Use CUCM CLI SQL Queries for DNs, Partitions, and User Associations

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Tables

SQL Queries

- List All DNs Associated with IP Phones
- List the DNs and Their Associated IP Phones
- List the Device Information and DNs with Associated Partitions
- Find Phones that Have a Line-Level User Association
- Find Phones that Do Not have a Line-Level User Association
- Find Phones that Have a Device-Level User Association

Introduction

This document describes SQL queries to obtain Directory Numbers (DNs) and partitions associated with IP phones.

Prerequisites

Requirements

Cisco recommends that you have knowledge of Cisco Unified Communications Manager (CUCM).

Components Used

The information in this document is based on CUCM Versions 8.X and later, but can work for earlier versions as well.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Tables

The SQL queries are formed with data from these tables:

- **Device** - This table contains device-level information like IP Phones and gateways.
- **NumPlan** - This table contains information about all patterns configured in CUCM.
- **DeviceNumPlanMap** - This table contains the mapping between data in Device table and NumPlan

table.

- **RoutePartition** - This table contains the information about all of the Partitions configured in CUCM
- **EndUser -** This table contains the information related to End Users.
- **DeviceNumPlanMapEndUserMap -** This table contains the mapping between data in the DeviceNumPlanMap table and the EndUser table.
- **EndUserDeviceMap -** This table contains the mapping between data in the Device table and EndUser table.

For more information about these tables, refer to the CUCM Data Dictionary for the respective version.

SQL Queries

The queries are written to find one association at a time in order to make them easier to understand.

List All DNs Associated with IP Phones

Here is a query that is designed to list all of the DNs associated with IP Phones:

```
run sql select dnorpattern as dn from numplan where pkid IN(select fknumplan from
devicenumplanmap where fkdevice IN (select pkid from device where tkclass =
1)) order by dn
```

Note: **tkclass = 1** is for IP phones. The **dnorpattern** is a column in the table numplan that has DNs associated with phones / hunt pilots / VM ports / route patterns / CTI RPs.

List the DNs and Their Associated IP Phones

Here is a query that is designed to list the DNs and their associated IP phones:

```
run sql select d.name, d.description, n.dnorpattern as DN from device as d,
numplan as n, devicenumplanmap as dnpm where dnpm.fkdevice = d.pkid and
dnpm.fknumplan = n.pkid and d.tkclass = 1
```

```
name            description                  dn
=============== ============================ ====
SEP0026CBBFF47E Line 1 - 7031; Line 2 - 7011 7011
SEP0026CBBFF47E Line 1 - 7031; Line 2 - 7011 7031
SEPC89C1DA3A5A9 Line 1 - 8021                8021
SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1213
SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1212
SEP503DE57D7DAC 8501                         8501
SEP0008308B289A Line 1 - 8023                8023
SEP0008308BBEBD Line 1 - 8032; Line 2 - 8012 8012
SEP0008308BBEBD Line 1 - 8032; Line 2 - 8012 8032
```

The query can be modified if you add the **tkmodel** number in order to list details for specific model IP

Phones. Here is a query for IP phone model 7945:

<#root>

```
run sql select d.name, d.description, n.dnorpattern as DN from device as d,
numplan as n, devicenumplanmap as dnpm where dnpm.fkdevice = d.pkid and
dnpm.fknumplan = n.pkid and d.tkclass = 1
```

and d.tkmodel='435'

Here is a query to obtain the **tkmodel** value for all IP phone models:

```
run sql select name,tkmodel from TypeProduct
```

Note: Any field can be filtered to include only information that is desired. For example, to filter on DN, add this at the end of the query: **where n.dnorpattern like '8%'**.
This query lists all entries that have a DN that begins with 8. The format is **where <*column name*> like '<*value*>%'.**

Note: Some other fields that provide useful information (user readable and not Primary Key Identifiers (PKIDs)) can be added to this query. Add them in between **run sql select** and **as DN**. These are the fields that can be added:
d.tkmodel - Use the query described in the previous note in order to obtain the value for each model.
d.tkdeviceprotocol - For Skinny Call Control Protocol (SCCP), the value is 0 and for Session Initiation Protocol (SIP), the value is 11.

List the Device Information and DNs with Associated Partitions

Here is a query that is designed to list the device information and DNs with associated partitions:

```
run sql select d.name, d.description, n.dnorpattern as DN, rp.name as partition
from device as d, numplan as n, devicenumplanmap as dnpm, routepartition as rp
where dnpm.fkdevice = d.pkid and dnpm.fknumplan = n.pkid and rp.pkid =
n.fkroutepartition and d.tkclass = 1
```

```
name            description                  dn   partition
=============== ============================ ==== ===========
SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1212 Internal_PT
SEP503DE57D7DAC 8501                         8501 Internal_PT
SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1213 Internal_PT
SEP0008308B289A Line 1 - 8023                8023 Internal_PT
```

Note: All DNs that do not have a partition are not listed here.

As more tables are added in the query, it is easier to use joins. The previous query can also be written:

```
run sql select d.name, d.description, n.dnorpattern as DN, rp.name as partition
from device as d inner join devicenumplanmap as dnpm on dnpm.fkdevice = d.pkid
inner join numplan as n on dnpm.fknumplan = n.pkid inner join routepartition as
rp on n.fkroutepartition=rp.pkid and d.tkclass = 1
```

The query can be modified if you add the **tkmodel** number in order to list details for specific model IP Phones. For IP phone model 7945, add **and d.tkmodel='435'** to the end of the query.

In order to obtain the **tkmodel** value for all IP phone models, enter:

```
run sql select name,tkmodel from TypeProduct
```

Note: Any field can be filtered to include only information that is desired. For example, in order to filter on DN, add this at the end of the query: **where n.dnorpattern like '8%'**
This query lists all entries that have a DN that begins with 8. The format is **where <*column name*> like '<*value*>%'.**

Note: Some other fields that provide useful information (user readable and not PKIDs) can be added to this query. Add them in between **run sql select** and **as DN**. These are the fields that can be added:
d.tkmodel - Use the query described in the previous note in order to obtain the value for each model.
d.tkdeviceprotocol - For SCCP, the value is 0 and for SIP, the value is 11.

Find Phones that Have a Line-Level User Association

Here is a query that is designed to find phones that have a line-level user association:

```
run sql select eu.userid, d.name, d.description, n.dnorpattern as DN, rp.name as
partition from device as d inner join devicenumplanmap as dnpm on dnpm.fkdevice =
d.pkid inner join devicenumplanmapendusermap as dnpeum on dnpeum.fkdevicenumplanmap=
dnpm.pkid inner join enduser as eu on dnpeum.fkenduser=eu.pkid inner join numplan as
n on dnpm.fknumplan = n.pkid inner join routepartition as rp on n.fkroutepartition=
rp.pkid and d.tkclass = 1
```

```
userid       name            description   dn   partition
============ =============== ============= ==== ===========
AMavilakandy SEP0008308B289A Line 1 - 8023 8023 Internal_PT
```

This query only provides information when the DN is in a partition. In order to include the ones in none partition, enter:

```
run sql select eu.userid, d.name, d.description, n.dnorpattern as DN from device
as d inner join devicenumplanmap as dnpm on dnpm.fkdevice = d.pkid inner join
devicenumplanmapendusermap as dnpeum on dnpeum.fkdevicenumplanmap=dnpm.pkid inner
join enduser as eu on dnpeum.fkenduser=eu.pkid inner join numplan as n on dnpm.fknumplan
= n.pkid and d.tkclass = 1
```

The query can be modified if you add the **tkmodel** number in order to list details for specific model IP phones. For IP phone model 7945, add **and d.tkmodel='435'** to the end of the query.

In order to obtain the **tkmodel** value for all IP Phone Models, enter:

```
run sql select name,tkmodel from TypeProduct
```

There aresome other fields that provide useful information (user readable and not PKIDs) that can be added to this query. Add them in between **run sql select** and **as DN.** These are the fields that can be added:

- eu.firstname
- eu.middlename
- eu.lastname
- eu.manager
- eu.department
- eu.telephonenumber
- eu.mailid
- eu.mobile
- eu.homephone
- eu.title
- d.tkmodel - Use the query mentioned previously in order to obtain the value for each model.
- d.tkdeviceprotocol - For SCCP the value is 0 and For SIP the value is 11.

Note: Any field can be filtered to include only information that is desired. For example, in order to filter on DN, add this at the end of the query: **where n.dnorpattern like '8%'**.
This query lists all entries that have a DN that begins with 8. The format is **where <*column name*> like '<*value*>%'.**

Find Phones that Do Not have a Line-Level User Association

Here is a query that is designed to find phones that do not have a line-level user association.

```
run sql select d.name, d.description, n.dnorpattern as DN, rp.name as partition
from device as d inner join devicenumplanmap as dnpm on dnpm.fkdevice = d.pkid
inner join devicenumplanmapendusermap as dnpeum on dnpeum.fkdevicenumplanmap!=
dnpm.pkid inner join numplan as n on dnpm.fknumplan = n.pkid inner join
routepartition as rp on n.fkroutepartition=rp.pkid and d.tkclass = 1
```

```
name            description                  dn   partition
=============== ============================ ==== ===========
SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1212 Internal_PT
```

```
SEP503DE57D7DAC 8501                          8501 Internal_PT
SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1213 Internal_PT
```

This query only provides information when the DN is in a partition. In order to include the ones in none partition, enter:

```
run sql select d.name, d.description, n.dnorpattern as DN from device as d inner
join devicenumplanmap as dnpm on dnpm.fkdevice = d.pkid inner join
devicenumplanmapendusermap as dnpeum on dnpeum.fkdevicenumplanmap!=dnpm.pkid
inner join numplan as n on dnpm.fknumplan = n.pkid and d.tkclass = 1
```

The query can be modified if you add the **tkmodel** number in order to list details for specific model IP Phones. For IP phone model 7945, add **and d.tkmodel='435'** to the end of the query.

In order to obtain the **tkmodel** value for all IP phone models, enter:

```
run sql select name,tkmodel from TypeProduct
```

There are some other fields that provide useful information (user readable and not PKIDs) that can be added to this query. Add them in between **run sql select** and **as DN**. These are the fields that can be added:

- eu.firstname
- eu.middlename
- eu.lastname
- eu.manager
- eu.department
- eu.telephonenumber
- eu.mailid
- eu.mobile
- eu.homephone
- eu.title
- d.tkmodel - Use the query mentioned previously in order to obtain the value for each model.
- d.tkdeviceprotocol - For SCCP, the value is 0 and for SIP, the value is 11.

Note: Any field can be filtered to include only information that is desired. For example, in order to filter on DN, add this at the end of the query: **where n.dnorpattern like '8%'**.
This lists all entries that have a DN that begins with 8. The format is **where <*column name*> like '<*value*>%'.**

Find Phones that Have a Device-Level User Association

Here is a query that is designed to find phones that have a device-level user association.

```
run sql select eu.userid, d.name, d.description, n.dnorpattern as DN, rp.name as
partition from device as d inner join devicenumplanmap as dnpm on dnpm.fkdevice =
```

```
d.pkid inner join enduserdevicemap as eudm on eudm.fkdevice=d.pkid inner join
enduser as eu on eudm.fkenduser=eu.pkid inner join numplan as n on dnpm.fknumplan =
n.pkid inner join routepartition as rp on n.fkroutepartition=rp.pkid and d.tkclass = 1
```

```
userid       name            description                  dn   partition
============ =============== ============================ ==== ===========
AMavilakandy SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1212 Internal_PT
AMavilakandy SEPA40CC3956C5C Line 1 - 1213; Line 2 - 1212 1213 Internal_PT
AMavilakandy SEP0008308B289A Line 1 - 8023                8023 Internal_PT
```

This query only provides information when the DN is in a partition. In order to include the ones in none partition, enter:

```
run sql select eu.userid, d.name, d.description, n.dnorpattern as DN from device
as d inner join devicenumplanmap as dnpm on dnpm.fkdevice = d.pkid inner join
enduserdevicemap as eudm on eudm.fkdevice=d.pkid inner join enduser as eu on
eudm.fkenduser=eu.pkid inner join numplan as n on dnpm.fknumplan = n.pkid and
d.tkclass = 1
```

The query can be modified if you add the **tkmodel** number in order to list details for specific model IP Phones. For IP Phone Model 7945, add **and d.tkmodel='435'** to the end of the query.

To obtain the **tkmodel** value for all IP Phone Models, enter:

```
run sql select name,tkmodel from TypeProduct
```

There are some other fields that provide useful information (user readable and not PKIDs) that can be added to this query. Add them in between **run sql select** and **as DN**. These are the fields that can be added:

- eu.firstname
- eu.middlename
- eu.lastname
- eu.manager
- eu.department
- eu.telephonenumber
- eu.mailid
- eu.mobile
- eu.homephone
- eu.title
- d.tkmodel - Use the query mentioned previously in order to obtain the value for each model.
- d.tkdeviceprotocol - For SCCP, the value is 0 and for SIP, the value is 11.

Note: Any field can be filtered to include only information that is desired. For example, in order to filter on DN, add this at the end of the query: **where n.dnorpattern like '8%'**.
This lists all entries that have a DN that begins with 8. The format is **where <*column name*> like**



'<*value*>%'.