

Enregistrement CVP - Dépannez les programmes en double

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Dépannez les programmes en double dans l'enregistrement CVP](#)

[Diagnostiquez](#)

[Contournement](#)

[Enlevez le programme en double](#)

[Retirez les entrées en double des Tableaux](#)

[Remettez à l'état initial les exécutions récentes](#)

[Vérifier](#)

[Informations connexes](#)

Introduction

Ce document décrit un processus de dépannage des programmes en double pour la caractéristique d'enregistrement du Customer Voice Portal de Cisco (CVP).

Contribué par Mohamed Mohasseb, Aleksey Yankovskyy et Alexandre Levichev, ingénieurs TAC Cisco.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Microsoft Windows Server
- Cisco CVP
- Outil d'Access de DB d'Informix

[Composants utilisés](#)

Les informations dans ce document sont basées sur la version 11.0 de serveur CVP, mais elles s'appliquent également pour les versions précédentes.

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.

Dépannez les programmes en double dans l'enregistrement CVP

Dans des contrôles de table d'**agg_schedule** de base de données de **ciscoadmin** combien de fois chaque agrégation est exécutée. Dans certains cas après qu'une mise à jour la table soit rechargée avec un nouveau programme cependant le vieil existe toujours. Ceci a comme conséquence chaque agrégation étant exécutée deux fois. Ceci affectera défavorablement la précision des tableaux synoptiques car chacun aura deux fois le nombre de lignes, et par conséquent des comptes, insérés dans lui.

Diagnostiquez

Vérifiez s'il y a des doublons dans la table d'**agg_schedule**.

Ouvrez une session au serveur d'enregistrement CVP avec l'utilisateur de **cvp_dbadmin**.

Outil de Windows **CMD** de début. Commande de **dbaccess** de type. L'onglet **Connection** choisi > **se connectent**. Le serveur de base de données choisi de **cvp** et appuient sur entrent. Vous serez incité à saisir les qualifications. Compte de **cvp_dbadmin** d'utilisation.

Base de données choisie de **ciscoadmin@cvp**. Requête-langage choisi > **nouveau**. Exécutez cette commande.

```
UNLOAD TO schedule.txt SELECT * FROM agg_schedule
```

Fichier ouvert de **schedule.txt**. Selon la version CVP il pourrait se trouvent dans des répertoires de **C:\Users\Administrator** ou de **C:\db\Informix\etc\sysadmin**.

Il devrait avoir de seules entrées du planning de données CVP. Par exemple **call_15** ou **applicationsummary_daily** comme représenté dans cette image.

```
1 cvp_data|call_15||15|2016-09-07 07:33:57|dbdatetime|N|1|60 units day|
2 cvp_data|call_30|call_15|30|2000-01-01 00:00:00|dbdatetime|Y|2|60 units day|
3 cvp_data|call_hourly|call_15|60|2000-01-01 00:00:00|dbdatetime|Y|3|60 units day|
4 cvp_data|call_daily|call_15|DD|2016-09-07 07:33:57|dbdatetime|N|4|550 units day|
5 cvp_data|call_weekly|call_daily|WW|2016-09-07 07:33:57|dbdatetime|N|5|10 units year|
6 cvp_data|applicationsummary_15||15|2016-09-07 07:33:57|a.dbdatetime|N|6|60 units day|
7 cvp_data|applicationsummary_daily|applicationsummary_15|DD|2016-09-07 07:33:57|dbdatetime|N|7|550 units day|
8 cvp_data|applicationsummary_weekly|applicationsummary_daily|WW|2016-09-07 07:33:57|dbdatetime|N|8|10 units year|
9 cvp_data|call_monthly|call_daily|MM|2016-09-07 07:33:57|dbdatetime|N|9|40 units year|
10 cvp_data|applicationsummary_monthly|applicationsummary_daily|MM|2016-09-07 07:33:57|dbdatetime|N|10|40 units year|
11
```

Assurez-vous qu'il n'y a aucun type de doublon. Si des doublons sont trouvés alors implémentez le contournement.

Contournement

Enlevez le programme en double

Désactivez le travail récapitulatif (**CVPSummary**) dans le programmeur de tâche de Windows.

Le fichier ouvert de **schedule.txt** et retirent toutes les lignes en double. Là devrait rester seulement les 10 premières entrées.

Connectez à la base de données de **ciscoadmin** comme décrit avant et exécutez cette requête.

Cette commande supprime toutes les entrées de table d'**agg_schedule**.

- `DELETE FROM agg_schedule WHERE 1=1;`

Chargez les nouvelles valeurs à partir du fichier de **schedule.txt** qui ne contient aucun doublon à la table d'**agg_schedule**.

- `LOAD FROM schedule.txt INSERT INTO agg_schedule;`

Assurez qu'il n'y a aucun doublon actuel dans la table d'**agg_schedule**. La sortie du fichier de **schedule1.txt** devrait contenir seulement 10 entrées.

- `UNLOAD TO schedule1.txt SELECT * FROM agg_schedule;`

Retirez les entrées en double des Tableaux

Puisque la table 15-minute est la base dont toutes autres tables lui sont remplies doit être réparée d'abord.

Connectez à la base de données de **cvp_data**.

Exécutez ces commandes pour la table **call_15**.

```
SELECT distinct * FROM call_15 into temp t1 with no log;
TRUNCATE table call_15;
INSERT into call_15 select * from t1;
DROP table t1;
```

Répétez les mêmes étapes pour la table **applicationsummary_15**.

```
SELECT distinct * from applicationsummary_15 into temp t1 with no log;
TRUNCATE table applicationsummary_15;
INSERT into applicationsummary_15 select * from t1;
DROP table t1;
```

Note: Si le problème est répétition commencée il y a plus de 60 jours les étapes ci-dessus pour les tables quotidiennes, hebdomadaires et mensuelles.

Remettez à l'état initial les exécutions récentes

Remettez à l'état initial le champ de **lastrun** pour les tables 15-minute.

Découvrez le moment où les tables 15-minute étaient mises à jour. Exécutez ces commandes contre la base de données de **cvp_data**.

```
SELECT max(dbdatetime) FROM applicationsummary_15;
SELECT max(dbdatetime) FROM call_15;
```

Notez le nombre de jours depuis la dernière modification.

Exécutez cette requête pour remettre à l'état initial la dernière exécution pour les tables 15-minute

sur le DB de **ciscoadmin**. Dans ce l'exemple 15-minute les tables étaient il y a des 17 jours mis à jour la fois passée.

```
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 17 units day) WHERE
dst_tabname LIKE 'call_15';
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 17 units day) WHERE
dst_tabname LIKE 'applicationsummary_15';
```

Note: Remplacez "17" de la commande par le nombre de jours où vous avez obtenu de l'étape pour chacun des deux des tables.

Ensuite que des tables de l'étape 15-minute seront corrigées.

Puisque les tables 15-minute retiennent des données pour les 60 derniers jours, remettez à l'état initial la valeur de lastrun pour les tables quotidiennes, hebdomadaires et mensuelles et supprimez toutes les données pendant 60 jours de retour pour ces tables. Cette manière assure le ce moment prochain où le process start aggregation.bat, il remplira données dedans quotidiennes, hebdomadaire, les tables mensuelles avec les valeurs correctes.

Remettez à l'état initial le **lastrun** pour les tables quotidiennes, hebdomadaires et mensuelles avec ces commandes exécutées contre la base de données de **ciscoadmin**.

```
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 60 units day) WHERE
dst_tabname LIKE 'call_daily';
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 60 units day) WHERE
dst_tabname LIKE 'call_weekly';
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 60 units day) WHERE
dst_tabname LIKE 'call_monthly';
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 60 units day) WHERE
dst_tabname LIKE 'applicationsummary_daily';
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 60 units day) WHERE
dst_tabname LIKE 'applicationsummary_weekly';
UPDATE ciscoadmin:agg_schedule SET lastrun = (current year to day - 60 units day) WHERE
dst_tabname LIKE 'applicationsummary_monthly';
```

Supprimez toutes les données dedans quotidiennes, hebdomadaire et les tables mensuelles pendant 60 jours soutiennent.

```
DELETE FROM cvp_data:call_daily WHERE dbdatetime > (current - 60 units day);
DELETE FROM cvp_data:call_weekly WHERE dbdatetime > (current - 60 units day);
DELETE FROM cvp_data:call_monthly WHERE dbdatetime > (current - 60 units day);
```

```
DELETE FROM cvp_data:applicationsummary_daily WHERE dbdatetime > (current - 60 units day);
DELETE FROM cvp_data:applicationsummary_weekly WHERE dbdatetime > (current - 60 units
day);
DELETE FROM cvp_data:applicationsummary_monthly WHERE dbdatetime > (current - 60 units
day);
```

Activez le travail récapitulatif **CVPSummary** dans le programmeur de tâche pour que le processus d'agrégation reprenne.

Vérfiez

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Assurez que les tableaux synoptiques sont mis à jour avec ces commandes exécutées contre la base de données de **cvp_db**.

```
SELECT MAX(dbdatetime) FROM applicationsummary_15;  
SELECT max(dbdatetime) FROM applicationsummary_daily;  
SELECT max(dbdatetime) FROM applicationsummary_weekly;  
SELECT max(dbdatetime) FROM applicationsummary_monthly;
```

```
SELECT MAX(dbdatetime) FROM call_15;  
SELECT MAX(dbdatetime) FROM call_daily;  
SELECT MAX(dbdatetime) FROM call_weekly;  
SELECT MAX(dbdatetime) FROM call_monthly;
```

[Informations connexes](#)

- [Intégrez le client de l'écureuil SQL avec l'exemple de configuration d'enregistrement CVP](#)
- Soutien technique et documentation