genoeg is om dezelfde informatie te verwerken die in MSSQL is opgeslagen. Dit is meestal een probleem in een UCCX-systeem dat een groot aantal oproepen bevat. Contextual Prompt and BLOB data moeten de **db\_cra\_cache** en de **uccx\_sbospace** delen met documenten, grammars en scripts, maar deze andere Repository types zijn doorgaans klein in omvang en aantal.

Neem bijvoorbeeld een UCCX versie 7.x-systeem met tienduizenden oproepen in acht, elk met slechts een paar seconden audio. In UCCX versie 7.x die MSSQL gebruikt, worden de snelle inhoud en contextuele informatie opgeslagen in dezelfde 3-GB unk. Omdat er veel kleine propjes

zijn, kan de database 50 MB contextuele informatie over de propts opslaan, maar slechts 2 GB BLOB data die de audio van de propts representeert. Daarom hebben de Prompts in de Repository ruim 2 GB van de 3 GB-limiet die bij het creëren van een database is ingesteld.

Wanneer u probeert om dit systeem te migreren naar UCCX versie 8.x en Informix, mislukt de migratie omdat de 50 MB aan contextuele informatie de 10,2 MB limiet van **db\_cra\_repostory** overschrijdt ook al past de 2 GB aan Prompt content goed onder de limiet van de **uccx\_sbospace**.

Omgekeerd, denk aan een UCCX versie 7.x systeem met minder, maar nog steeds veel, lange Prompts. Met minder snelle maar grotere hoeveelheden is de verhouding tussen snelle inhoud en contextuele informatie anders. In UCCX versie 7.x en MSSQL kan de Prompt-inhoud 2,8 GB van de **db\_cra\_opslagplaats** innemen en de contextuele informatie 3 MB. Dit systeem werkt met succes, aangezien 3 MB past in de **db\_cra\_cache** en de 2,8 GB past in de **uccx\_sbospace** toegewezen.

Meestal, wanneer u probeert te migreren naar UCCX versies 8.x en later, overschrijden de contextuele gegevens over de Prompts die naar de UCCX versies 7.x of eerder systeem zijn geüpload de groothandelsgrens van **db\_cra\_storage** voordat de vroege inhoud de groottegrens van de **uccx\_sbospace** overschrijdt. Bovendien is de echte vrije ruimte die beschikbaar is voor op maat gesneden bewaarinstantellingen 6,9 MB, omdat de standaardconfiguratie 3,4 MB van de **db\_cra\_cache** verwerkt.

## Probleem

Wanneer u nieuwe positieartikelen (documenten, grammofoons, benaderingen, scripts) probeert te uploaden naar een UCCX-systeem met versies 8 of hoger, ontvangt u deze foutmelding:

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

De migratie van UCCX versies 7.0(2) en eerder naar versies 8.0 en wijzigt later het besturingssysteem en de databases waarop de toepassing draait. De databases die bij UCCX versies 8.0 worden gebruikt, slaan later gegevens op een andere manier op dan die bij UCCX versies 7.x en hoger. Dit heeft gevolgen voor de migratie van UCCX, omdat databases die grote datasets in UCCX versie 7.x bevatten, mogelijk niet correct naar UCCX versie 8.x migreren.

## Oplossing

Voordat u naar UCCX versie 8.x migreert, moet u de hoeveelheid **db\_cra\_cache** en **uccx\_sbospace** schatten die nodig is om de huidige reserveonderdelen in het UCCX versie 7.x-systeem op te slaan, om eventuele toekomstige groei op te nemen.

## Schatting van de grootte van de actuele registratie

Bepaal om te beginnen het aantal rijen in elk van de tabellen van de Repository die informatie bevatten over zowel de items van de Repositor als de mappen.

## Bepaal het aantal rijen in de tabellen van de reservemap

Gebruik Microsoft SQL Query Analyzer om het aantal rijen van de tabellen van de Bewaarmap met deze opdrachten op te nemen:

- **SELECTEER COUNT(\*) UIT documentenmap**
- **SELECTEER COUNT(\*) UIT grammaticale parameters**
- **SELECTEER TELLER(\*) UIT de aanwijzingen**
- **SELECTEER COUNT(\*) UIT scriptsfoldertbl**

## Bepaal het aantal pagina's dat wordt gebruikt in de tabellen voor reservemappen

Informix maakt een boekhouding van de grootte-op-schijf in termen van pagina's. Bepaal het aantal pagina's dat in gebruik is van de inhoud van de tabellen voor de map Repository met deze formule en vervang het **aantal rijen** voor de tellingen die zijn verkregen met de eerder genoemde opdrachten. Bereken deze formule voor elke tabel en voeg het aantal pagina's toe. Het is niet mogelijk het aantal pagina's nauwkeurig te bepalen als het aantal rijen van elke tabel eerst wordt toegevoegd en dan wordt het resultaat van de formule berekend.

- # Pagina's documentsfoldertbl = **Aantal rijen in documenten / (2020 / (180 + 4))**
- # pagina's grammarsfoldertbl = **Aantal rijen in grammarsfoldertbl / (2020 / (180 + 4))**
- # Pagina's soufflant = **Aantal rijen in aanwijzingen / (2020 / (180 + 4))**
- # Pagina's scriptsfoldertbl = **Aantal rijen in scriptsfoldertbl / (2020 / (180 + 4))**

# pagina's documentsfoldertbl + # pagina's grammaticale - omslagvoud + # pagina's souvendertbl + # pagina's scriptsfoldertbl = **Totaal aantal pagina's voor tabellen met mappen**

## Bepaal het aantal pagina's dat wordt gebruikt in de tabellen voor bestandsindelingen van de optie

Voltooi dezelfde berekening om het totale aantal pagina's voor de bestandstabellen te bepalen die de eigenlijke positiebestanddelen bevatten. Voer deze opdrachten in met Microsoft SQL Query Analyzer:

- **SELECTEER COUNT(\*) UIT documentenbestand**
- **SELECTEER COUNT(\*) UIT grammarsfiletbl**
- **SELECTEER TELLER(\*) UIT VENSTER**
- **SELECTEER COUNT(\*) UIT scriptsfiletbl**

Bepaal het aantal pagina's dat door de inhoud van de tabellen op het bestand van de afbeelding wordt bezet met deze formule en vervang het **aantal rijen** met de tellingen die zijn verkregen uit de eerder genoemde opdrachten. Bereken de formule voor elke tabel en voeg het aantal pagina's toe.

- # pagina's documentaires = **Aantal rijen in documentsfiletbl / (2020 / (229 + 4))**
- # pagina's grammarsfiletbl = **Aantal rijen in grammarsfiletbl / (2020 / (229 + 4))**
- # Pagina's gevraagd = **Aantal rijen in aanwijzingen / (2020 / (229 + 4))**
- # Pagina's scriptsfiletbl = **Aantal rijen in scriptsfiletbl / (2020 / (229 + 4))**

# pagina's documentaires + # pagina's grammarsfiletbl + # pagina's loketten loketten + # pagina's scriptsfiletbl = **Totaal aantal pagina's voor bestandstabellen**

## berekenen

Voer deze berekeningen uit om de huidige schatting van de grootte van de reservegegevens te voltooien:

1. Bepaal het totale aantal pagina's dat nodig is om de huidige opslagplaats in Informix op te slaan met:

**Totaal aantal pagina's = Totaal aantal pagina's voor bestandstabellen + Totaal aantal pagina's voor mappentabellen**

2. Bepaal de totale ruimte in MB die op de pagina's wordt gebruikt:

**Totaal Aantal pagina's x 2 = Totale grootte in MB**

Als uit de berekeningen blijkt dat de contextuele informatie over de reserveonderdelen en mappen die momenteel geüpload worden in UCCX versie 7.x meer dan 3,4 MB bedraagt, wordt het aanbevolen het ontwerp van de reserveoptie opnieuw te bekijken. Hoewel de beschikbare vrije ruimte voor de contextuele informatie over reserveposten in **db\_cra\_cache** 6,9 MB is, wordt aangeraden om 50% beschikbaar te laten voor toekomstige groei. De groeiramingen en de maximaal toegestane bezette ruimte worden berekend per inzet, op basis van verwachte groeifactoren.

## Vragen

Omdat Prompts doorgaans de grootste verbruiker van de Repository-ruimte zijn, worden de methoden die worden gebruikt om het aantal Prompts in de Repository te verminderen in de rest van dit artikel besproken.

Als de Prompts die momenteel in de UCCX versie 7.x worden geüpload een belangrijk deel van de algehele opslagruimte voor opslagruimte voor opslagruimte voor opslaglocatie beslaan, dan dient u het juiste ontwerp, de opslag en het ophalen te herhalen voordat u naar UCCX versie 8.x gaat. Wanneer u probeert het juiste ontwerp opnieuw te activeren, bekijkt u deze opties:

- Vermindert het aantal korte mappen. Zoals getoond in de vorige berekeningen, bezet elke Prompt map een rij in de Repository database en verbruikt daarom ruimte tegen de **db\_cra\_limit**. Als u het aantal mappen vermindert, kunt u ruimte voor bestanden vrijgeven.
- Aanvragen voor gemeenschappelijke woorden of zinnen delen in toepassingen/scripts om het aantal oproepen dat in het systeem nodig is te beperken.
- Gebruik de door het systeem geboden PROMpts voor gemeenschappelijkheden tussen de PROMPEN, zoals aantallen en valuta's.
- Store Prompts op een afzonderlijke webserver binnen de onderneming, en haal de benodigde Prompts op aanvraag op en speel met Voice XML (VXML).

VXML wordt gebruikt om Prompts op bestelling van een off-box locatie op te halen en af te spelen. Als u grote hoeveelheden Prompts op een afzonderlijke webserver opslaat, kunt u:

- Elimineer de eerder genoemde migratieproblemen, omdat de Prompts niet langer opgeslagen worden in de **db\_cra\_cache**.
- Meer toegang en gemakkelijker beheer bieden.
- Verminder de back-uptijd voor het converteren, upgraden, migreren en maken van back-ups.

Hoewel er veel opties zijn om de aanpassing van Interactive Voice Response (IVR) in VXML te instrumenteren, worden het UCCX-script en de VXML-toepassing gebruikt om een Prompt van een off-box webserver op te halen en aan de beller af te spelen, als basis voor verdere ontwikkeling gebruikt. Net als andere aangepaste scripting in UCCX, worden de scripts die in deze sectie worden geboden als richtlijn en worden niet ondersteund door het Cisco Technical Assistance Center (TAC).

Opmerking: Een VXML toepassing wordt opgeroepen uit een UCCX script met de **Voice browser** stap. Meer informatie over de stap **Voice browser** wordt geleverd in de [Cisco Unified Contact Center Express Programming Guides](#).

De stap **Voice browser** verbruikt een VXML document. Dit document moet worden gemaakt na de stap **URL**-document maken en moet worden gehost op een webserver die niet aan UCCX is gekoppeld. Hoewel de VXML-toepassing is geschreven om de invoer van de beller te accepteren via Dual Tone Multi Frequency (DTMF), is deze toepassing alleen ontworpen om een prompt te spelen die buiten de box wordt gehost. Het kan echter worden uitgebreid om aanvullende functionaliteit in te bouwen. Aangenomen wordt dat de rest van het UCCX script, voordat de **Voice browser** stap wordt opgeroepen, de benodigde logica heeft om te bepalen welke prompt wordt weergegeven en een String variabele op de Prompt filename is ingesteld.

Aangezien het VXML document statisch is, maar de Prompt door het wordt gespeeld dynamisch is, wordt een server-side scripting taal gebruikt om het VXML document te maken. Dit kan elke server-side scripting taal zijn die de mogelijkheid heeft om de Content-type header van de XML **GET Application** response in te stellen. In dit voorbeeld wordt PHP gebruikt.

De PHP pagina is geschreven om een URL parameter te accepteren in een **GET aanvraag** die de audio Prompt naam representeert die wordt afgespeeld. De PHP pagina schakelt de VXML sjabloon aan met de Prompt filename die is doorgegeven in de URL-parameters van **GET Aanvraag** om het volledige VXML-document te vormen. Het stelt dan de Content-type header van de respons op XML in en stelt de inhoud van de reactie in op de VXML-inhoud.

```
<?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;
?>
```

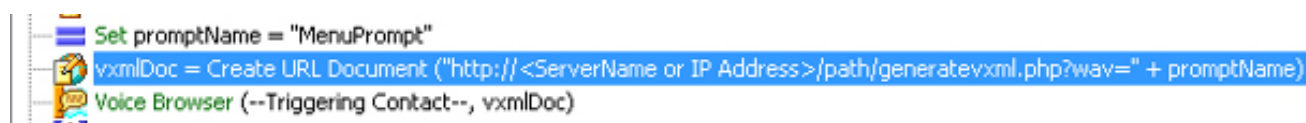
Om een goed gevormd VXML document te produceren, moet de voorbeeldpagina van PHP benaderd worden met een **GET aanvraag** die een parameter **wav** en een String waarde bevat, in de veronderstelling dat de voorbeeldpagina **generatevxml.php** genoemd wordt:

**http://**

Zorg ervoor dat het **MenuPrompt.wav** zich bevindt op de locatie op de externe webserver zoals gespecificeerd in de VXML sjabloon in de PHP pagina.

Gebruik in het UCCX-script de stap **URL-document maken** om een **GET aanvraag** van **generatevxml.php** uit te voeren door de basisURL van **http://<Server,naam of IP Address>/path/generatevxml.php?wav=** met de Prompt bestandsnaam die is afgeleid van de vorige scripting logica, aan te passen en plaats het resultaat in een documentvariabele.

Maak een stap **Voice** die de documentvariabele verwerkt.



Wanneer dit script wordt aangeroepen, mits zowel **generatevxml.php** als **MenuPrompt.wav** toegankelijk zijn op de webserver vanaf UCCX, **speelt** de Prompt.wav Prompt/Prompt aan de beller.

Als VXML toepassingen worden gebruikt om Prompts off-box op te slaan, zodat ze alleen toegankelijk zijn wanneer ze nodig zijn om ze aan de beller af te spelen, maakt het een grotere efficiëntie, beheersbaarheid en bruikbaarheid mogelijk. Dit is een probleem dat in overweging moet worden genomen als een UCCX versie 7.x-systeem is geüpgraded naar een UCCX versie 8.x-systeem en het aantal aanvragen zo is dat de inhoud van de contextuele informatie groter is dan de **db\_cra\_opslagplaats** of de **uccx\_sbspace**.