

# Relatório CVP - Pesquisa defeitos programações duplicadas

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Pesquisa defeitos programações duplicadas no relatório CVP](#)

[Diagnostique](#)

[Solução](#)

[Remova a programação duplicada](#)

[Remova as entradas duplicadas das tabelas](#)

[Restaure execuções recentes](#)

[Verificar](#)

[Informações Relacionadas](#)

## Introdução

Este original descreve um processo da pesquisa de defeitos de programações duplicadas para a característica portal do relatório da Voz de cliente Cisco (CVP).

Contribuído por Mohamed Mohasseb, por Aleksey Yankovskyy e por Alexander Levichev, engenheiros de TAC da Cisco.

## Pré-requisitos

### Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Microsoft Windows server
- Cisco CVP
- Ferramenta do acesso de Informix DB

### [Componentes Utilizados](#)

A informação neste documento é baseada na versão de servidor 11.0 CVP, mas é igualmente aplicável para as versões anterior.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

# Pesquisa defeitos programações duplicadas no relatório CVP

No `agg_schedule` do base de dados do `ciscoadmin` apresente controles como cada agregação é executada frequentemente. Em alguns casos depois que uma elevação a tabela é recarregada com uma programação nova contudo velha ainda existe. Isto conduz a cada agregação que está sendo executada duas vezes. Isto impactará adversamente a precisão das tabelas de sumário como cada um terá duas vezes o número de linha, e daqui das contagens, introduzidas nele.

## Diagnostique

Verifique se há duplicatas na tabela do `agg_schedule`.

Entre ao server do relatório CVP com usuário do `cvp_dbadmin`.

Ligue a ferramenta do **CMD** de Windows. Datilografe o comando dos `dbaccess`. A aba seleta da **conexão > conecta**. Selecione o servidor de base de dados do `cvp` e pressione-o entram. Você será alertado datilografar dentro as credenciais. Use a conta do `cvp_dbadmin`.

Selecione o base de dados de `ciscoadmin@cvp`. Selecione o idioma de consulta > novo. Execute este comando.

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

Abra o arquivo de `schedule.txt`. Segundo a versão CVP pôde ser ficado situado em dobradores de `C:\Users\Administrator` ou de `C:\db\Informix\etc\sysadmin`.

Deve ter entradas originais da programação dos dados CVP. Por exemplo `call_15` ou `applicationsummary_daily` como representado nesta imagem.

```
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
```

Assegure-se de que não haja nenhum tipo da duplicata. Se as duplicatas são encontradas então execute a ação alternativa.

## Solução

### Remova a programação duplicada

Desabilite o trabalho sumário (**CVPSummary**) no task scheduler de Windows.

Abra o arquivo de `schedule.txt` e remova todas as linhas duplicadas. Lá deve permanecer somente as primeiras entradas 10.

Conecte ao base de dados do `ciscoadmin` como descrito antes e execute esta pergunta. Este comando suprime de todas as entradas da tabela do `agg_schedule`.

- `DELETE FROM agg_schedule WHERE 1=1;`

Carregue os valores novos do arquivo de **schedule.txt** que não contém nenhuma duplicatas à tabela do **agg\_schedule**.

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

Assegure-se de que não haja nenhuma duplicata atual na tabela do **agg\_schedule**. A saída do arquivo de **schedule1.txt** deve conter somente as entradas 10.

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

## Remova as entradas duplicadas das tabelas

Desde que a tabela 15-minute é a base de que todas tabelas restantes lhe são povoadas precisa de ser fixada primeiramente.

Conecte ao base de dados do **cvp\_data**.

Execute estes comandos para a tabela **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;
```

Repita as mesmas etapas para a tabela **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:** Se o problema é uma repetição começada de mais de 60 dias há as etapas acima para tabelas diárias, semanais e mensais.

## Restaure execuções recentes

Restaure o campo do **lastrun** para as tabelas 15-minute.

Encontre o tempo em que as tabelas 15-minute foram atualizadas. Execute estes comandos contra o base de dados do **cvp\_data**.

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

Escreva para baixo o número de dias desde a última atualização.

Execute esta pergunta para restaurar a última execução para as tabelas 15-minute no **ciscoadmin** DB. Este no exemplo 15-minute as tabelas foram atualizadas uma última vez de 17 dias há.

```
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:** Substitua "17" do comando com o número de dias onde você obteve da etapa para cada um tabelas.

Em seguida que as tabelas da etapa 15-minute estarão corrigidas.

Desde que as tabelas 15-minute retêm dados para os últimos 60 dias, restaure o valor do lastrun para tabelas diárias, semanais e mensais e suprima de todos os dados 60 dias para trás para estas tabelas. Esta maneira assegura a essa próxima vez em que o começo do processo aggregation.bat, ele povoará os dados dentro diários, tabelas semanais, mensais com os valores corretos.

Restaure o **lastrun** para tabelas diárias, semanais e mensais com estes comandos executados contra o base de dados do **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';
```

Suprima de todos os dados dentro diários, semanal e as tabelas mensais por 60 dias suportam.

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

Permita o trabalho sumário **CVPSummary** no task scheduler para que o processo da agregação comece outra vez.

## Verificar

Use esta seção para confirmar se a sua configuração funciona corretamente.

Assegure-se de que as tabelas de sumário estejam sendo atualizadas com estes comandos executados contra o base de dados do **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;
```

## Informações Relacionadas

- [Integre o cliente SQL do esquilo com exemplo de configuração do relatório CVP](#)
- Suporte técnico & documentação