--------------------------------------------------------

-- File created - Thursday-May-03-2018

--------------------------------------------------------

--------------------------------------------------------

-- DDL for Type NETSMART\_DETAIL\_REC

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TYPE "NG\_REPORTS"."NETSMART\_DETAIL\_REC" AS OBJECT (

ROWNUMBER NUMBER,

--AREA VARCHAR2(50 CHAR),

--REGION VARCHAR2(50 CHAR),

TERRITORY VARCHAR2(100 CHAR),

MARKET\_TERRITORY VARCHAR2(100 CHAR),

SUB\_MARKET VARCHAR2(100 CHAR),

LEAF\_DOMAIN\_NAME VARCHAR2(80 CHAR),

TID VARCHAR2(50 CHAR),

TID\_TYPE VARCHAR2(50 CHAR),

LOCATION VARCHAR2(80 CHAR),

CONTACTS VARCHAR2(80 CHAR),

IP\_ADDR VARCHAR2(40 CHAR),

MATCH\_CODE VARCHAR2(20 CHAR),

MATCH\_STATUS VARCHAR2(500 CHAR),

NE\_INST\_ID NUMBER,

EQP\_NAME VARCHAR2(100 CHAR),

SITE\_REFERENCE\_ID NUMBER,

SITE\_NAME VARCHAR2(100 CHAR),

EQP\_TYPE VARCHAR2(50 CHAR),

EQP\_VENDOR VARCHAR2(50 CHAR),

EQP\_MODEL VARCHAR2(50 CHAR),

NUM\_XCONNECT NUMBER,

DISCREPANCY NUMBER,

LIVE\_XCONNECTS NUMBER,

OTHER\_XCONNECTS NUMBER,

COMPLIANCE VARCHAR2(10 CHAR)

);

/

--------------------------------------------------------

-- DDL for Type NETSMART\_DETAIL\_TBL

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TYPE "NG\_REPORTS"."NETSMART\_DETAIL\_TBL" AS TABLE OF NETSMART\_DETAIL\_REC;

/

--------------------------------------------------------

-- DDL for Type NETSMART\_SUMMARY\_REC

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TYPE "NG\_REPORTS"."NETSMART\_SUMMARY\_REC" AS OBJECT (

--"AREA" VARCHAR2(50 CHAR),

--"REGION" VARCHAR2(50 CHAR),

TERRITORY\_MARKET\_SUB\_MARKET VARCHAR2(100 CHAR),

TERRITORY VARCHAR2(100 CHAR),

MARKET\_TERRITORY VARCHAR2(100 CHAR),

SUB\_MARKET VARCHAR2(100 CHAR),

TOTAL\_NE NUMBER,

DUPLICATE\_NE NUMBER,

MATCHED\_NE NUMBER,

PERCENT\_MATCHED NUMBER,

PASSED\_AUDIT NUMBER,

NAMING\_COMPLIANCE NUMBER,

DISCOVERED\_XCONNECTS NUMBER,

LIVE\_XCONNECTS NUMBER,

OTHER\_XCONNECTS NUMBER,

TOTAL\_COMPLIANCE NUMBER

);

/

--------------------------------------------------------

-- DDL for Type NETSMART\_SUMMARY\_TBL

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TYPE "NG\_REPORTS"."NETSMART\_SUMMARY\_TBL" AS TABLE OF NETSMART\_SUMMARY\_REC;

/

--------------------------------------------------------

-- DDL for Type NETSMART\_XCONNECT\_REC

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TYPE "NG\_REPORTS"."NETSMART\_XCONNECT\_REC" AS OBJECT (

ROWNUMBER NUMBER,

TID VARCHAR2(50 CHAR),

DEVICE\_XCONNECTS VARCHAR2(80 CHAR),

AUDIT\_STATUS VARCHAR2(20 CHAR),

TRAIL\_ID NUMBER,

TRAIL\_NAME VARCHAR2(100 CHAR),

PATH\_STATUS VARCHAR2(30 CHAR),

BANDWIDTH VARCHAR2(30 CHAR),

COMMENTS VARCHAR2(500 CHAR)

);

/

--------------------------------------------------------

-- DDL for Type NETSMART\_XCONNECT\_TBL

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TYPE "NG\_REPORTS"."NETSMART\_XCONNECT\_TBL" AS TABLE OF NETSMART\_XCONNECT\_REC;

/

--------------------------------------------------------

-- DDL for Sequence CIENA\_6500\_NE\_EQUIP\_ID\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP\_ID\_SEQ" MINVALUE 1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 3203 NOCACHE NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence CIENA\_6500\_PARSE\_INST\_ID\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."CIENA\_6500\_PARSE\_INST\_ID\_SEQ" MINVALUE 0 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 794974 NOCACHE NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN\_SEQ" MINVALUE 0 MAXVALUE 9223372036854775807 INCREMENT BY 1 START WITH 10176 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence JV\_CDO\_CLASS\_CDO\_CLASS\_PK\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."JV\_CDO\_CLASS\_CDO\_CLASS\_PK\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence JV\_COMMIT\_COMMIT\_PK\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."JV\_COMMIT\_COMMIT\_PK\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence JV\_GLOBAL\_ID\_GLOBAL\_ID\_PK\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."JV\_GLOBAL\_ID\_GLOBAL\_ID\_PK\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence JV\_SNAPSHOT\_SNAPSHOT\_PK\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."JV\_SNAPSHOT\_SNAPSHOT\_PK\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence NETSMART\_EQPT\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."NETSMART\_EQPT\_SEQ" MINVALUE 1 MAXVALUE 9999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence NETSMART\_XCONNECT\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."NETSMART\_XCONNECT\_SEQ" MINVALUE 1 MAXVALUE 9999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence PORT\_PORT\_REFERENCE\_ID\_1SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."PORT\_PORT\_REFERENCE\_ID\_1SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence PORT\_PORT\_REFERENCE\_ID\_2SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."PORT\_PORT\_REFERENCE\_ID\_2SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Sequence PORT\_PORT\_REFERENCE\_ID\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "NG\_REPORTS"."PORT\_PORT\_REFERENCE\_ID\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE ;

--------------------------------------------------------

-- DDL for Table BRIX\_CSR\_DISCOVERY\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BRIX\_CSR\_DISCOVERY\_WK"

( "VERIFIER\_NAME" VARCHAR2(50 BYTE),

"CELL\_SITE\_NAME" VARCHAR2(50 BYTE),

"MAC" VARCHAR2(50 BYTE),

"CST\_MODIFY\_DATE" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BRIX\_CSR\_EXTRACT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BRIX\_CSR\_EXTRACT"

( "VERIFIER\_NAME" VARCHAR2(50 BYTE),

"CELL\_SITE\_NAME" VARCHAR2(50 BYTE),

"MAC" VARCHAR2(50 BYTE),

"CST\_MODIFYDATE" VARCHAR2(50 BYTE)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE( INITIAL 8388608 NEXT 8388608 MINEXTENTS 1

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_CELL\_DCG\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_CELL\_DCG\_MAP"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"CELL\_NUMBER" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE,

"RADIO\_CONFIG" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_CELL\_DCG\_MAP\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_CELL\_DCG\_MAP\_WRK"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"CELL\_NUMBER" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE,

"RADIO\_CONFIG" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_DCG\_SPAN\_VOICE\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_DCG\_SPAN\_VOICE\_MAP"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_DCG\_SPAN\_VOICE\_MAP\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_DCG\_SPAN\_VOICE\_MAP\_WRK"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_ETHERNET\_INV

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_ETHERNET\_INV"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_ETHERNET\_INV\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_ETHERNET\_INV\_WRK"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_METRO\_INV

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_METRO\_INV"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"BCKHTYPE" VARCHAR2(100 BYTE),

"CMTYPE" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_METRO\_INV\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_METRO\_INV\_WRK"

( "SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONGNAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"BCKHTYPE" VARCHAR2(100 BYTE),

"CMTYPE" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_MTX\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_MTX\_MAP"

( "VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_MTX" VARCHAR2(100 BYTE),

"MTX\_STATUS" VARCHAR2(20 BYTE) DEFAULT 'Live',

"BSM\_STATUS" VARCHAR2(20 BYTE) DEFAULT 'Live'

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BSM\_MTX\_MAP\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BSM\_MTX\_MAP\_WRK"

( "VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_MTX" VARCHAR2(100 BYTE),

"MTX\_STATUS" VARCHAR2(20 BYTE),

"BSM\_STATUS" VARCHAR2(20 BYTE)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_AUDIT\_DATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_AUDIT\_DATA"

( "STRIPPED\_EC\_CIRCUIT\_ID" VARCHAR2(60 BYTE),

"BAN\_CUSTOM\_FIELD\_1" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_3" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_4" VARCHAR2(50 BYTE),

"INVENTORY\_VENDOR\_CODE" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_2" VARCHAR2(50 BYTE),

"SERV\_ESTAB\_DATE" VARCHAR2(50 BYTE),

"CHARGE\_AMOUNT" VARCHAR2(50 BYTE),

"MASTER\_BAN" VARCHAR2(50 BYTE),

"BAN" VARCHAR2(50 BYTE),

"BILL\_DATE\_MONTH" VARCHAR2(20 BYTE),

"BILL\_DATE\_YEAR" VARCHAR2(20 BYTE),

"BILL\_DATE" VARCHAR2(50 BYTE),

"MANUAL\_ENTRY" VARCHAR2(10 BYTE),

"BTOD\_EXTRACT\_DATE" VARCHAR2(50 BYTE) DEFAULT TO\_CHAR(SYSDATE, 'yyyymmdd')

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_AUDIT\_DATA\_PROCESS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_AUDIT\_DATA\_PROCESS"

( "STRIPPED\_EC\_CIRCUIT\_ID" VARCHAR2(60 BYTE),

"BTOD\_CIRC\_STRIP" VARCHAR2(60 BYTE),

"BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_2" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_1" VARCHAR2(50 BYTE),

"MASTER\_BAN" VARCHAR2(50 BYTE),

"BAN" VARCHAR2(50 BYTE),

"BTOD\_AMT" NUMBER,

"BTOD\_BAN\_STRIP" VARCHAR2(50 BYTE),

"INVENTORY\_VENDOR\_CODE" VARCHAR2(50 BYTE),

"BTOD\_BD" DATE,

"BTOD\_BD\_CHAR" VARCHAR2(50 BYTE),

"BTOD\_EXTRACT\_DATE" DATE,

"BAN\_CUSTOM\_FIELD\_4" VARCHAR2(50 BYTE),

"BILL\_DATE\_MONTH" VARCHAR2(20 BYTE),

"BILL\_DATE\_YEAR" VARCHAR2(20 BYTE),

"MANUAL\_ENTRY" VARCHAR2(10 BYTE),

"BAN\_CUSTOM\_FIELD\_3" VARCHAR2(50 BYTE),

"SERV\_ESTAB\_DATE" DATE,

"SEQNO" NUMBER,

"PROCESSED" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_NTLS\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_NTLS\_AUDIT"

( "BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_2" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_1" VARCHAR2(50 BYTE),

"MASTER\_BAN" VARCHAR2(50 BYTE),

"BAN" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_4" VARCHAR2(50 BYTE),

"STRIPPED\_EC\_CIRCUIT\_ID" VARCHAR2(60 BYTE),

"INVENTORY\_VENDOR\_CODE" VARCHAR2(50 BYTE),

"CHARGE\_AMOUNT" NUMBER,

"BILL\_DATE\_MONTH" VARCHAR2(20 BYTE),

"BILL\_DATE\_YEAR" VARCHAR2(20 BYTE),

"MANUAL\_ENTRY" VARCHAR2(10 BYTE),

"BTOD\_EXTRACT\_DATE" DATE,

"BTOD\_NTLS\_MATCH" VARCHAR2(1 BYTE),

"NTLS\_TERRITORY" VARCHAR2(100 BYTE),

"NTLS\_MARKET" VARCHAR2(50 BYTE),

"NTLS\_SUB\_MARKET" VARCHAR2(100 BYTE),

"NTLS\_SEGMENT\_ID" NUMBER,

"NTLS\_SEGMENT\_NAME" VARCHAR2(60 BYTE),

"NTLS\_TYPE" VARCHAR2(30 BYTE),

"NTLS\_VENDOR" VARCHAR2(30 BYTE),

"NTLS\_BANDWIDTH" VARCHAR2(30 BYTE),

"NTLS\_STATUS" VARCHAR2(20 BYTE),

"NTLS\_CURR\_PATH\_INST\_ID" NUMBER,

"NTLS\_NEXT\_PATH\_INST\_ID" NUMBER,

"NTLS\_EXTRACT\_DATE" DATE,

"UNIQUELY\_IDENTIFIED" VARCHAR2(1 BYTE),

"BILL\_DATE" DATE,

"BAN\_CUSTOM\_FIELD\_3" VARCHAR2(50 BYTE),

"SERV\_ESTAB\_DATE" DATE,

"MATCH\_CODE" VARCHAR2(20 BYTE),

"MATCH\_STATUS" VARCHAR2(50 BYTE),

"NTLS\_BAN" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_NTLS\_AUDIT\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_NTLS\_AUDIT\_SUMMARY"

( "TERRITORY" VARCHAR2(100 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"CKT\_COUNT" NUMBER,

"CKTS\_MATCHED" NUMBER,

"CKTS\_EXCP" NUMBER,

"CKTS\_MATCHED\_PCT\_STR" VARCHAR2(15 BYTE),

"CKTS\_MATCHED\_PCT\_NUM" NUMBER,

"EXTRACT\_DATE" DATE,

"TERRITORY\_MARKET\_SUB" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_NTLS\_AUDIT\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_NTLS\_AUDIT\_WRK"

( "BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_2" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_1" VARCHAR2(50 BYTE),

"MASTER\_BAN" VARCHAR2(50 BYTE),

"BAN" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_4" VARCHAR2(50 BYTE),

"STRIPPED\_EC\_CIRCUIT\_ID" VARCHAR2(60 BYTE),

"INVENTORY\_VENDOR\_CODE" VARCHAR2(50 BYTE),

"CHARGE\_AMOUNT" NUMBER,

"BILL\_DATE\_MONTH" VARCHAR2(20 BYTE),

"BILL\_DATE\_YEAR" VARCHAR2(20 BYTE),

"MANUAL\_ENTRY" VARCHAR2(10 BYTE),

"BTOD\_EXTRACT\_DATE" DATE,

"BTOD\_NTLS\_MATCH" VARCHAR2(1 BYTE),

"NTLS\_TERRITORY" VARCHAR2(100 BYTE),

"NTLS\_MARKET" VARCHAR2(50 BYTE),

"NTLS\_SUB\_MARKET" VARCHAR2(100 BYTE),

"NTLS\_SEGMENT\_ID" NUMBER,

"NTLS\_SEGMENT\_NAME" VARCHAR2(60 BYTE),

"NTLS\_TYPE" VARCHAR2(30 BYTE),

"NTLS\_VENDOR" VARCHAR2(30 BYTE),

"NTLS\_BANDWIDTH" VARCHAR2(30 BYTE),

"NTLS\_STATUS" VARCHAR2(20 BYTE),

"NTLS\_CURR\_PATH\_INST\_ID" NUMBER,

"NTLS\_NEXT\_PATH\_INST\_ID" NUMBER,

"NTLS\_EXTRACT\_DATE" DATE,

"UNIQUELY\_IDENTIFIED" VARCHAR2(1 BYTE),

"BILL\_DATE" DATE,

"BAN\_CUSTOM\_FIELD\_3" VARCHAR2(50 BYTE),

"SERV\_ESTAB\_DATE" DATE,

"MATCH\_CODE" VARCHAR2(20 BYTE),

"MATCH\_STATUS" VARCHAR2(50 BYTE),

"NTLS\_BAN" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_NTLS\_DOMAIN\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_NTLS\_DOMAIN\_MAP"

( "BTOD\_BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"NTLS\_DOMAIN\_ID" NUMBER(9,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_NTLS\_PP\_T

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_NTLS\_PP\_T"

( "BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_2" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_1" VARCHAR2(50 BYTE),

"MASTER\_BAN" VARCHAR2(50 BYTE),

"BAN" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_4" VARCHAR2(50 BYTE),

"STRIPPED\_EC\_CIRCUIT\_ID" VARCHAR2(60 BYTE),

"INVENTORY\_VENDOR\_CODE" VARCHAR2(50 BYTE),

"CHARGE\_AMOUNT" NUMBER,

"BTOD\_EXTRACT\_DATE" DATE,

"NTLS\_TERRITORY" VARCHAR2(100 BYTE),

"NTLS\_MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"NTLS\_SUB\_MARKET" VARCHAR2(100 BYTE),

"BTOD\_NTLS\_MATCH" CHAR(1 BYTE),

"NTLS\_SEGMENT\_ID" NUMBER,

"NTLS\_SEGMENT\_NAME" VARCHAR2(200 BYTE),

"NTLS\_TYPE" VARCHAR2(500 BYTE),

"NTLS\_VENDOR" VARCHAR2(330 BYTE),

"NTLS\_BANDWIDTH" VARCHAR2(90 BYTE),

"NTLS\_STATUS" VARCHAR2(100 BYTE),

"NTLS\_EXTRACT\_DATE" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_3" VARCHAR2(50 BYTE),

"BTOD\_IN\_SERVICE\_DATE" DATE,

"NTLS\_IN\_SERVICE\_DATE" DATE,

"ORIG\_COMPANY\_NUM" VARCHAR2(10 BYTE),

"ORIG\_COMPANY\_NAME" VARCHAR2(100 BYTE),

"INVOICE" VARCHAR2(200 BYTE),

"INVOICE\_STATUS" VARCHAR2(200 BYTE),

"LATEST\_BILL\_DATE" DATE,

"Z\_SIDE\_SITE" VARCHAR2(300 BYTE),

"Z\_SIDE\_LATITUDE" VARCHAR2(20 BYTE),

"Z\_SIDE\_LONGITUDE" VARCHAR2(20 BYTE),

"ESTIMATED\_COST" NUMBER(10,2),

"RENEWAL\_DATE" DATE,

"TERM\_DURATION" NUMBER(13,3),

"PATH911" VARCHAR2(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_PP\_DATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_PP\_DATA"

( "STRIPPED\_EC\_CIRCUIT\_ID" VARCHAR2(60 BYTE),

"BAN\_CUSTOM\_FIELD\_1" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_3" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_4" VARCHAR2(50 BYTE),

"INVENTORY\_VENDOR\_CODE" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_2" VARCHAR2(50 BYTE),

"SERV\_ESTAB\_DATE" VARCHAR2(50 BYTE),

"CHARGE\_AMOUNT" VARCHAR2(50 BYTE),

"MASTER\_BAN" VARCHAR2(50 BYTE),

"BAN" VARCHAR2(50 BYTE),

"BILL\_DATE\_MONTH" VARCHAR2(20 BYTE),

"BILL\_DATE\_YEAR" VARCHAR2(20 BYTE),

"BILL\_DATE" VARCHAR2(50 BYTE),

"MANUAL\_ENTRY" VARCHAR2(10 BYTE),

"IN\_SERVICE\_DATE" VARCHAR2(60 BYTE),

"ORIG\_COMPANY\_NUM" VARCHAR2(10 BYTE),

"ORIG\_COMPANY\_NAME" VARCHAR2(100 BYTE),

"INVOICE" VARCHAR2(50 BYTE),

"TERM\_LENGTH" VARCHAR2(10 BYTE),

"BTOD\_EXTRACT\_DATE" VARCHAR2(50 BYTE) DEFAULT TO\_CHAR(SYSDATE, 'yyyymmdd')

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table BTOD\_UPDATE\_DATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."BTOD\_UPDATE\_DATA"

( "STRIPPED\_EC\_CIRCUIT\_ID" VARCHAR2(60 BYTE),

"BAN\_CUSTOM\_FIELD\_1" VARCHAR2(50 BYTE),

"BAN\_CUSTOM\_FIELD\_3" VARCHAR2(50 BYTE),

"INVENTORY\_VENDOR\_CODE" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_1" VARCHAR2(50 BYTE),

"BAN\_CATEGORY\_2" VARCHAR2(50 BYTE),

"MASTER\_BAN" VARCHAR2(50 BYTE),

"BAN" VARCHAR2(50 BYTE),

"BILL\_DATE\_MONTH" VARCHAR2(50 BYTE),

"BILL\_DATE\_YEAR" VARCHAR2(50 BYTE),

"BILL\_DATE" VARCHAR2(50 BYTE),

"CHARGE\_AMOUNT" VARCHAR2(50 BYTE),

"MANUAL\_ENTRY\_IND" VARCHAR2(50 BYTE),

"LOGGED\_DATE" VARCHAR2(50 BYTE),

"INVOICE\_SEQUENCE\_ID" VARCHAR2(50 BYTE),

"BTOD\_EXTRACT\_DATE" VARCHAR2(20 BYTE) DEFAULT to\_char(sysdate,'yyyymmdd')

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CARD

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CARD"

( "CARD\_REFERENCE\_ID" NUMBER(18,0),

"CARD\_NAME" VARCHAR2(100 BYTE),

"CARD\_DESCRIPTION" VARCHAR2(100 BYTE),

"CARD\_TYPE" VARCHAR2(50 BYTE),

"CARD\_SPEC\_REF\_ID" NUMBER(18,0),

"EQP\_REFERENCE\_ID" NUMBER(18,0),

"SLOT\_REFERENCE\_ID" NUMBER(18,0),

"SLOT\_NAME" VARCHAR2(32 BYTE),

"INV\_STATUS" VARCHAR2(32 BYTE),

"IP\_ADDRESS" VARCHAR2(32 BYTE),

"CARD\_ROLE" VARCHAR2(32 BYTE),

"CARD\_FUNCTIONAL\_TYPE" VARCHAR2(32 BYTE),

"MAX\_PORTS" NUMBER(18,0),

"NBR\_SUB\_CARDS" NUMBER(18,0),

"SUBSLOTS" NUMBER(18,0),

"IS\_CHASSIS" VARCHAR2(3 BYTE),

"SLOT\_OCCUPANCY" NUMBER(18,0),

"PARENT\_CARD\_REF\_ID" NUMBER(18,0),

"IN\_SERVICE\_DATE" VARCHAR2(32 BYTE),

"INSTALLED\_DATE" VARCHAR2(32 BYTE),

"LAST\_MODIFIED\_TIME\_STAMP" VARCHAR2(32 BYTE),

"LAST\_MODIFIED\_BY" VARCHAR2(100 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE),

"PHYSICAL\_CARD\_REFERENCE\_ID" NUMBER(18,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CARD\_ATTRIBUTES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CARD\_ATTRIBUTES"

( "CARD\_ATTRIBUTES\_ID" NUMBER(18,0),

"CARD\_REFERENCE\_ID" NUMBER(18,0),

"CARD\_DESCPT\_ID" NUMBER(18,0),

"CARD\_GROUP\_NAME" VARCHAR2(50 BYTE),

"CARD\_NAME" VARCHAR2(50 BYTE),

"CARD\_VALUE" VARCHAR2(4000 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_DISC\_METADATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_DISC\_METADATA"

( "COMMAND\_ID" NUMBER,

"COMMAND" VARCHAR2(200 BYTE),

"RESPONSE\_SYNTAX" VARCHAR2(4000 BYTE),

"CMD\_START\_REGEX" VARCHAR2(200 BYTE),

"CMD\_RESPONSE\_REGEX" VARCHAR2(250 BYTE),

"CMD\_END\_REGEX" VARCHAR2(100 BYTE),

"OBJECT\_TYPE" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_ISSUES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_ISSUES"

( "ISSUE\_ID" NUMBER,

"ERROR" VARCHAR2(256 BYTE),

"RESOLUTION" VARCHAR2(256 BYTE),

"COMMENTS" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_EQUIP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP"

( "ID" NUMBER,

"DEVICE\_NAME" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"IP" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_EQUIP\_SAVE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP\_SAVE"

( "ID" NUMBER,

"DEVICE\_NAME" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"IP" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_EQUIP\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP\_WK"

( "ID" NUMBER,

"DEVICE\_NAME" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"IP" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_ERRORS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_ERRORS"

( "EQ\_ID" NUMBER,

"COMMAND\_ID" NUMBER,

"ISSUE\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_ERRORS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_ERRORS\_WK"

( "EQ\_ID" NUMBER,

"COMMAND\_ID" NUMBER,

"ISSUE\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_KEY\_VALUE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_KEY\_VALUE"

( "PARSED\_INST\_ID" NUMBER,

"KEY" VARCHAR2(256 BYTE),

"VALUE" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_KEY\_VALUE\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_KEY\_VALUE\_WK"

( "PARSED\_INST\_ID" NUMBER,

"KEY" VARCHAR2(256 BYTE),

"VALUE" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_PARSED\_DATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_PARSED\_DATA"

( "INST\_ID" NUMBER,

"COMMAND\_ID" NUMBER,

"EQ\_ID" NUMBER,

"RESP\_FIELD\_1" VARCHAR2(250 BYTE),

"RESP\_FIELD\_2" VARCHAR2(250 BYTE),

"RESP\_FIELD\_3" VARCHAR2(2000 BYTE),

"RESP\_FIELD\_4" VARCHAR2(1000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_PARSED\_DATA\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_PARSED\_DATA\_WK"

( "INST\_ID" NUMBER,

"COMMAND\_ID" NUMBER,

"EQ\_ID" NUMBER,

"RESP\_FIELD\_1" VARCHAR2(250 BYTE),

"RESP\_FIELD\_2" VARCHAR2(250 BYTE),

"RESP\_FIELD\_3" VARCHAR2(2000 BYTE),

"RESP\_FIELD\_4" VARCHAR2(1000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIENA\_6500\_NE\_TID\_LIST

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_TID\_LIST"

( "DEVICE\_NAME" VARCHAR2(100 BYTE),

"IP" VARCHAR2(50 BYTE),

"SYS\_UPDATE\_DATE" DATE,

"ISSUE\_ID" NUMBER,

"TID\_FILE\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIRC\_SEG\_NEW\_DECOM\_DETAIL

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL"

( "RPT\_ID" NUMBER(10,2) DEFAULT 0,

"DOMAIN\_ID" NUMBER(10,2),

"DOMAIN\_NAME" VARCHAR2(50 BYTE),

"AREA" VARCHAR2(50 BYTE),

"REGION" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SEGMENT\_ID" NUMBER(10,2),

"SEGMENT\_NAME" VARCHAR2(60 BYTE),

"SEGMENT\_TYPE" VARCHAR2(30 BYTE),

"Z\_SITE\_ID" NUMBER(10,2),

"SITE\_NAME" VARCHAR2(100 BYTE),

"VENDOR" VARCHAR2(30 BYTE),

"STATUS" VARCHAR2(20 BYTE),

"ACTION\_TYPE" VARCHAR2(10 BYTE),

"COMMIT\_DATE" DATE,

"AUTHOR" VARCHAR2(50 BYTE),

"BANDWIDTH" VARCHAR2(30 BYTE),

"MRC" NUMBER(10,2),

"DISCONNECT\_PON" VARCHAR2(3100 BYTE),

"DISCONNECT\_ORDER\_NUM" VARCHAR2(3100 BYTE),

"DISCONNECT\_ORDER\_TS" VARCHAR2(30 BYTE),

"DISCONNECT\_TS" VARCHAR2(30 BYTE),

"BILLING\_CODE" VARCHAR2(30 BYTE),

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN"

( "RPT\_ID" NUMBER(10,2),

"PROC\_START\_TS" DATE,

"PROC\_END\_TS" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIRC\_SEG\_NEW\_DECOM\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY"

( "PERIOD" VARCHAR2(50 BYTE),

"PERIOD\_TYPE" VARCHAR2(50 BYTE),

"AREA" VARCHAR2(50 BYTE),

"REGION" VARCHAR2(50 BYTE),

"COUNT\_NEW" NUMBER(10,2),

"COUNT\_DECOM" NUMBER(10,2),

"COUNT\_NEW\_TOTAL" NUMBER(10,2) DEFAULT 0,

"COUNT\_NEW\_BY\_TESS" NUMBER(10,2) DEFAULT 0,

"COUNT\_NEW\_NOT\_TESS\_CONTRACT" NUMBER(10,2) DEFAULT 0,

"COUNT\_NEW\_NOT\_TESS\_NO\_CONTRACT" NUMBER(10,2) DEFAULT 0,

"PERCENTAGE\_NEW\_BY\_TESS" NUMBER(10,2) DEFAULT 0.00,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CIRC\_TEST\_RESULTS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CIRC\_TEST\_RESULTS"

( "TEST\_ID" NUMBER,

"SERVICE\_NAME" VARCHAR2(50 BYTE),

"VLAN\_NUMBER" NUMBER,

"PORT\_NAME" VARCHAR2(100 BYTE),

"TEST\_TYPE" VARCHAR2(2 BYTE),

"FILE\_NAME" VARCHAR2(100 BYTE),

"TAM\_CUSTOM\_DATA" VARCHAR2(100 BYTE),

"URL" VARCHAR2(200 BYTE),

"TEST\_START\_DATE" VARCHAR2(50 BYTE),

"TEST\_END\_DATE" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE,

"SERVICE\_ID" VARCHAR2(10 BYTE),

"FRAME\_DELAY" VARCHAR2(10 BYTE),

"FRAME\_DELAY\_VARIATION" VARCHAR2(10 BYTE),

"FRAME\_LOSS" VARCHAR2(10 BYTE),

"WL\_OBJECT\_ID" VARCHAR2(30 BYTE),

"WL\_STATUS" VARCHAR2(30 BYTE),

"FRAME\_SIZE" NUMBER,

"BC\_SOURCE" VARCHAR2(10 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CISCO\_ANA\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CISCO\_ANA\_WK"

( "CONTEXT" VARCHAR2(50 BYTE),

"A\_END\_POINT" VARCHAR2(100 BYTE),

"Z\_END\_POINT" VARCHAR2(100 BYTE),

"LINK\_TYPE" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CKIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CKIT"

( "DATA1" VARCHAR2(1000 BYTE),

"DATA\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CLLI\_DOMAINS\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CLLI\_DOMAINS\_MAP"

( "CLLI" VARCHAR2(8 BYTE),

"LEAF\_DOMAIN\_INST\_ID" NUMBER,

"STATUS" VARCHAR2(10 BYTE) DEFAULT 'Active',

"COMMENTS" VARCHAR2(4000 BYTE),

"LAST\_MOD\_BY" VARCHAR2(50 BYTE),

"LAST\_MOD\_TS" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CLLI\_DOMAIN\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CLLI\_DOMAIN\_MAP"

( "CLLI" VARCHAR2(8 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER(10,0),

"STATUS" VARCHAR2(10 BYTE),

"COMMENTS" VARCHAR2(4000 BYTE),

"LAST\_MOD\_BY" VARCHAR2(50 BYTE),

"LAST\_MOD\_TS" VARCHAR2(50 BYTE),

"GI\_LEAF\_DOMAIN\_INST\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CRS\_BANDWIDTH\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CRS\_BANDWIDTH\_WK"

( "DEVICE\_NAME" VARCHAR2(100 BYTE),

"PORT\_NAME" VARCHAR2(100 BYTE),

"BPS\_BANDWIDTH" NUMBER(18,0),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CRS\_DEVICE\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CRS\_DEVICE\_WK"

( "DEVICE\_NAME" VARCHAR2(100 BYTE),

"DEVICE\_IP" VARCHAR2(20 BYTE),

"DEVICE\_MODEL\_NAME" VARCHAR2(100 BYTE),

"MODULE\_MODEL" VARCHAR2(100 BYTE),

"MODULE\_SLOT" VARCHAR2(100 BYTE),

"MODULE\_DESCRIPTION" VARCHAR2(500 BYTE),

"MODULE\_SERIAL\_NUMBER" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CRS\_PORT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CRS\_PORT\_WK"

( "HOST\_NAME" VARCHAR2(100 BYTE),

"DEVICE\_IP" VARCHAR2(20 BYTE),

"PORT\_NAME" VARCHAR2(100 BYTE),

"PORT\_TYPE" VARCHAR2(100 BYTE),

"PORT\_IP" VARCHAR2(20 BYTE),

"PORT\_STATUS" VARCHAR2(100 BYTE),

"RUNNING\_PORT\_STATE" VARCHAR2(100 BYTE),

"DESCRIPTION" VARCHAR2(500 BYTE),

"VLAN\_ID" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CSR\_DEVICE\_AUDIT\_ISSUES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CSR\_DEVICE\_AUDIT\_ISSUES"

( "CSR\_DEVICE\_NAME" VARCHAR2(50 BYTE),

"CSR\_ISSUE\_ID" NUMBER,

"CSR\_VENDOR" VARCHAR2(20 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CSR\_DEVICE\_AUDIT\_ISSUES\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CSR\_DEVICE\_AUDIT\_ISSUES\_WK"

( "CSR\_DEVICE\_NAME" VARCHAR2(50 BYTE),

"CSR\_ISSUE\_ID" NUMBER,

"CSR\_VENDOR" VARCHAR2(10 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CSR\_ISSUE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CSR\_ISSUE"

( "CSR\_ISSUE\_ID" NUMBER(\*,0),

"DESCRIPTION" VARCHAR2(100 BYTE),

"IS\_CRITICAL" CHAR(1 BYTE) DEFAULT 'N',

"COMMENTS" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table CSR\_VLAN\_AUDIT\_ISSUES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."CSR\_VLAN\_AUDIT\_ISSUES"

( "CSR\_DEVICE\_NAME" VARCHAR2(50 BYTE),

"VLAN\_NUMBER" NUMBER,

"CSR\_ISSUE\_ID" NUMBER,

"CSR\_VENDOR" VARCHAR2(20 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table DACS\_QT\_TEST\_HEADS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."DACS\_QT\_TEST\_HEADS"

( "QT\_NAME" VARCHAR2(100 BYTE),

"QT\_DISPLAY\_NAME" VARCHAR2(100 BYTE),

"QT\_IP" VARCHAR2(30 BYTE),

"TAM\_TYPE" VARCHAR2(100 BYTE),

"MSPP\_TAC\_PORT" VARCHAR2(100 BYTE),

"MSPP\_IP" VARCHAR2(30 BYTE),

"MSPP\_TCP\_PORT" VARCHAR2(100 BYTE),

"MSPP\_USER\_ACCOUNT" VARCHAR2(100 BYTE),

"DCS\_TYPE" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table DACS\_TEST\_HEADS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."DACS\_TEST\_HEADS"

( "TEST\_HEAD\_NAME" VARCHAR2(100 BYTE),

"DCS\_TYPE" VARCHAR2(100 BYTE),

"ID" NUMBER,

"TEST\_HEAD" NUMBER,

"T\_ACTIVE" NUMBER,

"NT\_ACTIVE" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table DIR\_INVENTORY\_DEFINITION

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."DIR\_INVENTORY\_DEFINITION"

( "DEFINITION\_ID" NUMBER(10,0),

"INV\_DESCPT\_ID" NUMBER(10,0),

"INV\_DESCPT\_NAME" VARCHAR2(100 BYTE),

"INV\_DEF\_ATTR\_ID" NUMBER(10,0),

"GROUP\_NAME" VARCHAR2(100 BYTE),

"NAME" VARCHAR2(100 BYTE),

"DEFAULT\_VALUE" VARCHAR2(100 BYTE),

"IS\_MANDATORY" VARCHAR2(1 BYTE),

"IS\_MODIFIABLE" VARCHAR2(1 BYTE),

"DISPLAY\_SEQUENCE" NUMBER(10,0),

"GROUP\_DISPLAY\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table DIR\_INVENTORY\_DESCRIPTOR

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."DIR\_INVENTORY\_DESCRIPTOR"

( "INV\_DESCPT\_ID" NUMBER(18,0),

"INV\_DESCPT\_NAME" VARCHAR2(100 BYTE),

"EQUIPMENT\_TYPE" VARCHAR2(100 BYTE),

"ENTITY\_TYPE" VARCHAR2(100 BYTE),

"ENTITY\_NAME" VARCHAR2(100 BYTE),

"FUNCTIONAL\_TYPE" VARCHAR2(30 BYTE),

"STATUS" VARCHAR2(10 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table DIR\_INV\_DEF\_ATTRIBUTE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."DIR\_INV\_DEF\_ATTRIBUTE"

( "INV\_DEF\_ATTR\_ID" NUMBER,

"INV\_DEF\_ATTR\_NAME" VARCHAR2(50 BYTE),

"DISPLAY\_NAME" VARCHAR2(50 BYTE),

"DATA\_TYPE" VARCHAR2(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table DOMAIN\_PARENTS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."DOMAIN\_PARENTS"

( "DOMAIN\_ID" NUMBER(10,0),

"GI\_DOMAIN\_INST\_ID" NUMBER(10,0),

"PARENT\_DOMAIN\_ID" NUMBER(10,0),

"GI\_PARENT\_DOMAIN\_INST\_ID" NUMBER(10,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table EBH\_DISCONNECT\_DETAILS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."EBH\_DISCONNECT\_DETAILS"

( "DOMAIN\_NAME" VARCHAR2(30 BYTE),

"DF\_SEG\_ID" NUMBER(10,0),

"DF\_HAS\_BILL" CHAR(1 BYTE),

"NDF\_SEG\_ID" NUMBER(10,0),

"EBH\_DISCONNECT\_DATE" DATE,

"DF\_IN\_SERVICE\_DATE" DATE,

"DURATION" NUMBER,

"DF\_SITE\_ID" NUMBER(10,0),

"AUDIT\_DATE" DATE,

"EBH\_BILL\_DATE" DATE,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"EBH\_DIS\_ORD\_DATE" DATE,

"EBH\_TERM\_FEE" VARCHAR2(250 BYTE),

"EBH\_Z\_SITE\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table ECP\_SWITCH\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."ECP\_SWITCH\_MAP"

( "SYS\_ID" NUMBER,

"ECP\_SID" NUMBER,

"SWITCH\_ID" VARCHAR2(8 BYTE),

"NG\_DOMAIN\_INST\_ID" NUMBER,

"ECP\_ABBREV" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_ECP" VARCHAR2(50 BYTE),

"ECP\_STATUS" VARCHAR2(50 BYTE) DEFAULT 'Live'

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table EQUIPMENT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."EQUIPMENT"

( "EQP\_REFERENCE\_ID" NUMBER(18,0),

"EQP\_NAME" VARCHAR2(110 BYTE),

"ALTERNATE\_NAME" VARCHAR2(32 BYTE),

"CONTAINER" VARCHAR2(32 BYTE),

"EQP\_MODEL" VARCHAR2(100 BYTE),

"EQP\_TYPE" VARCHAR2(50 BYTE),

"SHELF\_TYPE" VARCHAR2(32 BYTE),

"FUNCTIONAL\_TYPE" VARCHAR2(32 BYTE),

"EQP\_VENDOR" VARCHAR2(32 BYTE),

"EQP\_MANUFACTURER" VARCHAR2(32 BYTE),

"SITE\_REFERENCE\_ID" NUMBER(18,0),

"LOCATION\_CLLI" VARCHAR2(150 BYTE),

"LOGICAL\_SHELF" VARCHAR2(50 BYTE),

"NO\_OF\_SLOTS" NUMBER(18,0),

"IS\_MULTI\_TID\_SHELF" VARCHAR2(32 BYTE),

"PARENT\_EQP\_TYPE" VARCHAR2(32 BYTE),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(18,0),

"PARENT\_SHELF\_REFERENCE\_ID" NUMBER(18,0),

"INEFFECT\_DATE" VARCHAR2(32 BYTE),

"INSTALLED\_DATE" VARCHAR2(32 BYTE),

"DECOMMISSION\_DATE" VARCHAR2(32 BYTE),

"SCHEDULED\_DATE" VARCHAR2(32 BYTE),

"FIELD\_ID" VARCHAR2(16 BYTE),

"POINT\_CODE" VARCHAR2(32 BYTE),

"TID\_LOGICAL" VARCHAR2(120 BYTE),

"TID\_PHYSICAL" VARCHAR2(120 BYTE),

"MGMT\_IP\_ADDRESS" VARCHAR2(32 BYTE),

"IPV4\_ADDRESS" VARCHAR2(32 BYTE),

"IPV6\_ADDRESS" VARCHAR2(32 BYTE),

"LAST\_MODIFIED\_TIME\_STAMP" VARCHAR2(32 BYTE),

"LAST\_MODIFIED\_BY" VARCHAR2(100 BYTE),

"ORDER\_DATE" VARCHAR2(32 BYTE),

"ORDER\_NUMBER" VARCHAR2(40 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE),

"PHYSICAL\_EQP\_REFERENCE\_ID" NUMBER(18,0),

"INV\_STATUS" VARCHAR2(32 BYTE),

"BAR\_CODE" VARCHAR2(70 BYTE),

"SERIAL\_NUMBER" VARCHAR2(70 BYTE),

"BATCH\_NUMBER" VARCHAR2(70 BYTE),

"FRAME" VARCHAR2(70 BYTE),

"LINE\_UP" VARCHAR2(70 BYTE),

"PHYSICAL\_SHELF\_POSITION" VARCHAR2(70 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table EQUIPMENT\_ATTRIBUTES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."EQUIPMENT\_ATTRIBUTES"

( "EQP\_ATTRIBUTES\_ID" NUMBER(18,0),

"EQP\_REFERENCE\_ID" NUMBER(18,0),

"EQP\_DESCPT\_ID" NUMBER(18,0),

"EQP\_GROUP\_NAME" VARCHAR2(50 BYTE),

"EQP\_NAME" VARCHAR2(50 BYTE),

"EQP\_VALUE" VARCHAR2(4000 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table EQUIPMENT\_DOMAIN\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."EQUIPMENT\_DOMAIN\_MAP"

( "EQP\_REFERENCE\_ID" NUMBER(18,0),

"DOMAIN\_ID" NUMBER(10,0),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table ERICSSON\_ENB\_IP\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."ERICSSON\_ENB\_IP\_WK"

( "NODE" VARCHAR2(100 BYTE),

"IP" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table ERICSSON\_ENB\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."ERICSSON\_ENB\_WK"

( "NODE" VARCHAR2(100 BYTE),

"IP" VARCHAR2(50 BYTE),

"CONNECT\_STATUS" VARCHAR2(50 BYTE),

"SYNC\_STATUS" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table ERICSSON\_ENM\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."ERICSSON\_ENM\_WK"

( "MKT\_ID" VARCHAR2(20 BYTE),

"NODE\_NAME" VARCHAR2(50 BYTE),

"IP\_ADDRESS" VARCHAR2(256 BYTE),

"VERSION\_NUMBER" VARCHAR2(50 BYTE),

"CHASSIS\_TYPE" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table ERICSSON\_RRU

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."ERICSSON\_RRU"

( "RRU" VARCHAR2(50 BYTE),

"APN\_PORT" VARCHAR2(100 BYTE),

"SERIAL\_NUMBER" VARCHAR2(100 BYTE),

"CELLS" VARCHAR2(100 BYTE),

"ENODEB" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table GEOPLAN\_DISCOVERY\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."GEOPLAN\_DISCOVERY\_WK"

( "CELL\_NAME" VARCHAR2(200 BYTE),

"CELL\_NUMBER" VARCHAR2(100 BYTE),

"AIR\_INTERFACE" VARCHAR2(100 BYTE),

"SWITCH\_VSM\_CLLI\_CODE" VARCHAR2(100 BYTE),

"SWITCH\_VENDOR" VARCHAR2(200 BYTE),

"STREET\_ADDRESS" VARCHAR2(400 BYTE),

"CITY" VARCHAR2(100 BYTE),

"STATE" VARCHAR2(100 BYTE),

"ZIP\_CODE" VARCHAR2(20 BYTE),

"CELL" VARCHAR2(256 BYTE),

"SWITCH\_NAME" VARCHAR2(200 BYTE),

"COUNTY" VARCHAR2(40 BYTE),

"ELEVATION\_AMSL\_FT" VARCHAR2(40 BYTE),

"LATITUDE\_DEGREES\_NAD83" VARCHAR2(40 BYTE),

"LONGITUDE\_DEGREES\_NAD83" VARCHAR2(40 BYTE),

"STRUCTURE\_TYPE" VARCHAR2(200 BYTE),

"VENDOR" VARCHAR2(100 BYTE),

"PEOPLESOFT\_LOCATION\_CODE" VARCHAR2(100 BYTE),

"IS\_REPEATER" VARCHAR2(10 BYTE),

"LOCATION\_WORKFLOW\_ID" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table GEOPLAN\_SMALLCELL\_DISCOVERY\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."GEOPLAN\_SMALLCELL\_DISCOVERY\_WK"

( "LOCATION\_WORKFLOW\_ID" NUMBER(10,0),

"CELL\_NUMBER" VARCHAR2(100 BYTE),

"CELL\_NAME" VARCHAR2(200 BYTE),

"CELL\_VERSION" VARCHAR2(20 BYTE),

"AIR\_INTERFACE" VARCHAR2(100 BYTE),

"BAND\_CLASS" VARCHAR2(100 BYTE),

"ENODEB" VARCHAR2(100 BYTE),

"ENODEB\_SITENAME" VARCHAR2(100 BYTE),

"ENODEB\_SECTOR" VARCHAR2(20 BYTE),

"VENDOR" VARCHAR2(100 BYTE),

"LTE\_CELLTYPE" VARCHAR2(20 BYTE),

"IS\_SMALLCELL" VARCHAR2(10 BYTE),

"IS\_REPEATER" VARCHAR2(10 BYTE),

"MHZ2100ANTENNAMANUFACTURER" VARCHAR2(100 BYTE),

"MHZ1900ANTENNAMANUFACTURER" VARCHAR2(100 BYTE),

"MHZ1900LTEANTENNAMANUFACTURER" VARCHAR2(100 BYTE),

"MHZ700ANTENNAMANUFACTURER" VARCHAR2(100 BYTE),

"MHZ2100ANTENNAMODEL" VARCHAR2(100 BYTE),

"MHZ1900ANTENNAMODEL" VARCHAR2(100 BYTE),

"MHZ1900LTEANTENNAMODEL" VARCHAR2(100 BYTE),

"MHZ700ANTENNAMODEL" VARCHAR2(100 BYTE),

"SECTOR\_CARRIER\_EQUIP\_TYPE" VARCHAR2(100 BYTE),

"SECTOR\_EQUIP\_TYPE" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table HIGH\_CAPACITY\_PATH\_CHAN

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN"

( "DOMAIN\_ID" NUMBER,

"DOMAIN\_NAME" VARCHAR2(30 BYTE),

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"A\_SIDE\_SITE" VARCHAR2(100 BYTE),

"Z\_SIDE\_SITE" VARCHAR2(100 BYTE),

"PATHPARENT\_NAME" VARCHAR2(100 BYTE),

"PATHPARENT\_ID" NUMBER,

"PATHPARENT\_TYPE" VARCHAR2(30 BYTE),

"PATHPARENT\_STATUS" VARCHAR2(20 BYTE),

"PATHPARENT\_BANDWIDTH" VARCHAR2(30 BYTE),

"PATHPARENT\_NCHL" NUMBER(6,0),

"PATHPARENT\_NCHLA" NUMBER(6,0),

"PATHPARENT\_TOT\_MRC" NUMBER(10,2),

"PATH\_NAME" VARCHAR2(100 BYTE),

"CIRC\_PATH\_INST\_ID" NUMBER(9,0),

"PATH\_TYPE" VARCHAR2(30 BYTE),

"STATUS" VARCHAR2(20 BYTE),

"BANDWIDTH" VARCHAR2(30 BYTE),

"NBR\_CHANNELS" NUMBER(6,0),

"NBR\_CHAN\_ASSIGNED" NUMBER(6,0),

"TOT\_MRC" NUMBER(10,2),

"PATHPARENT\_PCT\_ASGN" NUMBER,

"PATHPARENT\_PCT\_GRP" NUMBER,

"RUN\_DATE" DATE,

"PATH\_TYPE\_CK" VARCHAR2(30 BYTE),

"CHANNEL\_USED" NUMBER(6,0),

"TOTAL\_BPS\_AVAIL" NUMBER(18,0),

"BPS\_ASSIGNED" NUMBER(18,0),

"BPS\_UTIL" NUMBER(5,0),

"BPS\_UTIL\_PREV" NUMBER(5,0),

"BPS\_UTIL\_MOD\_TS" DATE,

"PP\_TRANS\_ACT" VARCHAR2(50 BYTE),

"PP\_EST\_COMP" DATE,

"PP\_ACT\_JUST" VARCHAR2(256 BYTE),

"PP\_ACT\_DATE" DATE,

"PP\_EST\_COMP\_GRP" NUMBER,

"TRANS\_ACT" VARCHAR2(50 BYTE),

"EST\_COMP" DATE,

"ACT\_JUST" VARCHAR2(256 BYTE),

"ACT\_DATE" DATE,

"EST\_COMP\_GRP" NUMBER,

"PP\_ACT\_TOT\_MRC" NUMBER,

"ACT\_TOT\_MRC" NUMBER,

"P\_TOP\_AR" NUMBER,

"P\_TOP\_ARBW" NUMBER,

"EST\_COMP\_DAYS" NUMBER,

"PP\_EST\_COMP\_DAYS" NUMBER,

"PP\_ACT\_JUST\_COMT" VARCHAR2(256 BYTE),

"ACT\_JUST\_COMT" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table HIGH\_CAPACITY\_PATH\_CHAN\_ACT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_ACT"

( "TRAIL\_ID" NUMBER(10,0),

"TRANS\_ACT" VARCHAR2(50 BYTE),

"EST\_COMP" DATE,

"ACT\_JUST" VARCHAR2(256 BYTE),

"ACT\_DATE" DATE,

"EST\_COMP\_GRP" NUMBER,

"ACT\_TOT\_MRC" NUMBER,

"EST\_COMP\_DAYS" NUMBER,

"ACT\_JUST\_COMT" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 52428800 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table HIGH\_CAPACITY\_PATH\_CHAN\_CHILD

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_CHILD"

( "TRAIL\_ID" NUMBER,

"PATH\_NAME" VARCHAR2(100 BYTE),

"PATH\_TYPE" VARCHAR2(30 BYTE),

"STATUS" VARCHAR2(20 BYTE),

"BANDWIDTH" VARCHAR2(30 BYTE),

"NBR\_CHANNELS" NUMBER(6,0),

"NBR\_CHAN\_ASSIGNED" NUMBER,

"TOT\_MRC" NUMBER,

"PATH\_TYPE\_CK" VARCHAR2(30 BYTE),

"PARENT\_PATH\_INST\_ID" NUMBER(10,0),

"CHANNEL\_USED" NUMBER(6,0),

"TRANS\_ACT" VARCHAR2(50 BYTE),

"EST\_COMP" DATE,

"ACT\_JUST" VARCHAR2(256 BYTE),

"ACT\_DATE" DATE,

"EST\_COMP\_GRP" NUMBER,

"ACT\_TOT\_MRC" NUMBER,

"EST\_COMP\_DAYS" NUMBER,

"ACT\_JUST\_COMT" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 83886080 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table HIGH\_CAPACITY\_PATH\_CHAN\_PARENT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_PARENT"

( "DOMAIN\_ID" NUMBER(10,0),

"DOMAIN\_NAME" VARCHAR2(30 BYTE),

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"A\_SITE\_NAME" VARCHAR2(100 BYTE),

"Z\_SITE\_NAME" VARCHAR2(100 BYTE),

"PATHPARENT\_NAME" VARCHAR2(100 BYTE),

"PATHPARENT\_ID" NUMBER(10,0),

"PATHPARENT\_TYPE" VARCHAR2(30 BYTE),

"PATHPARENT\_STATUS" VARCHAR2(20 BYTE),

"PATHPARENT\_BANDWIDTH" VARCHAR2(30 BYTE),

"PATHPARENT\_NCHL" NUMBER(6,0),

"PATHPARENT\_NCHLA" NUMBER,

"P\_TOT\_MRC" NUMBER,

"P\_PCT\_ASGN" NUMBER,

"P\_PCT\_GRP" NUMBER,

"TOTAL\_CAPACITY\_AVL" NUMBER(18,0),

"TOTAL\_CAPACITY\_USED" NUMBER(18,0),

"BPS\_UTIL" NUMBER(5,0),

"P\_TRANS\_ACT" VARCHAR2(50 BYTE),

"P\_EST\_COMP" DATE,

"P\_ACT\_JUST" VARCHAR2(256 BYTE),

"P\_ACT\_DATE" DATE,

"P\_EST\_COMP\_GRP" NUMBER,

"P\_ACT\_TOT\_MRC" NUMBER,

"P\_TOP\_AR" NUMBER,

"P\_TOP\_ARBW" NUMBER,

"P\_EST\_COMP\_DAYS" NUMBER,

"P\_ACT\_JUST\_COMT" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 31457280 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table HIGH\_CAPACITY\_PATH\_CHAN\_UTIL

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_UTIL"

( "PATHPARENT\_ID" NUMBER(10,0),

"TOTAL\_BPS\_AVAIL" NUMBER(18,0),

"BPS\_ASSIGNED" NUMBER(18,0),

"BPS\_UTIL" NUMBER,

"LAST\_MOD\_BY" VARCHAR2(50 BYTE),

"LAST\_MOD\_TS" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 20971520 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table HIGH\_CAPACITY\_PATH\_CHAN\_UTIL\_B

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_UTIL\_B"

( "PATHPARENT\_ID" NUMBER(10,0),

"TOTAL\_CAPACITY\_AVL" NUMBER(18,0),

"TOTAL\_CAPACITY\_USED" NUMBER(18,0),

"BPS\_UTIL" NUMBER,

"LAST\_MOD\_BY" VARCHAR2(50 BYTE),

"LAST\_MOD\_TS" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 20971520 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table HPOV\_CSR\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."HPOV\_CSR\_WK"

( "IP" VARCHAR2(100 BYTE),

"DEVICE\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK"

( "NG\_EQUIP\_INST\_ID" NUMBER,

"NE\_TID" VARCHAR2(30 BYTE),

"AUDIT\_STATUS" VARCHAR2(30 BYTE),

"NG\_CIRC\_PATH\_INST\_ID" NUMBER,

"DEVICE\_XCONN\_ELEMENTS" VARCHAR2(128 BYTE),

"MEMBER\_NBR" NUMBER,

"DISCREPANCY\_DETAILS" VARCHAR2(256 BYTE),

"NG\_LEG\_INST\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK"

( "NG\_SITE\_INST\_ID" NUMBER(9,0),

"NG\_EQUIP\_INST\_ID" NUMBER(9,0),

"TID" VARCHAR2(30 BYTE),

"NG\_DEVICE\_TYPE" VARCHAR2(20 BYTE),

"AUDIT\_STATUS" VARCHAR2(20 BYTE),

"AUDIT\_DATE" DATE,

"MATCHED\_XCONN\_COUNT" NUMBER(9,0),

"DISCREPANCY\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"MATCHED\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"DESCRIPTION" VARCHAR2(1024 BYTE),

"STATUS" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_ADM\_COMP\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_ADM\_COMP\_SUMMARY"

( "AREA" VARCHAR2(30 BYTE),

"REGION" VARCHAR2(128 BYTE),

"INC\_MSPP\_COUNT" NUMBER,

"XCONNECT\_COUNT" NUMBER,

"VSM\_DEVICE\_COUNT" NUMBER,

"INC\_VSM\_MATCH\_COUNT" NUMBER,

"INC\_VSM\_XNG\_MATCH\_COUNT" NUMBER,

"LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"NOT\_LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"MSPP\_REPORTED" NUMBER,

"TOTAL\_FOR\_MSPP\_REPORTED" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK"

( "AREA" VARCHAR2(30 BYTE),

"REGION" VARCHAR2(128 BYTE),

"INC\_MSPP\_COUNT" NUMBER,

"XCONNECT\_COUNT" NUMBER,

"VSM\_DEVICE\_COUNT" NUMBER,

"INC\_VSM\_MATCH\_COUNT" NUMBER,

"INC\_VSM\_XNG\_MATCH\_COUNT" NUMBER,

"LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"NOT\_LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"MSPP\_REPORTED" NUMBER,

"TOTAL\_FOR\_MSPP\_REPORTED" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_COMPLIANCE\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_COMPLIANCE\_SUMMARY"

( "AREA" VARCHAR2(30 BYTE),

"REGION" VARCHAR2(128 BYTE),

"INC\_MSPP\_COUNT" NUMBER,

"XCONNECT\_COUNT" NUMBER,

"VSM\_DEVICE\_COUNT" NUMBER,

"INC\_VSM\_MATCH\_COUNT" NUMBER,

"INC\_VSM\_XNG\_MATCH\_COUNT" NUMBER,

"LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"NOT\_LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"MSPP\_REPORTED" NUMBER,

"TOTAL\_FOR\_MSPP\_REPORTED" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK"

( "AREA" VARCHAR2(30 BYTE),

"REGION" VARCHAR2(128 BYTE),

"INC\_MSPP\_COUNT" NUMBER,

"XCONNECT\_COUNT" NUMBER,

"VSM\_DEVICE\_COUNT" NUMBER,

"INC\_VSM\_MATCH\_COUNT" NUMBER,

"INC\_VSM\_XNG\_MATCH\_COUNT" NUMBER,

"LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"NOT\_LIVE\_XCONNECT\_MATCH\_COUNT" NUMBER,

"MSPP\_REPORTED" NUMBER,

"TOTAL\_FOR\_MSPP\_REPORTED" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_CONNECTION\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_CONNECTION\_WK"

( "CIRCUIT\_ID" VARCHAR2(256 BYTE),

"TID" VARCHAR2(50 BYTE),

"A\_LOGICAL\_PORT" VARCHAR2(50 BYTE),

"Z\_LOGICAL\_PORT" VARCHAR2(50 BYTE),

"SIGNAL\_TYPE" VARCHAR2(20 BYTE),

"CONNECTION\_TIME" DATE,

"WHO" VARCHAR2(25 BYTE),

"ORDER\_NUM" VARCHAR2(20 BYTE),

"USER\_NAME" VARCHAR2(20 BYTE),

"CCT" VARCHAR2(20 BYTE),

"A\_END\_RINGID" VARCHAR2(20 BYTE),

"Z\_END\_RINGID" VARCHAR2(20 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_LOGICAL\_PORT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_LOGICAL\_PORT\_WK"

( "TID" VARCHAR2(50 BYTE),

"RACK\_ADDRESS" VARCHAR2(5 BYTE),

"SUBRACK\_ADDRESS" VARCHAR2(20 BYTE),

"SLOT\_ADDRESS" VARCHAR2(30 BYTE),

"PORT\_ADDRESS" VARCHAR2(50 BYTE),

"LOGICALPORT\_ADDRESS" VARCHAR2(60 BYTE),

"LOGICALPORT\_TYPE" VARCHAR2(25 BYTE),

"CONNECT\_STATUS" VARCHAR2(15 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_NE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_NE"

( "TID" VARCHAR2(50 BYTE),

"NETYPE" VARCHAR2(50 BYTE),

"NESOFTWARERELEASE" VARCHAR2(30 BYTE),

"GNE\_TID" VARCHAR2(30 BYTE),

"DNO\_DATE" DATE,

"NEID" NUMBER,

"EXTRACT\_DATE" DATE,

"IP\_ADDRESS" VARCHAR2(30 BYTE),

"PORT" VARCHAR2(5 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_NE\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_NE\_WK"

( "TID" VARCHAR2(50 BYTE),

"NE\_TYPE" VARCHAR2(50 BYTE),

"NE\_SOFTWARE\_RELEASE" VARCHAR2(30 BYTE),

"GNE\_TID" VARCHAR2(30 BYTE),

"DNO\_DATE" DATE,

"NE\_ID" NUMBER(10,0),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_OPTICAL\_LINK\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_OPTICAL\_LINK\_WK"

( "A\_TID" VARCHAR2(50 BYTE),

"A\_PORT\_ADDRESS" VARCHAR2(50 BYTE),

"Z\_TID" VARCHAR2(50 BYTE),

"Z\_PORT\_ADDRESS" VARCHAR2(50 BYTE),

"SIGNAL\_TYPE" VARCHAR2(20 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_PHYSICAL\_PORT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_PHYSICAL\_PORT\_WK"

( "TID" VARCHAR2(50 BYTE),

"RACK\_ADDRESS" VARCHAR2(5 BYTE),

"SUBRACK\_ADDRESS" VARCHAR2(20 BYTE),

"SLOT\_ADDRESS" VARCHAR2(30 BYTE),

"PORT\_ADDRESS" VARCHAR2(50 BYTE),

"PORT\_TYPE" VARCHAR2(15 BYTE),

"SERVICE\_STATE" VARCHAR2(15 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_PROTECT\_PORT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_PROTECT\_PORT\_WK"

( "TID" VARCHAR2(50 BYTE),

"WORKING\_PORT" VARCHAR2(50 BYTE),

"PROTECT\_PORT" VARCHAR2(50 BYTE),

"PROTECTION\_TYPE" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_RACK\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_RACK\_WK"

( "TID" VARCHAR2(50 BYTE),

"RACK\_TYPE" VARCHAR2(15 BYTE),

"RACK\_POSITION" NUMBER(10,0),

"RACK\_ADDRESS" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_SLOT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_SLOT"

( "TID" VARCHAR2(50 BYTE),

"RACKADDRESS" VARCHAR2(5 BYTE),

"SUBRACKADDRESS" VARCHAR2(15 BYTE),

"SLOTSTATE" VARCHAR2(20 BYTE),

"SLOTADDRESS" VARCHAR2(50 BYTE),

"TYPE" VARCHAR2(100 BYTE),

"APPLICATION\_CODE" VARCHAR2(50 BYTE),

"SSN" VARCHAR2(20 BYTE),

"CLEI" VARCHAR2(20 BYTE),

"ECI" VARCHAR2(30 BYTE),

"SERIALNUMBER" VARCHAR2(25 BYTE),

"VERSION" VARCHAR2(20 BYTE),

"SERVICESTATE" VARCHAR2(20 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_SLOT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_SLOT\_WK"

( "TID" VARCHAR2(50 BYTE),

"RACK\_ADDRESS" VARCHAR2(20 BYTE),

"SUBRACK\_ADDRESS" VARCHAR2(20 BYTE),

"SLOT\_STATE" VARCHAR2(20 BYTE),

"SLOT\_ADDRESS" VARCHAR2(50 BYTE),

"TYPE" VARCHAR2(100 BYTE),

"APPLICATION\_CODE" VARCHAR2(50 BYTE),

"SSN" VARCHAR2(20 BYTE),

"CLEI" VARCHAR2(60 BYTE),

"ECI" VARCHAR2(30 BYTE),

"SERIAL\_NUMBER" VARCHAR2(30 BYTE),

"VERSION" VARCHAR2(30 BYTE),

"SERVICE\_STATE" VARCHAR2(30 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_XCONN\_AUDIT\_DETAILS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_DETAILS"

( "XNG\_EQUIP\_INST\_ID" NUMBER,

"NE\_TID" VARCHAR2(30 BYTE),

"AUDIT\_STATUS" VARCHAR2(30 BYTE),

"XNG\_CIRC\_PATH\_INST\_ID" NUMBER,

"DEVICE\_XCONN\_ELEMENTS" VARCHAR2(128 BYTE),

"MEMBER\_NBR" NUMBER,

"DISCREPANCY\_DETAILS" VARCHAR2(256 BYTE),

"XNG\_LEG\_INST\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_XCONN\_AUDIT\_DETAILS\_T

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_DETAILS\_T"

( "NG\_EQUIP\_INST\_ID" NUMBER,

"NE\_TID" VARCHAR2(30 BYTE),

"AUDIT\_STATUS" VARCHAR2(30 BYTE),

"NG\_CIRC\_PATH\_INST\_ID" NUMBER,

"DEVICE\_XCONN\_ELEMENTS" VARCHAR2(128 BYTE),

"MEMBER\_NBR" NUMBER,

"DISCREPANCY\_DETAILS" VARCHAR2(256 BYTE),

"NG\_LEG\_INST\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table INC\_MSPP\_XCONN\_AUDIT\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_SUMMARY"

( "NG\_SITE\_INST\_ID" NUMBER(9,0),

"NG\_EQUIP\_INST\_ID" NUMBER(9,0),

"TID" VARCHAR2(30 BYTE),

"NG\_DEVICE\_TYPE" VARCHAR2(20 BYTE),

"AUDIT\_STATUS" VARCHAR2(20 BYTE),

"AUDIT\_DATE" DATE,

"MATCHED\_XCONN\_COUNT" NUMBER(9,0),

"DISCREPANCY\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"MATCHED\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"DESCRIPTION" VARCHAR2(1024 BYTE),

"STATUS" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table JUNIPER\_DEVICES\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."JUNIPER\_DEVICES\_WK"

( "HOST\_NAME" VARCHAR2(50 BYTE),

"DEVICE\_IP" VARCHAR2(20 BYTE),

"DEVICE\_ID" NUMBER(10,0),

"DEVICE\_VENDOR" VARCHAR2(50 BYTE),

"DEVICE\_MODEL" VARCHAR2(50 BYTE),

"DEVICE\_PARTITION" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table JUNIPER\_VLANS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."JUNIPER\_VLANS\_WK"

( "HOST\_NAME" VARCHAR2(50 BYTE),

"DEVICE\_IP" VARCHAR2(20 BYTE),

"VLAN\_ID" NUMBER(10,0),

"VLAN\_NAME" VARCHAR2(50 BYTE),

"VLAN\_TYPE" VARCHAR2(50 BYTE),

"VLAN\_PARTITION" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table JV\_CDO\_CLASS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."JV\_CDO\_CLASS"

( "CDO\_CLASS\_PK" NUMBER(38,0),

"QUALIFIED\_NAME" VARCHAR2(200 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table JV\_COMMIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."JV\_COMMIT"

( "COMMIT\_PK" NUMBER(38,0),

"AUTHOR" VARCHAR2(200 CHAR),

"COMMIT\_DATE" DATE,

"COMMIT\_ID" NUMBER(38,2)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table JV\_GLOBAL\_ID

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."JV\_GLOBAL\_ID"

( "GLOBAL\_ID\_PK" NUMBER(38,0),

"LOCAL\_ID" VARCHAR2(200 CHAR),

"FRAGMENT" VARCHAR2(200 CHAR),

"CDO\_CLASS\_FK" NUMBER(38,0),

"OWNER\_ID\_FK" NUMBER(38,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table JV\_SNAPSHOT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."JV\_SNAPSHOT"

( "SNAPSHOT\_PK" NUMBER(4,0),

"TYPE" VARCHAR2(26 BYTE),

"VERSION" NUMBER(3,0),

"STATE" VARCHAR2(4000 BYTE),

"CHANGED\_PROPERTIES" VARCHAR2(1024 BYTE),

"GLOBAL\_ID\_FK" NUMBER(4,0),

"COMMIT\_FK" NUMBER(4,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table LUCENT\_AUDIT\_REG\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."LUCENT\_AUDIT\_REG\_SUMM"

( "T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"EBH\_REPORTED" NUMBER,

"EBH\_CSR\_MATCHED\_LIVE" NUMBER,

"EBH\_CSR\_MATCHED\_NONLIVE" NUMBER,

"EBH\_CSR\_REPORTED" NUMBER,

"EBH\_CSR\_COMP" NUMBER,

"OVERALL\_CSR\_COMP" NUMBER,

"EBH\_COMP" NUMBER,

"OVERALL\_COMP" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table LUCENT\_AUDIT\_SWITCH\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."LUCENT\_AUDIT\_SWITCH\_SUMM"

( "MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"LEAF\_DOMAIN\_NAME" VARCHAR2(100 BYTE),

"SWITCH\_ID" VARCHAR2(8 BYTE),

"T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"EBH\_REPORTED" NUMBER,

"EBH\_CSR\_MATCHED\_LIVE" NUMBER,

"EBH\_CSR\_MATCHED\_NONLIVE" NUMBER,

"EBH\_CSR\_REPORTED" NUMBER,

"EBH\_CSR\_COMP" NUMBER,

"OVERALL\_CSR\_COMP" NUMBER,

"EBH\_COMP" NUMBER,

"OVERALL\_COMP" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table LUCENT\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."LUCENT\_NE\_VS\_NG\_AUDIT"

( "NE\_BTS\_NAME" VARCHAR2(50 BYTE),

"NE\_CELL\_TYPE" VARCHAR2(10 BYTE),

"SYS\_ID" NUMBER,

"ECP\_SID" NUMBER,

"NE\_CELL\_NUM" NUMBER,

"NE\_URC\_NUM" NUMBER,

"NE\_DS1\_NUM" NUMBER,

"NE\_VERSION" VARCHAR2(50 BYTE),

"SWITCH\_ID" VARCHAR2(8 BYTE),

"SERVICE" VARCHAR2(50 BYTE),

"TERMINATION\_TYPE" VARCHAR2(8 BYTE),

"NG\_CELL\_NUM" NUMBER,

"NG\_URC\_NUM" NUMBER,

"NG\_DS1\_NUM" NUMBER,

"TRAIL\_ID" NUMBER,

"PATH\_NAME" VARCHAR2(100 BYTE),

"NG\_PATH\_STATUS" VARCHAR2(20 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(10 BYTE),

"BANDWIDTH" VARCHAR2(30 BYTE),

"MATCH\_CODE" VARCHAR2(8 BYTE),

"MATCH\_STATUS" VARCHAR2(256 BYTE),

"AUDIT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MARKET\_PREFIX\_ENB\_NAMING

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MARKET\_PREFIX\_ENB\_NAMING"

( "MARKET\_ID" VARCHAR2(6 BYTE),

"LTE\_MARKET\_NAME" VARCHAR2(50 BYTE),

"CLLI" VARCHAR2(50 BYTE),

"VENDOR" VARCHAR2(50 BYTE),

"MPEN\_EXTRACT\_DATE" VARCHAR2(20 BYTE) DEFAULT to\_char(sysdate,'yyyymmdd')

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table META\_DATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."META\_DATA"

( "UPDATE\_TYPE" VARCHAR2(50 BYTE),

"SELECTOR\_TYPE" VARCHAR2(100 BYTE),

"SIDE\_CAR\_URL" VARCHAR2(100 BYTE),

"SERVICE" VARCHAR2(50 BYTE),

"OPERATION" VARCHAR2(100 BYTE),

"PATH\_PARAMETER" VARCHAR2(1 BYTE),

"METHOD" VARCHAR2(20 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_BTS\_LOCATION\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_BTS\_LOCATION\_MAP"

( "BTS\_NUMBER" NUMBER,

"BTS\_NAME" VARCHAR2(50 BYTE),

"SANE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE,

"BTS\_TYPE" VARCHAR2(50 BYTE),

"CBSC\_NUMBER" NUMBER,

"LATITUDE" VARCHAR2(20 BYTE),

"LONGITUDE" VARCHAR2(20 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_CDMA\_AUDIT\_DEV\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_CDMA\_AUDIT\_DEV\_SUMM"

( "LEAF\_DOMAIN" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_MTX" VARCHAR2(100 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER(7,2) DEFAULT 0.00,

"EBH\_COMP" NUMBER(7,2) DEFAULT 0.00,

"OVERALL\_COMP" NUMBER(7,2) DEFAULT 0.00,

"EBH\_MATCHED\_NONLIVE\_CSR" NUMBER,

"EBH\_MATCHED\_LIVE\_CSR" NUMBER,

"UBS\_COUNT" NUMBER(\*,0),

"NON\_UBS\_COUNT" NUMBER(\*,0),

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_CDMA\_AUDIT\_REG\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_CDMA\_AUDIT\_REG\_SUMM"

( "T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER(7,2) DEFAULT 0.00,

"EBH\_COMP" NUMBER(7,2) DEFAULT 0.00,

"OVERALL\_COMP" NUMBER(7,2) DEFAULT 0.00,

"T1\_REPORTED" NUMBER,

"EBH\_REPORTED" NUMBER,

"EBH\_MATCHED\_NONLIVE\_CSR" NUMBER,

"EBH\_MATCHED\_LIVE\_CSR" NUMBER,

"UBS\_COUNT" NUMBER,

"NON\_UBS\_COUNT" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_CDMA\_ETHERNET\_INV

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_CDMA\_ETHERNET\_INV"

( "BTS\_NUMBER" NUMBER,

"CLUSTER\_NUMBER" NUMBER,

"ROUTER\_GRP" NUMBER,

"BANDWIDTH" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE,

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_CDMA\_NE\_VS\_NG\_AUDIT"

( "NE\_OMCR" VARCHAR2(50 BYTE),

"NE\_BTS\_NUMBER" NUMBER,

"NE\_CLUSTER\_NUMBER" NUMBER,

"NE\_BTS\_ROUTER\_GROUP" NUMBER,

"NE\_BANDWIDTH" VARCHAR2(50 BYTE),

"NE\_SPAN\_NUMBER" NUMBER,

"NE\_ACCESS\_NODE\_NUMBER" NUMBER,

"NE\_AG\_NODE\_NUMBER" NUMBER,

"NE\_AG\_SPAN\_NUMBER" NUMBER,

"NE\_T1\_LINE\_NUMBER" NUMBER,

"TERMINATION\_TYPE" VARCHAR2(8 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"MTX" VARCHAR2(100 BYTE),

"NG\_PATH\_NAME" VARCHAR2(100 BYTE),

"NG\_PATH\_INST\_ID" NUMBER,

"NG\_BTS\_NUMBER" NUMBER,

"NG\_CLUSTER\_NUMBER" NUMBER,

"NG\_SPAN\_NUMBER" NUMBER,

"NG\_BANDWIDTH" VARCHAR2(50 BYTE),

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"MATCH\_CODE" VARCHAR2(8 BYTE),

"MATCH\_STATUS" VARCHAR2(200 BYTE),

"EXTRACT\_DATE" DATE,

"AUDIT\_DATE" DATE,

"NE\_BTS\_TYPE" VARCHAR2(50 BYTE),

"NE\_BTS\_STATUS" VARCHAR2(20 BYTE),

"NG\_IPBSCDO\_NUMBER" NUMBER,

"NE\_SPAN\_TYPE" VARCHAR2(10 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_CDMA\_NE\_VS\_NG\_AUDIT\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_CDMA\_NE\_VS\_NG\_AUDIT\_WRK"

( "NE\_OMCR" VARCHAR2(50 BYTE),

"NE\_BTS\_NUMBER" NUMBER,

"NE\_CLUSTER\_NUMBER" NUMBER,

"NE\_BTS\_ROUTER\_GROUP" NUMBER,

"NE\_BANDWIDTH" VARCHAR2(50 BYTE),

"NE\_SPAN\_NUMBER" NUMBER,

"NE\_ACCESS\_NODE\_NUMBER" NUMBER,

"NE\_AG\_NODE\_NUMBER" NUMBER,

"NE\_AG\_SPAN\_NUMBER" NUMBER,

"NE\_T1\_LINE\_NUMBER" NUMBER,

"TERMINATION\_TYPE" VARCHAR2(8 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"MTX" VARCHAR2(100 BYTE),

"NG\_PATH\_NAME" VARCHAR2(100 BYTE),

"NG\_PATH\_INST\_ID" NUMBER,

"NG\_BTS\_NUMBER" NUMBER,

"NG\_CLUSTER\_NUMBER" NUMBER,

"NG\_SPAN\_NUMBER" NUMBER,

"NG\_BANDWIDTH" VARCHAR2(50 BYTE),

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"MATCH\_CODE" VARCHAR2(8 BYTE),

"MATCH\_STATUS" VARCHAR2(200 BYTE),

"EXTRACT\_DATE" DATE,

"AUDIT\_DATE" DATE,

"NE\_BTS\_TYPE" VARCHAR2(50 BYTE),

"NE\_BTS\_STATUS" VARCHAR2(20 BYTE),

"NG\_IPBSCDO\_NUMBER" NUMBER,

"NE\_SPAN\_TYPE" VARCHAR2(10 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_CDMA\_T1\_INV

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_CDMA\_T1\_INV"

( "BTS\_NUMBER" NUMBER,

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"CLUSTER\_NUMBER" NUMBER,

"BTS\_ROUTER\_GROUP" NUMBER,

"SPAN\_NUMBER" NUMBER,

"ACCESS\_NODE\_NUMBER" NUMBER,

"AG\_NODE\_NUMBER" NUMBER,

"T1\_LINE\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE,

"AG\_SPAN\_NUMBER" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_EVDO\_AUDIT\_DEV\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_EVDO\_AUDIT\_DEV\_SUMM"

( "LEAF\_DOMAIN" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_AEMS" VARCHAR2(100 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER,

"EBH\_COMP" NUMBER,

"OVERALL\_COMP" NUMBER,

"EBH\_MATCHED\_NONLIVE\_CSR" NUMBER,

"EBH\_MATCHED\_LIVE\_CSR" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_EVDO\_AUDIT\_REG\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_EVDO\_AUDIT\_REG\_SUMM"

( "T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER,

"EBH\_COMP" NUMBER,

"OVERALL\_COMP" NUMBER,

"T1\_REPORTED" NUMBER,

"EBH\_REPORTED" NUMBER,

"EBH\_MATCHED\_NONLIVE\_CSR" NUMBER,

"EBH\_MATCHED\_LIVE\_CSR" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_EVDO\_ETHERNET\_INV

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_EVDO\_ETHERNET\_INV"

( "VSM\_DEVICE\_NAME\_AEMS" VARCHAR2(50 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"MCCDO\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_EVDO\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_EVDO\_NE\_VS\_NG\_AUDIT"

( "AEMS" VARCHAR2(20 BYTE),

"NE\_IPBSCDO\_NUMBER" NUMBER,

"NE\_MCCDO\_IDENTIFIER" NUMBER,

"NE\_SPAN\_NUMBER" NUMBER,

"NE\_STATUS" VARCHAR2(10 BYTE),

"MATCH\_STATUS" VARCHAR2(500 BYTE),

"NG\_PATH\_NAME" VARCHAR2(100 BYTE),

"NG\_PATH\_INST\_ID" NUMBER,

"NG\_PATH\_TYPE" VARCHAR2(20 BYTE),

"NG\_PATH\_STATUS" VARCHAR2(20 BYTE),

"NG\_IPBSCDO\_NUMBER" NUMBER,

"XNG\_MCCDO\_IDENTIFIER" NUMBER,

"NG\_SPAN\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE,

"MATCH\_CODE" VARCHAR2(20 BYTE),

"AUDIT\_DATE" DATE,

"NG\_BANDWIDTH" VARCHAR2(50 BYTE),

"TERMINATION\_TYPE" VARCHAR2(10 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_EVDO\_NE\_VS\_NG\_AUDIT\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_EVDO\_NE\_VS\_NG\_AUDIT\_WRK"

( "AEMS" VARCHAR2(20 BYTE),

"NE\_IPBSCDO\_NUMBER" NUMBER,

"NE\_MCCDO\_IDENTIFIER" NUMBER,

"NE\_SPAN\_NUMBER" NUMBER,

"NE\_STATUS" VARCHAR2(10 BYTE),

"MATCH\_STATUS" VARCHAR2(500 BYTE),

"NG\_PATH\_NAME" VARCHAR2(100 BYTE),

"NG\_PATH\_INST\_ID" NUMBER,

"NG\_PATH\_TYPE" VARCHAR2(20 BYTE),

"NG\_PATH\_STATUS" VARCHAR2(20 BYTE),

"NG\_IPBSCDO\_NUMBER" NUMBER,

"XNG\_MCCDO\_IDENTIFIER" NUMBER,

"NG\_SPAN\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE,

"MATCH\_CODE" VARCHAR2(20 BYTE),

"AUDIT\_DATE" DATE,

"NG\_BANDWIDTH" VARCHAR2(50 BYTE),

"TERMINATION\_TYPE" VARCHAR2(10 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_EVDO\_T1\_INV

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_EVDO\_T1\_INV"

( "VSM\_DEVICE\_NAME\_AEMS" VARCHAR2(50 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"MCCDO\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"STATUS" VARCHAR2(30 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MOTO\_VSM\_NAME\_AEMS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MOTO\_VSM\_NAME\_AEMS"

( "VSM\_DEVICE\_NAME\_AEMS" VARCHAR2(30 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"IPBSCDO\_NAME" VARCHAR2(30 BYTE),

"STATUS" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MSPP\_CIRC\_PATH\_ELEMENTS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MSPP\_CIRC\_PATH\_ELEMENTS"

( "CIRC\_PATH\_INST\_ID" NUMBER(9,0),

"CIRC\_PATH\_HUM\_ID" VARCHAR2(100 BYTE),

"PATH\_STATUS" VARCHAR2(20 BYTE),

"BANDWIDTH" VARCHAR2(30 BYTE),

"LEG\_INST\_ID" NUMBER(9,0),

"MEMBER\_NBR" NUMBER(6,0),

"NEXT\_PATH\_INST\_ID" NUMBER(9,0),

"PREV\_PATH\_INST\_ID" NUMBER(9,0),

"NE\_INST\_ID" NUMBER(9,0),

"EQUIP\_INST\_ID" NUMBER(9,0),

"SITE\_INST\_ID" NUMBER(9,0),

"VENDOR" VARCHAR2(30 BYTE),

"MODEL" VARCHAR2(30 BYTE),

"DESCR" VARCHAR2(100 BYTE),

"NODE\_ID" VARCHAR2(200 BYTE),

"TID" VARCHAR2(30 BYTE),

"PATH\_ELEMENTS" VARCHAR2(4000 BYTE),

"MSPP\_PATH\_ELEMENTS" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MSPP\_CIRC\_PATH\_ELEMENTS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MSPP\_CIRC\_PATH\_ELEMENTS\_WK"

( "CIRC\_PATH\_INST\_ID" NUMBER(9,0),

"CIRC\_PATH\_HUM\_ID" VARCHAR2(100 BYTE),

"PATH\_STATUS" VARCHAR2(20 BYTE),

"BANDWIDTH" VARCHAR2(30 BYTE),

"LEG\_INST\_ID" NUMBER(9,0),

"MEMBER\_NBR" NUMBER(6,0),

"NEXT\_PATH\_INST\_ID" NUMBER(9,0),

"PREV\_PATH\_INST\_ID" NUMBER(9,0),

"NE\_INST\_ID" NUMBER(9,0),

"EQUIP\_INST\_ID" NUMBER(9,0),

"SITE\_INST\_ID" NUMBER(9,0),

"VENDOR" VARCHAR2(30 BYTE),

"MODEL" VARCHAR2(30 BYTE),

"DESCR" VARCHAR2(100 BYTE),

"NODE\_ID" VARCHAR2(200 BYTE),

"TID" VARCHAR2(30 BYTE),

"PATH\_ELEMENTS" VARCHAR2(4000 BYTE),

"MSPP\_PATH\_ELEMENTS" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MSPP\_SUMMARY\_REGION\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MSPP\_SUMMARY\_REGION\_MAP"

( "AREA" VARCHAR2(50 BYTE),

"REGION" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_NAME" VARCHAR2(30 BYTE),

"LEAF\_DOMAIN\_INST\_ID" NUMBER,

"NG\_SITE\_INST\_ID" NUMBER(9,0),

"NG\_EQUIP\_INST\_ID" NUMBER(9,0),

"TID" VARCHAR2(30 BYTE),

"NG\_DEVICE\_TYPE" VARCHAR2(30 BYTE),

"AUDIT\_STATUS" VARCHAR2(20 BYTE),

"AUDIT\_DATE" DATE,

"MATCHED\_XCONN\_COUNT" NUMBER(9,0),

"DISCREPANCY\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"MATCHED\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"DESCRIPTION" VARCHAR2(300 BYTE),

"STATUS" VARCHAR2(30 BYTE),

"CARDTYPE" VARCHAR2(50 BYTE),

"IP\_ADDRESS" VARCHAR2(15 BYTE),

"EQUIP\_STATUS" VARCHAR2(30 BYTE),

"LIVE\_COUNT" NUMBER,

"NOT\_LIVE\_COUNT" NUMBER,

"NAMING\_STANDARD" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MSPP\_SUMMARY\_REGION\_MAP\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MSPP\_SUMMARY\_REGION\_MAP\_WK"

( "AREA" VARCHAR2(50 BYTE),

"REGION" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_NAME" VARCHAR2(30 BYTE),

"LEAF\_DOMAIN\_INST\_ID" NUMBER,

"XNG\_SITE\_INST\_ID" NUMBER(9,0),

"NG\_EQUIP\_INST\_ID" NUMBER(9,0),

"TID" VARCHAR2(30 BYTE),

"NG\_DEVICE\_TYPE" VARCHAR2(30 BYTE),

"AUDIT\_STATUS" VARCHAR2(20 BYTE),

"AUDIT\_DATE" DATE,

"MATCHED\_XCONN\_COUNT" NUMBER(9,0),

"DISCREPANCY\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"MATCHED\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"DESCRIPTION" VARCHAR2(300 BYTE),

"STATUS" VARCHAR2(30 BYTE),

"CARDTYPE" VARCHAR2(50 BYTE),

"IP\_ADDRESS" VARCHAR2(15 BYTE),

"EQUIP\_STATUS" VARCHAR2(30 BYTE),

"LIVE\_COUNT" NUMBER,

"NOT\_LIVE\_COUNT" NUMBER,

"NAMING\_STANDARD" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MSPP\_VSM\_MATCH

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MSPP\_VSM\_MATCH"

( "VSM\_NODE\_ID" VARCHAR2(100 BYTE),

"TID" VARCHAR2(50 BYTE),

"NETYPE" VARCHAR2(50 BYTE),

"CARDTYPE" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_INST\_ID" NUMBER,

"PARENT\_DOMAIN\_INST\_ID" NUMBER,

"AREA" VARCHAR2(30 BYTE),

"REGION" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MSPP\_VSM\_MATCH\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MSPP\_VSM\_MATCH\_WK"

( "VSM\_NODE\_ID" VARCHAR2(100 BYTE),

"TID" VARCHAR2(50 BYTE),

"NETYPE" VARCHAR2(50 BYTE),

"CARDTYPE" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_INST\_ID" NUMBER,

"PARENT\_DOMAIN\_INST\_ID" NUMBER,

"AREA" VARCHAR2(30 BYTE),

"REGION" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_AEMS\_IPBSCDO

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_AEMS\_IPBSCDO"

( "SANE\_NAME\_AEMS" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_AEMS" VARCHAR2(50 BYTE),

"RAW\_STRING" VARCHAR2(50 BYTE),

"IPBSCDO\_NAME" VARCHAR2(50 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"STATUS" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT SYSDATE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_AEMS\_OMCR\_EMH\_IP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_AEMS\_OMCR\_EMH\_IP"

( "SANE\_NAME\_AEMS" VARCHAR2(50 BYTE),

"DEVICE\_ID" VARCHAR2(50 BYTE),

"IP" VARCHAR2(50 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE DEFAULT SYSDATE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_AEMS\_OMCR\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_AEMS\_OMCR\_MAP"

( "SANE\_NAME\_AEMS" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_AEMS" VARCHAR2(50 BYTE),

"RAW\_STRING" VARCHAR2(50 BYTE),

"RAW\_NAME\_OMCR" VARCHAR2(50 BYTE),

"OMCR\_NAME" VARCHAR2(50 BYTE),

"OMCR\_NUMBER" NUMBER,

"STATUS" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT SYSDATE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_EMH\_MCCDO\_BTS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_EMH\_MCCDO\_BTS"

( "IPBSCDO\_NUMBER" NUMBER,

"MCCDO\_NUMBER" NUMBER,

"BTS\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE DEFAULT SYSDATE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_OMCR\_BTS\_CONCONF

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_OMCR\_BTS\_CONCONF"

( "SANE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"BTS\_NUMBER" NUMBER,

"BTS\_MLPPP\_NUMBER" NUMBER,

"BTS\_CON\_NUMBER" NUMBER,

"PBTS\_NUMBER" NUMBER,

"PBTS\_SPAN\_NUMBER" NUMBER,

"ACCESS\_NODE\_NUMBER" NUMBER,

"AG\_NODE\_NUMBER" NUMBER,

"AG\_SPAN\_NUMBER" NUMBER,

"T1\_LINE\_NUMBER" NUMBER,

"PKT\_MCCDO\_CON\_1\_NUMBER" NUMBER,

"PKT\_MCCDO\_CON\_2\_NUMBER" NUMBER,

"PKT\_MCCDO\_CON\_3\_NUMBER" NUMBER,

"PKT\_MCCDO\_CON\_4\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_OMCR\_BTS\_DO\_INFO

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_OMCR\_BTS\_DO\_INFO"

( "SANE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"BTS\_NUMBER" NUMBER,

"IPBSCDO\_NUMBER" NUMBER,

"HD\_MODEM\_DO\_ID" NUMBER,

"BSCDO\_FRAME" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_OMCR\_BTS\_RTR\_GRP\_CONF

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_OMCR\_BTS\_RTR\_GRP\_CONF"

( "SANE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"BTS\_NUMBER" NUMBER,

"BTS\_RTR\_CLUSTER" NUMBER,

"BTS\_RTR\_GRP" NUMBER,

"OTI\_BACKHAUL" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_OMCR\_BTS\_STATUS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_OMCR\_BTS\_STATUS"

( "SANE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"BTS\_NUMBER" NUMBER,

"BTS\_MLPPP\_NUMBER" NUMBER,

"BTS\_CON\_NUMBER" NUMBER,

"PBTS\_NUMBER" NUMBER,

"PBTS\_SPAN\_NUMBER" NUMBER,

"CBSC\_NUMBER" NUMBER,

"TEL\_STATE" VARCHAR2(20 BYTE),

"ADMIN\_STATE" VARCHAR2(20 BYTE),

"OP\_STATE" VARCHAR2(20 BYTE),

"USAGE\_STATE" VARCHAR2(20 BYTE),

"CONTROL\_STATUS" VARCHAR2(120 BYTE),

"AVAIL\_STATUS" VARCHAR2(20 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_OMCR\_IPS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_OMCR\_IPS"

( "SANE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"IP" VARCHAR2(50 BYTE),

"HOSTNAME" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table MT\_VSM\_SANE\_RAW\_OMCR\_NAME

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."MT\_VSM\_SANE\_RAW\_OMCR\_NAME"

( "SANE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"VSM\_DEVICE\_NAME\_OMCR" VARCHAR2(50 BYTE),

"OMCR\_RAW\_NAME" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_CSR\_VLAN\_AUDITS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_AUDITS"

( "CSR\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"CSR\_VENDOR" VARCHAR2(50 BYTE),

"NE\_HOSTNAME" VARCHAR2(100 BYTE),

"DISPLAYED\_NAME" VARCHAR2(50 BYTE),

"NE\_VLAN\_NUMBER" NUMBER,

"DEVICE\_IP" VARCHAR2(50 BYTE),

"MATCH\_CODE" VARCHAR2(50 BYTE),

"MATCH\_STATUS" VARCHAR2(500 BYTE),

"NG\_VLAN\_NUMBER" NUMBER,

"VLAN\_INST\_ID" NUMBER(9,0),

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"AUDIT\_DATE" DATE,

"MARKET" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"CLLI" VARCHAR2(50 BYTE),

"DESCRIPTION" VARCHAR2(250 BYTE),

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_CSR\_VLAN\_CLLI\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_CLLI\_SUMM"

( "MARKET" VARCHAR2(50 BYTE),

"CLLI" VARCHAR2(50 BYTE),

"NE\_TOTAL\_CSR\_VLAN" NUMBER,

"MATCHED\_L\_CSR\_VLAN" NUMBER,

"MATCHED\_NL\_CSR\_VLAN" NUMBER,

"MISMATCHED\_CSR\_VLAN" NUMBER,

"VLAN\_COMPLIANCE" NUMBER,

"NT\_NL\_2\_TOTAL\_VLAN\_PCT" NUMBER,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_CSR\_VLAN\_REGION\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_REGION\_SUMM"

( "NE\_TOTAL\_CSR\_VLAN" NUMBER,

"MATCHED\_L\_CSR\_VLAN" NUMBER,

"MATCHED\_NL\_CSR\_VLAN" NUMBER,

"MISMATCHED\_CSR\_VLAN" NUMBER,

"VLAN\_COMPLIANCE" NUMBER,

"NT\_NL\_2\_TOTAL\_VLAN\_PCT" NUMBER,

"AUDIT\_DATE" DATE,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_CSR\_VLAN\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_WK"

( "HOSTNAME" VARCHAR2(100 BYTE),

"DEVICE\_IP" VARCHAR2(100 BYTE),

"PORT\_NAME" VARCHAR2(100 BYTE),

"PORT\_TYPE" VARCHAR2(100 BYTE),

"PORT\_IP" VARCHAR2(100 BYTE),

"DESCRIPTION" VARCHAR2(250 BYTE),

"PARTITION" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE),

"SUB\_INTERFACE\_NAME" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_CSR\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_CSR\_WK"

( "HOSTNAME" VARCHAR2(100 BYTE),

"DEVICE\_IP" VARCHAR2(100 BYTE),

"DEVICE\_VENDOR" VARCHAR2(100 BYTE),

"DEVICE\_MODEL" VARCHAR2(100 BYTE),

"PARTITION" VARCHAR2(100 BYTE),

"LAST\_SUCCESSFUL\_SNAPSHOT" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT"

( "CSR\_VENDOR" VARCHAR2(20 BYTE),

"CSR\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"NCM\_CSR\_HOSTNAME" VARCHAR2(100 BYTE),

"NCM\_CSR\_PARTITION" VARCHAR2(100 BYTE),

"NCM\_CSR\_DEVICE\_IP" VARCHAR2(100 BYTE),

"NCM\_CSR\_VLAN\_HOSTNAME" VARCHAR2(100 BYTE),

"NCM\_CSR\_VLAN\_PARTITION" VARCHAR2(100 BYTE),

"NCM\_CSR\_VLAN\_DEVICE\_IP" VARCHAR2(100 BYTE),

"HP\_CSR\_HOSTNAME" VARCHAR2(100 BYTE),

"HP\_CSR\_IP" VARCHAR2(40 BYTE),

"NT\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"EQP\_REFERENCE\_ID" NUMBER,

"EQP\_NAME" VARCHAR2(100 BYTE),

"LIVE\_IN\_NG" CHAR(1 BYTE),

"MATCH\_CODE" VARCHAR2(20 BYTE),

"EH\_CSR\_HOSTNAME" VARCHAR2(100 BYTE),

"EH\_CSR\_IPADDRESS" VARCHAR2(100 BYTE),

"BRIX\_CSR\_SITE" VARCHAR2(50 BYTE),

"CELL\_SITE\_NAME" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM"

( "MARKET" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"CLLI\_6" VARCHAR2(6 BYTE),

"NCM\_CSRS" NUMBER,

"NCM\_CSRS\_W\_VLAN" NUMBER,

"HPOV\_CSRS" NUMBER,

"NT\_CSRS" NUMBER,

"NT\_L\_CSRS" NUMBER,

"NT\_NL\_CSRS" NUMBER,

"NCM\_CSRS\_W\_ISSUES" NUMBER,

"NCM\_CSRS\_W\_VLAN\_PER" NUMBER,

"NCM\_HPOV\_MATCH" NUMBER,

"NCM\_HPOV\_ISSUES" NUMBER,

"NCM\_HPOV\_PER" NUMBER,

"NCM\_NT\_MATCH" NUMBER,

"NCM\_NT\_L\_MATCH" NUMBER,

"NCM\_NT\_NL\_MATCH" NUMBER,

"NCM\_NT\_ISSUES" NUMBER,

"NCM\_NT\_PER" NUMBER,

"HPOV\_NT\_MATCH" NUMBER,

"HPOV\_NT\_L\_MATCH" NUMBER,

"HPOV\_NT\_NL\_MATCH" NUMBER,

"HPOV\_NT\_ISSUES" NUMBER,

"HPOV\_NT\_PER" NUMBER,

"AUDIT\_DATE" DATE DEFAULT sysdate,

"EH\_NT\_MATCH" NUMBER,

"EH\_NT\_L\_MATCH" NUMBER,

"EH\_NT\_NL\_MATCH" NUMBER,

"EH\_NT\_ISSUES" NUMBER,

"EH\_NT\_PER" NUMBER,

"EH\_CSRS" NUMBER,

"BRIX\_CSRS" NUMBER DEFAULT 0,

"BRIX\_NT\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_L\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_NL\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_ISSUES" NUMBER DEFAULT 0,

"BRIX\_NT\_PER" NUMBER DEFAULT 0,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_HPOV\_EH\_NT\_CSR\_REGN\_SUM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_HPOV\_EH\_NT\_CSR\_REGN\_SUM"

( "NCM\_CSRS" NUMBER,

"NCM\_CSRS\_W\_VLAN" NUMBER,

"HPOV\_CSRS" NUMBER,

"NT\_CSRS" NUMBER,

"NT\_L\_CSRS" NUMBER,

"NT\_NL\_CSRS" NUMBER,

"NCM\_CSRS\_W\_ISSUES" NUMBER,

"NCM\_CSRS\_W\_VLAN\_PER" NUMBER,

"NCM\_HPOV\_MATCH" NUMBER,

"NCM\_HPOV\_ISSUES" NUMBER,

"NCM\_HPOV\_PER" NUMBER,

"NCM\_NT\_MATCH" NUMBER,

"NCM\_NT\_L\_MATCH" NUMBER,

"NCM\_NT\_NL\_MATCH" NUMBER,

"NCM\_NT\_ISSUES" NUMBER,

"NCM\_NT\_PER" NUMBER,

"HPOV\_NT\_MATCH" NUMBER,

"HPOV\_NT\_L\_MATCH" NUMBER,

"HPOV\_NT\_NL\_MATCH" NUMBER,

"HPOV\_NT\_ISSUES" NUMBER,

"HPOV\_NT\_PER" NUMBER,

"AUDIT\_DATE" DATE,

"EH\_CSRS" NUMBER,

"EH\_NT\_MATCH" NUMBER,

"EH\_NT\_L\_MATCH" NUMBER,

"EH\_NT\_NL\_MATCH" NUMBER,

"EH\_NT\_ISSUES" NUMBER,

"EH\_NT\_PER" NUMBER,

"BRIX\_CSRS" NUMBER DEFAULT 0,

"BRIX\_NT\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_L\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_NL\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_PER" NUMBER DEFAULT 0,

"BRIX\_NT\_ISSUES" NUMBER DEFAULT 0,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_NGMLS\_VLAN\_AUDIT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_NGMLS\_VLAN\_AUDIT\_WK"

( "NGMLS\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"NGMLS\_VENDOR" VARCHAR2(50 BYTE),

"NE\_HOSTNAME" VARCHAR2(100 BYTE),

"NE\_SUB\_INTERFACE\_NAME" VARCHAR2(50 BYTE),

"NE\_VLAN\_NUMBER" NUMBER,

"NE\_PORT\_IP" VARCHAR2(50 BYTE),

"MATCH\_CODE" VARCHAR2(10 BYTE),

"MATCH\_STATUS" VARCHAR2(250 BYTE),

"EQP\_REFERENCE\_ID" NUMBER,

"XNG\_VLAN\_NUMBER" NUMBER,

"VLAN\_REFERENCE\_ID" NUMBER(9,0),

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"AUDIT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_NGMLS\_VLAN\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_NGMLS\_VLAN\_WK"

( "HOSTNAME" VARCHAR2(100 BYTE),

"DEVICE\_IP" VARCHAR2(100 BYTE),

"PORT\_NAME" VARCHAR2(100 BYTE),

"PORT\_TYPE" VARCHAR2(100 BYTE),

"PORT\_IP" VARCHAR2(100 BYTE),

"DESCRIPTION" VARCHAR2(250 BYTE),

"PARTITION" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE),

"SUB\_INTERFACE\_NAME" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NCM\_NGMLS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NCM\_NGMLS\_WK"

( "HOSTNAME" VARCHAR2(100 BYTE),

"DEVICE\_IP" VARCHAR2(100 BYTE),

"DEVICE\_VENDOR" VARCHAR2(100 BYTE),

"DEVICE\_MODEL" VARCHAR2(100 BYTE),

"PARTITION" VARCHAR2(100 BYTE),

"LAST\_SUCCESSFUL\_SNAPSHOT" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_ADM\_EQ\_SUMM\_TMSM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_ADM\_EQ\_SUMM\_TMSM"

( "TERRITORY" VARCHAR2(100 CHAR),

"MARKET\_TERRITORY" VARCHAR2(100 CHAR),

"SUB\_MARKET" VARCHAR2(100 CHAR),

"TOTAL\_NE" NUMBER,

"DUPLICATE\_NE" NUMBER,

"MATCHED\_NE" NUMBER,

"PERCENT\_MATCHED" NUMBER,

"PASSED\_AUDIT" NUMBER,

"NAMING\_COMPLIANCE" NUMBER,

"DISCOVERED\_XCONNECTS" NUMBER,

"LIVE\_XCONNECTS" NUMBER,

"OTHER\_XCONNECTS" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_ADM\_FUJITSU\_EQUIPMENT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_ADM\_FUJITSU\_EQUIPMENT"

( "TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"NE\_LEVEL" NUMBER(9,0),

"NE\_INST\_ID" NUMBER(9,0),

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"SITE\_REFERENCE\_ID" NUMBER(9,0),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(9,0),

"EQ\_CLASS\_TYPE" VARCHAR2(20 CHAR),

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_TYPE" VARCHAR2(50 CHAR),

"EQP\_STATUS" VARCHAR2(50 CHAR),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"CONTAINER" VARCHAR2(20 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP"

( "LEAF\_DOMAIN\_NAME" VARCHAR2(80 CHAR),

"TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"LOCATION" VARCHAR2(80 CHAR),

"CONTACTS" VARCHAR2(80 CHAR),

"IP\_ADDR" VARCHAR2(40 CHAR),

"MATCH\_CODE" CHAR(20 CHAR),

"MATCH\_STATUS" VARCHAR2(500 CHAR),

"NE\_INST\_ID" NUMBER,

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"MATCH8" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_ADM\_NE\_VS\_NG\_AUDIT"

( "LEAF\_DOMAIN\_NAME" VARCHAR2(80 CHAR),

"TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"LOCATION" VARCHAR2(80 CHAR),

"CONTACTS" VARCHAR2(80 CHAR),

"IP\_ADDR" VARCHAR2(40 CHAR),

"MATCH\_CODE" CHAR(20 CHAR),

"MATCH\_STATUS" VARCHAR2(500 CHAR),

"NE\_INST\_ID" NUMBER,

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"MATCH8" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_ADM\_NG\_PORTS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_ADM\_NG\_PORTS"

( "TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"SLOT\_NAME" VARCHAR2(50 CHAR),

"PORT\_NAME" VARCHAR2(50 CHAR),

"AID" VARCHAR2(100 CHAR),

"TRAIL\_ID" NUMBER(9,0),

"COMPONENT\_ID" NUMBER(9,0),

"CABLE\_MEMBER" NUMBER(9,0),

"A\_SITE\_ID" NUMBER(9,0),

"A\_SITE\_NAME" VARCHAR2(100 CHAR),

"Z\_SITE\_ID" NUMBER(9,0),

"Z\_SITE\_NAME" VARCHAR2(100 CHAR),

"TRAIL\_NAME" VARCHAR2(100 CHAR),

"BANDWIDTH" VARCHAR2(30 CHAR),

"PATH\_STATUS" VARCHAR2(20 CHAR),

"NEXT\_TRAIL\_REVISION\_ID" NUMBER(9,0),

"PREV\_PATH\_INST\_ID" NUMBER(9,0),

"NE\_INST\_ID" NUMBER(9,0),

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(9,0),

"SITE\_REFERENCE\_ID" NUMBER(9,0),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_NAME" VARCHAR2(100 CHAR),

"PORT\_REFERENCE\_ID" NUMBER(9,0),

"CARD\_REFERENCE\_ID" NUMBER(9,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_ADM\_XCONNECT\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_ADM\_XCONNECT\_AUDIT"

( "TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"MATCH\_CODE" VARCHAR2(20 CHAR),

"MATCH\_STATUS" VARCHAR2(500 CHAR),

"XCONNECT\_ASIDE" VARCHAR2(80 CHAR),

"A\_EQUIPMENT\_ID" NUMBER(9,0),

"A\_SIDE\_PATH\_INST\_ID" NUMBER(9,0),

"A\_COMPONENT\_ID" NUMBER(9,0),

"A\_CABLE\_MEMBER" NUMBER(9,0),

"A\_PORT\_NAME" VARCHAR2(50 CHAR),

"A\_PORT\_AID" VARCHAR2(100 CHAR),

"XCONNECT\_ZSIDE" VARCHAR2(80 CHAR),

"Z\_EQUIPMENT\_ID" NUMBER(9,0),

"Z\_SIDE\_PATH\_INST\_ID" NUMBER(9,0),

"Z\_COMPONENT\_ID" NUMBER(9,0),

"Z\_CABLE\_MEMBER" NUMBER(9,0),

"Z\_PORT\_NAME" VARCHAR2(50 CHAR),

"Z\_PORT\_ACCESS\_ID" VARCHAR2(100 CHAR),

"TRAIL\_ID" NUMBER(9,0),

"TRAIL\_NAME" VARCHAR2(100 CHAR),

"PATH\_STATUS" VARCHAR2(20 CHAR),

"BANDWIDTH" VARCHAR2(30 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_EQPT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_EQPT"

( "INST\_ID" NUMBER(10,0),

"SNCNE\_ID" NUMBER(10,0),

"COMPONENT\_ID" VARCHAR2(255 CHAR),

"COMPONENT\_TYPE" VARCHAR2(255 CHAR),

"ACTDCM" VARCHAR2(80 CHAR),

"ACTSL" VARCHAR2(80 CHAR),

"ACTTYPE" VARCHAR2(80 CHAR),

"AVAILBW" VARCHAR2(80 CHAR),

"BERSDFIC" VARCHAR2(80 CHAR),

"CACMODE" VARCHAR2(80 CHAR),

"CARDCURRFEED" VARCHAR2(80 CHAR),

"CASCADED" VARCHAR2(80 CHAR),

"CATEGORY" VARCHAR2(80 CHAR),

"CFGMODE" VARCHAR2(80 CHAR),

"CLEI" VARCHAR2(80 CHAR),

"COMPLTYPE" VARCHAR2(80 CHAR),

"CURRENTDRAWNFEED1" VARCHAR2(80 CHAR),

"CURRENTDRAWNFEED2" VARCHAR2(80 CHAR),

"DCCMODE" VARCHAR2(80 CHAR),

"DCMCHK" VARCHAR2(80 CHAR),

"DETWAVE" VARCHAR2(80 CHAR),

"DOM" VARCHAR2(80 CHAR),

"EXPHT2H" VARCHAR2(80 CHAR),

"EXPWAVE" VARCHAR2(80 CHAR),

"EXTOW" VARCHAR2(80 CHAR),

"FFPCONF" VARCHAR2(80 CHAR),

"FICTYPE" VARCHAR2(80 CHAR),

"FREQGRID" VARCHAR2(80 CHAR),

"FREQMAX" VARCHAR2(80 CHAR),

"FREQMIN" VARCHAR2(80 CHAR),

"FUSEFEED1" VARCHAR2(80 CHAR),

"FUSEFEED2" VARCHAR2(80 CHAR),

"GCADD" VARCHAR2(80 CHAR),

"HKIN" VARCHAR2(80 CHAR),

"HLOL" VARCHAR2(80 CHAR),

"LAMBDA" VARCHAR2(80 CHAR),

"LAMBDAINFO" VARCHAR2(80 CHAR),

"LANMODE" VARCHAR2(80 CHAR),

"LANTSLOT" VARCHAR2(80 CHAR),

"LANTYPE" VARCHAR2(80 CHAR),

"LASER" VARCHAR2(80 CHAR),

"LMBDGRID" VARCHAR2(80 CHAR),

"LMBDMAX" VARCHAR2(80 CHAR),

"LMBDMIN" VARCHAR2(80 CHAR),

"LOCATION" VARCHAR2(80 CHAR),

"MAID" VARCHAR2(80 CHAR),

"MANREF" VARCHAR2(80 CHAR),

"MGTSYS" VARCHAR2(80 CHAR),

"NSMODE" VARCHAR2(80 CHAR),

"OOF" VARCHAR2(80 CHAR),

"OWSEL" VARCHAR2(80 CHAR),

"POSTLASER" VARCHAR2(80 CHAR),

"PRELASER" VARCHAR2(80 CHAR),

"RAMAN1" VARCHAR2(80 CHAR),

"RAMAN2" VARCHAR2(80 CHAR),

"REGEN" VARCHAR2(80 CHAR),

"RLOLE1" VARCHAR2(80 CHAR),

"RLOLE2" VARCHAR2(80 CHAR),

"RLOLE3" VARCHAR2(80 CHAR),

"RLOLE4" VARCHAR2(80 CHAR),

"RLOLE5" VARCHAR2(80 CHAR),

"RLOLE6" VARCHAR2(80 CHAR),

"RLOLE7" VARCHAR2(80 CHAR),

"RLOLE8" VARCHAR2(80 CHAR),

"RVFA1" VARCHAR2(80 CHAR),

"RVFA2" VARCHAR2(80 CHAR),

"SCADD" VARCHAR2(80 CHAR),

"SERIALNO" VARCHAR2(80 CHAR),

"SFP1" VARCHAR2(80 CHAR),

"SFP2" VARCHAR2(80 CHAR),

"SFP3" VARCHAR2(80 CHAR),

"SFP4" VARCHAR2(80 CHAR),

"SFP5" VARCHAR2(80 CHAR),

"SFP6" VARCHAR2(80 CHAR),

"SFP7" VARCHAR2(80 CHAR),

"SFP8" VARCHAR2(80 CHAR),

"SHCNFIG1" VARCHAR2(80 CHAR),

"SHCNFIG2" VARCHAR2(80 CHAR),

"SHELFLBL" VARCHAR2(80 CHAR),

"SHFTYPE" VARCHAR2(80 CHAR),

"SMODE" VARCHAR2(80 CHAR),

"SSBITSEL" VARCHAR2(80 CHAR),

"STATUS" VARCHAR2(80 CHAR),

"SWPR" VARCHAR2(80 CHAR),

"SYSREF" VARCHAR2(80 CHAR),

"TCAREF" VARCHAR2(80 CHAR),

"TMGLS" VARCHAR2(80 CHAR),

"TMGOUTLS" VARCHAR2(80 CHAR),

"TRBOW" VARCHAR2(80 CHAR),

"TVFA1" VARCHAR2(80 CHAR),

"TVFA2" VARCHAR2(80 CHAR),

"TYPEINFO01" VARCHAR2(256 CHAR),

"UNITNAME" VARCHAR2(80 CHAR),

"USERLBL" VARCHAR2(80 CHAR),

"USI" VARCHAR2(80 CHAR),

"USRREF" VARCHAR2(80 CHAR),

"VENDID" VARCHAR2(80 CHAR),

"VOLTAGE" VARCHAR2(80 CHAR),

"WDMMODE" VARCHAR2(80 CHAR),

"STATUS1" VARCHAR2(80 CHAR),

"STATUS2" VARCHAR2(80 CHAR),

"LAST\_UPDATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_EQPT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_EQPT\_WK"

( "TID" VARCHAR2(50 BYTE),

"AID" VARCHAR2(50 BYTE),

"TYPE" VARCHAR2(50 BYTE),

"KEYWORD\_DOMAIN" VARCHAR2(1024 BYTE),

"STATUS" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(50 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_EQPT\_WK"."STATUS" IS 'PST (Primary Service State), SST (Secondary Service State), IS-NR (In Service-Normal), OOS-MA (Out of Service - Management), OOS-AUMA (Out of Service - Autonomous Management), OOS-AU (Out of Service - Autonomous), ACT (Active), AINS (Automatic In-Service Provisioning), FLT (Fault), MEA (Mismatch of equipment and attributes), SGEO (Supporting entity outage), STBYH (Standby, hot (Ready to carry traffic)), UAS (Unassigned), UEQ (Unequipped)';

--------------------------------------------------------

-- DDL for Table NETSMART\_EQ\_SUMM\_TMSM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_EQ\_SUMM\_TMSM"

( "TERRITORY" VARCHAR2(100 CHAR),

"MARKET\_TERRITORY" VARCHAR2(100 CHAR),

"SUB\_MARKET" VARCHAR2(100 CHAR),

"TOTAL\_NE" NUMBER,

"DUPLICATE\_NE" NUMBER,

"MATCHED\_NE" NUMBER,

"PERCENT\_MATCHED" NUMBER,

"PASSED\_AUDIT" NUMBER,

"NAMING\_COMPLIANCE" NUMBER,

"DISCOVERED\_XCONNECTS" NUMBER,

"LIVE\_XCONNECTS" NUMBER,

"OTHER\_XCONNECTS" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_FUJITSU\_EQUIPMENT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_FUJITSU\_EQUIPMENT"

( "TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"NE\_LEVEL" NUMBER(9,0),

"NE\_INST\_ID" NUMBER(9,0),

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"SITE\_REFERENCE\_ID" NUMBER(9,0),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(9,0),

"EQ\_CLASS\_TYPE" VARCHAR2(20 CHAR),

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_TYPE" VARCHAR2(50 CHAR),

"INV\_STATUS" VARCHAR2(50 CHAR),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"CONTAINER" VARCHAR2(20 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_IPADDR\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_IPADDR\_WK"

( "TID" VARCHAR2(50 BYTE),

"AID" VARCHAR2(50 BYTE),

"TYPE" VARCHAR2(50 BYTE),

"KEYWORD\_DOMAIN" VARCHAR2(1024 BYTE),

"STATUS" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(50 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_IPADDR\_WK"."AID" IS 'LCN (Local Communication Network), LMP(Local Management Port, NEM(NE Management), LMPL(???)';

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_IPADDR\_WK"."TYPE" IS 'Keyword(Type)=Domain(Value)';

--------------------------------------------------------

-- DDL for Table NETSMART\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_NE\_VS\_NG\_AUDIT"

( "LEAF\_DOMAIN\_NAME" VARCHAR2(80 CHAR),

"TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"LOCATION" VARCHAR2(80 CHAR),

"CONTACTS" VARCHAR2(80 CHAR),

"IP\_ADDR" VARCHAR2(40 CHAR),

"MATCH\_CODE" CHAR(20 CHAR),

"MATCH\_STATUS" VARCHAR2(500 CHAR),

"NE\_INST\_ID" NUMBER,

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"MATCH8" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP"

( "LEAF\_DOMAIN\_NAME" VARCHAR2(80 CHAR),

"TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"LOCATION" VARCHAR2(80 CHAR),

"CONTACTS" VARCHAR2(80 CHAR),

"IP\_ADDR" VARCHAR2(40 CHAR),

"MATCH\_CODE" CHAR(20 CHAR),

"MATCH\_STATUS" VARCHAR2(500 CHAR),

"NE\_INST\_ID" NUMBER,

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"MATCH8" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_NG\_PORTS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_NG\_PORTS"

( "TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"SLOT\_NAME" VARCHAR2(50 CHAR),

"PORT\_NAME" VARCHAR2(50 CHAR),

"AID" VARCHAR2(100 CHAR),

"TRAIL\_ID" NUMBER(9,0),

"COMPONENT\_ID" NUMBER(9,0),

"CABLE\_MEMBER" NUMBER(9,0),

"A\_SITE\_ID" NUMBER(9,0),

"A\_SITE\_NAME" VARCHAR2(100 CHAR),

"Z\_SITE\_ID" NUMBER(9,0),

"Z\_SITE\_NAME" VARCHAR2(100 CHAR),

"TRAIL\_NAME" VARCHAR2(100 CHAR),

"BANDWIDTH" VARCHAR2(30 CHAR),

"PATH\_STATUS" VARCHAR2(20 CHAR),

"NEXT\_PATH\_INST\_ID" NUMBER(9,0),

"PREV\_PATH\_INST\_ID" NUMBER(9,0),

"NE\_INST\_ID" NUMBER(9,0),

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(9,0),

"SITE\_REFERENCE\_ID" NUMBER(9,0),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_NAME" VARCHAR2(100 CHAR),

"PORT\_INST\_ID" NUMBER(9,0),

"CARD\_INST\_ID" NUMBER(9,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_SNCNE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_SNCNE"

( "INST\_ID" NUMBER(9,0),

"TID" VARCHAR2(30 CHAR),

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"TID\_TYPE" VARCHAR2(50 CHAR),

"AUTOMSG" VARCHAR2(20 CHAR),

"USERLOGIN" VARCHAR2(20 CHAR),

"INSERVICE" VARCHAR2(20 CHAR),

"LAST\_UPDATE" DATE,

"LOCATION" VARCHAR2(80 CHAR),

"CONTACTS" VARCHAR2(80 CHAR),

"IP\_ADDR" VARCHAR2(40 CHAR),

"IP\_SUBNET" VARCHAR2(40 CHAR),

"IP\_NETMASK" VARCHAR2(40 CHAR),

"IP\_DEFAULT\_GATEWAY" VARCHAR2(40 CHAR),

"IP\_PORT" VARCHAR2(20 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_SNCNE"."USERLOGIN" IS 'ACT (An uplink session is active for this NE), LOFF (No uplink session exists for this NE)';

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_SNCNE"."INSERVICE" IS 'ACT (The NetSmart has an active connection to this NE), LOFF (The NetSmart does not have an active connection to this NE)';

--------------------------------------------------------

-- DDL for Table NETSMART\_SNCNE\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_SNCNE\_WK"

( "TID" VARCHAR2(50 BYTE),

"TID\_TYPE" VARCHAR2(50 BYTE),

"AUTO\_MSG" VARCHAR2(50 BYTE),

"USER\_LOGIN" VARCHAR2(50 BYTE),

"IN\_SERVICE" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(50 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_SYS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_SYS\_WK"

( "TID" VARCHAR2(50 BYTE),

"AID" VARCHAR2(50 BYTE),

"TYPE" VARCHAR2(50 BYTE),

"KEYWORD\_DOMAIN" VARCHAR2(1024 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(50 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_XCONNECT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_XCONNECT"

( "INST\_ID" NUMBER(10,0),

"SNCNE\_ID" NUMBER(10,0),

"XCONNECT\_ID" VARCHAR2(165 CHAR),

"XCONNECT\_ASIDE" VARCHAR2(80 CHAR),

"XCONNECT\_ZSIDE" VARCHAR2(80 CHAR),

"XCONNECT\_TYPE" VARCHAR2(80 CHAR),

"XCONNECT\_CKTID" VARCHAR2(80 CHAR),

"XCONNECT\_FROMT1" VARCHAR2(80 CHAR),

"XCONNECT\_NUT" VARCHAR2(80 CHAR),

"XCONNECT\_PSWDEF" VARCHAR2(80 CHAR),

"XCONNECT\_RDLNE" VARCHAR2(80 CHAR),

"XCONNECT\_RVRTV" VARCHAR2(80 CHAR),

"XCONNECT\_TAP" VARCHAR2(80 CHAR),

"XCONNECT\_TOT1" VARCHAR2(80 CHAR),

"XCONNECT\_WTR" VARCHAR2(80 CHAR),

"XCONNECT\_SOURCE" VARCHAR2(80 CHAR),

"LAST\_UPDATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_XCONNECT\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_XCONNECT\_AUDIT"

( "TID" VARCHAR2(50 CHAR),

"TID\_TYPE" VARCHAR2(50 CHAR),

"MATCH\_CODE" VARCHAR2(20 CHAR),

"MATCH\_STATUS" VARCHAR2(500 CHAR),

"XCONNECT\_ASIDE" VARCHAR2(80 CHAR),

"A\_EQUIPMENT\_ID" NUMBER(9,0),

"A\_SIDE\_PATH\_INST\_ID" NUMBER(9,0),

"A\_COMPONENT\_ID" NUMBER(9,0),

"A\_CABLE\_MEMBER" NUMBER(9,0),

"A\_PORT\_NAME" VARCHAR2(50 CHAR),

"A\_PORT\_AID" VARCHAR2(100 CHAR),

"XCONNECT\_ZSIDE" VARCHAR2(80 CHAR),

"Z\_EQUIPMENT\_ID" NUMBER(9,0),

"Z\_SIDE\_PATH\_INST\_ID" NUMBER(9,0),

"Z\_COMPONENT\_ID" NUMBER(9,0),

"Z\_CABLE\_MEMBER" NUMBER(9,0),

"Z\_PORT\_NAME" VARCHAR2(50 CHAR),

"Z\_PORT\_ACCESS\_ID" VARCHAR2(100 CHAR),

"TRAIL\_ID" NUMBER(9,0),

"TRAIL\_NAME" VARCHAR2(100 CHAR),

"PATH\_STATUS" VARCHAR2(20 CHAR),

"BANDWIDTH" VARCHAR2(30 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_XCONNECT\_EQUIPMENT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_XCONNECT\_EQUIPMENT"

( "NE\_LEVEL" NUMBER(9,0),

"NE\_INST\_ID" NUMBER(9,0),

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"SITE\_REFERENCE\_ID" NUMBER(9,0),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(9,0),

"EQ\_CLASS\_TYPE" VARCHAR2(20 CHAR),

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(50 CHAR),

"EQP\_TYPE" VARCHAR2(50 CHAR),

"INV\_STATUS" VARCHAR2(50 CHAR),

"EQP\_VENDOR" VARCHAR2(50 CHAR),

"CONTAINER" VARCHAR2(20 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NETSMART\_XCONNECT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NETSMART\_XCONNECT\_WK"

( "TID" VARCHAR2(50 BYTE),

"CMD" VARCHAR2(50 BYTE),

"AID" VARCHAR2(100 BYTE),

"CCT" VARCHAR2(50 BYTE),

"KEYWORD\_DOMAIN" VARCHAR2(1024 BYTE),

"STATUS" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(50 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_XCONNECT\_WK"."AID" IS 'AID (Access Identifier) - From AID, To Aid';

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_XCONNECT\_WK"."CCT" IS 'CCT (Cross Connect Type) Value is either 1-WAY or 2-WAY (Default)';

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_XCONNECT\_WK"."KEYWORD\_DOMAIN" IS 'Type = Value';

COMMENT ON COLUMN "NG\_REPORTS"."NETSMART\_XCONNECT\_WK"."STATUS" IS 'PST (Primary Service State is always <NULL>. SST (Secondary Service State) values are: DEF (Path delector is selecting the default path),SWITCH (PATH selector is switched away from the default path.),<NULL> Secondary state is NULL.)';

--------------------------------------------------------

-- DDL for Table NE\_VS\_NG\_ENB\_AUDIT\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NE\_VS\_NG\_ENB\_AUDIT\_WRK"

( "NE\_MARKET\_ID" VARCHAR2(3 BYTE),

"NODE" VARCHAR2(100 BYTE),

"FROM\_EMS" VARCHAR2(6 BYTE),

"IP" VARCHAR2(50 BYTE),

"CONNECTION\_STATUS" VARCHAR2(50 BYTE),

"SYNCH\_STATUS" VARCHAR2(50 BYTE),

"MATCH\_CODE" VARCHAR2(10 BYTE),

"MATCH\_STATUS" VARCHAR2(500 BYTE),

"NG\_EQUIP\_NAME" VARCHAR2(100 BYTE),

"NG\_MARKET\_ID" VARCHAR2(11 BYTE),

"NG\_ENB\_NUMBER" VARCHAR2(6 BYTE),

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"CONTAINER" CHAR(1 BYTE),

"HAS\_MULTIPLES" VARCHAR2(1 BYTE),

"VENDOR" VARCHAR2(30 BYTE),

"MODEL" VARCHAR2(30 BYTE),

"STATUS" VARCHAR2(20 BYTE),

"NE\_STATUS" VARCHAR2(50 BYTE),

"TYPE" VARCHAR2(30 BYTE),

"COMMENTS" VARCHAR2(256 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NE\_VS\_NT\_SP\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NE\_VS\_NT\_SP\_AUDIT"

( "DATABASE\_NAME" VARCHAR2(50 BYTE),

"SERVER\_IP" VARCHAR2(50 BYTE),

"GROUP\_NAME" VARCHAR2(100 BYTE),

"SITE\_REFERENCE\_ID" NUMBER,

"SITE\_NAME" VARCHAR2(100 BYTE),

"ADDRESS" VARCHAR2(4000 BYTE),

"CITY" VARCHAR2(50 BYTE),

"STATE" VARCHAR2(50 BYTE),

"POST\_CODE" VARCHAR2(20 BYTE),

"SITE\_TYPE" VARCHAR2(40 BYTE),

"MATCH\_CODE" VARCHAR2(50 BYTE),

"MATCH\_STATUS" VARCHAR2(500 BYTE),

"AUDIT\_DATE" VARCHAR2(50 BYTE),

"DUPLICATE" VARCHAR2(50 BYTE),

"DOMAIN\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NE\_VS\_NT\_SP\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NE\_VS\_NT\_SP\_SUMMARY"

( "MATCHED\_NT" NUMBER(6,0),

"NO\_MATCH\_IN\_NT" NUMBER(6,0),

"PER\_MATCHED" NUMBER(5,2) DEFAULT 0,

"TOTAL" NUMBER(6,0) DEFAULT 0,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_DEVICE\_AUDIT\_ISSUES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_DEVICE\_AUDIT\_ISSUES"

( "NGMLS\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"NGMLS\_ISSUE\_ID" VARCHAR2(20 BYTE),

"NGMLS\_VENDOR" VARCHAR2(20 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_EQUIP\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_EQUIP\_AUDIT"

( "NGMLS\_VENDOR" VARCHAR2(20 BYTE),

"NGMLS\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"NE\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"EQP\_NAME" VARCHAR2(100 BYTE),

"MATCH\_CODE" VARCHAR2(20 BYTE),

"INV\_STATUS" VARCHAR2(20 BYTE),

"EQP\_REFERENCE\_ID" NUMBER,

"MARKET" VARCHAR2(100 BYTE),

"TERRITORY" VARCHAR2(100 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"LIVE\_IN\_NG" VARCHAR2(20 BYTE),

"GI\_DEVICE\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_ISSUE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_ISSUE"

( "NGMLS\_ISSUE\_ID" VARCHAR2(20 BYTE),

"DESCRIPTION" VARCHAR2(200 BYTE),

"IS\_CRITICAL" VARCHAR2(20 BYTE),

"COMMENTS" VARCHAR2(200 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_REGION\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_REGION\_SUMM"

( "TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"NCM\_NGMLSS" NUMBER,

"SAM\_NCM\_NGMLSS\_W\_VLAN" NUMBER,

"SAM\_NCM\_NGMLSS\_WO\_VLAN" NUMBER,

"SAM\_NCM\_NGMLSS\_W\_VLAN\_PER" NUMBER,

"NG\_NGMLSS" NUMBER,

"NG\_L\_NGMLSS" NUMBER,

"NG\_NL\_NGMLSS" NUMBER,

"SAM\_NCM\_NG\_MATCH" NUMBER,

"SAM\_NCM\_NG\_L\_MATCH" NUMBER,

"SAM\_NCM\_NG\_NL\_MATCH" NUMBER,

"SAM\_NCM\_NG\_MISMATCH" NUMBER,

"NCM\_GI\_PER" NUMBER,

"AUDIT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_VLAN\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_VLAN\_AUDIT"

( "NGMLS\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"NGMLS\_VENDOR" VARCHAR2(50 BYTE),

"NE\_HOSTNAME" VARCHAR2(100 BYTE),

"NE\_SUB\_INTERFACE\_NAME" VARCHAR2(50 BYTE),

"NE\_VLAN\_NUMBER" NUMBER,

"NE\_PORT\_IP" VARCHAR2(50 BYTE),

"MATCH\_CODE" VARCHAR2(30 BYTE),

"MATCH\_STATUS" VARCHAR2(500 BYTE),

"EQP\_REFERENCE\_ID" NUMBER,

"NG\_VLAN\_NUMBER" NUMBER,

"VLAN\_REFERENCE\_ID" NUMBER(9,0),

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"AUDIT\_DATE" DATE DEFAULT sysdate,

"PORT\_NAME" VARCHAR2(40 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_VLAN\_AUDIT\_ISSUES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_VLAN\_AUDIT\_ISSUES"

( "NGMLS\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER,

"NGMLS\_ISSUE\_ID" NUMBER,

"NGMLS\_VENDOR" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_VLAN\_DEVICE\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_VLAN\_DEVICE\_SUMM"

( "MARKET" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"NGMLS\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"NGMLS\_VENDOR" VARCHAR2(20 BYTE),

"NE\_TOTAL\_NGMLS\_VLAN" NUMBER,

"MATCHED\_L\_NGMLS\_VLAN" NUMBER,

"MATCHED\_NL\_NGMLS\_VLAN" NUMBER,

"MISMATCHED\_NGMLS\_VLAN" NUMBER,

"VLAN\_COMPLIANCE" NUMBER,

"GI\_NL\_2\_TOTAL\_VLAN\_PCT" NUMBER,

"AUDIT\_DATE" DATE DEFAULT sysdate,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGMLS\_VLAN\_REGION\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGMLS\_VLAN\_REGION\_SUMM"

( "NE\_TOTAL\_NGMLS\_VLAN" NUMBER,

"MATCHED\_L\_NGMLS\_VLAN" NUMBER,

"MATCHED\_NL\_NGMLS\_VLAN" NUMBER,

"MISMATCHED\_NGMLS\_VLAN" NUMBER,

"VLAN\_COMPLIANCE" NUMBER,

"GI\_NL\_2\_TOTAL\_VLAN\_PCT" NUMBER,

"AUDIT\_DATE" DATE DEFAULT sysdate,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGT\_QT\_NE\_VS\_NTLS\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGT\_QT\_NE\_VS\_NTLS\_AUDIT"

( "NE\_TEST\_HEAD\_NAME" VARCHAR2(100 BYTE),

"EQ\_TEST\_HEAD\_NAME" VARCHAR2(100 BYTE),

"EQUIP\_INST\_ID" NUMBER,

"EQUIP\_TYPE" VARCHAR2(5 BYTE),

"MATCH\_CODE" CHAR(10 BYTE),

"MATCH\_STATUS" VARCHAR2(100 BYTE),

"CARD\_INST\_ID" NUMBER,

"CARD\_TEST\_HEAD\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGT\_QT\_NE\_VS\_NTLS\_AUDIT\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGT\_QT\_NE\_VS\_NTLS\_AUDIT\_WK"

( "NE\_TEST\_HEAD\_NAME" VARCHAR2(100 BYTE),

"EQ\_TEST\_HEAD\_NAME" VARCHAR2(100 BYTE),

"EQUIP\_INST\_ID" NUMBER,

"EQUIP\_TYPE" VARCHAR2(5 BYTE),

"MATCH\_CODE" CHAR(10 BYTE),

"MATCH\_STATUS" VARCHAR2(100 BYTE),

"CARD\_INST\_ID" NUMBER,

"CARD\_TEST\_HEAD\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGT\_QT\_REG\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGT\_QT\_REG\_SUMM"

( "TOTAL" NUMBER,

"GOOD\_MATCH" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NGT\_QT\_REG\_SUMM\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NGT\_QT\_REG\_SUMM\_WK"

( "TOTAL" NUMBER,

"GOOD\_MATCH" NUMBER,

"TOTAL\_COMPLIANCE" NUMBER,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NG\_ADM\_FW\_EQUIP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NG\_ADM\_FW\_EQUIP"

( "NE\_INST\_ID" NUMBER,

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"SITE\_REFERENCE\_ID" NUMBER(9,0),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(9,0),

"EQ\_CLASS\_TYPE" VARCHAR2(5 CHAR),

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(30 CHAR),

"EQP\_TYPE" VARCHAR2(30 CHAR),

"INV\_STATUS" VARCHAR2(20 CHAR),

"EQP\_VENDOR" VARCHAR2(40 CHAR),

"CONTAINER" VARCHAR2(10 BYTE),

"PARSE\_STATUS" VARCHAR2(250 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NG\_EQUIPMENT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NG\_EQUIPMENT"

( "EQP\_REFERENCE\_ID" NUMBER,

"EQP\_NAME" VARCHAR2(110 BYTE),

"ALTERNATE\_NAME" VARCHAR2(32 BYTE),

"CONTAINER" VARCHAR2(32 BYTE),

"EQP\_MODEL" VARCHAR2(100 BYTE),

"EQP\_TYPE" VARCHAR2(50 BYTE),

"SHELF\_TYPE" VARCHAR2(32 BYTE),

"FUNCTIONAL\_TYPE" VARCHAR2(32 BYTE),

"EQP\_VENDOR" VARCHAR2(32 BYTE),

"EQP\_MANUFACTURER" VARCHAR2(32 BYTE),

"PART\_NUM" VARCHAR2(32 BYTE),

"SERIAL\_NUMBER" VARCHAR2(32 BYTE),

"MATERIAL\_ID" VARCHAR2(32 BYTE),

"EQUIPMENT\_SPEC\_REF\_ID" NUMBER,

"EQUIPMENT\_SPEC\_NAME" VARCHAR2(32 BYTE),

"INV\_STATUS" VARCHAR2(32 BYTE),

"EQP\_CLEI" VARCHAR2(32 BYTE),

"SITE\_REFERENCE\_ID" NUMBER,

"LOCATION\_CLLI" VARCHAR2(32 BYTE),

"HI\_LOCATION\_TYPE" NUMBER,

"HI\_LOCATION\_REFERENCE" VARCHAR2(32 BYTE),

"LOW\_LOCATION\_TYPE" NUMBER,

"LOW\_LOCATION\_REFERENCE" VARCHAR2(32 BYTE),

"LINE\_UP" VARCHAR2(32 BYTE),

"FRAME" VARCHAR2(32 BYTE),

"PHYSICAL\_SHELF\_POSITION" VARCHAR2(32 BYTE),

"LOGICAL\_SHELF" VARCHAR2(32 BYTE),

"NO\_OF\_SLOTS" NUMBER,

"WIRE\_CENTER" VARCHAR2(32 BYTE),

"IS\_MULTI\_TID\_SHELF" VARCHAR2(32 BYTE),

"PARENT\_EQP\_TYPE" VARCHAR2(32 BYTE),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER,

"PARENT\_SHELF\_REFERENCE\_ID" NUMBER,

"BAR\_CODE" VARCHAR2(32 BYTE),

"BATCH\_NUMBER" VARCHAR2(32 BYTE),

"COMMENTS" VARCHAR2(800 BYTE),

"CUSTOMER\_REFERENCE\_ID" NUMBER,

"EQP\_HEIGHT" NUMBER,

"EQP\_WIDTH" NUMBER,

"EQP\_DEPTH" NUMBER,

"DIST\_FROM\_Z" NUMBER,

"DIST\_FROM\_Y" NUMBER,

"DIST\_FROM\_X" NUMBER,

"DUE\_DATE" VARCHAR2(32 BYTE),

"INEFFECT\_DATE" VARCHAR2(32 BYTE),

"INSTALLED\_DATE" VARCHAR2(32 BYTE),

"DECOMMISION\_DATE" VARCHAR2(32 BYTE),

"SCHEDULED\_DATE" VARCHAR2(32 BYTE),

"FIELD\_ID" VARCHAR2(32 BYTE),

"POINT\_CODE" VARCHAR2(32 BYTE),

"TID\_LOGICAL" VARCHAR2(100 BYTE),

"TID\_PHYSICAL" VARCHAR2(100 BYTE),

"HARDWARE\_REVISION" VARCHAR2(32 BYTE),

"SOFTWARE\_REVISION" VARCHAR2(32 BYTE),

"EMS" VARCHAR2(32 BYTE),

"MGMT\_IP\_ADDRESS" VARCHAR2(32 BYTE),

"AID" VARCHAR2(32 BYTE),

"IPV4\_ADDRESS" VARCHAR2(32 BYTE),

"IPV6\_ADDRESS" VARCHAR2(32 BYTE),

"LAST\_MODIFIED\_TIME\_STAMP" VARCHAR2(32 BYTE),

"LAST\_MODIFIED\_BY" VARCHAR2(100 BYTE),

"ORDER\_DATE" VARCHAR2(32 BYTE),

"ORDER\_NUMBER" VARCHAR2(32 BYTE),

"ASSET\_LIFE" NUMBER,

"EQP\_PURCHASE\_DATE" VARCHAR2(32 BYTE),

"EQP\_PURCHASE\_PRICE" VARCHAR2(32 BYTE),

"EQP\_SOURCE" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE),

"ASSET\_OWNER" VARCHAR2(32 BYTE),

"SUB\_LOCATION" VARCHAR2(32 BYTE),

"CUSTOMER\_NAME" VARCHAR2(32 BYTE),

"NETWORK\_TYPE" VARCHAR2(32 BYTE),

"NETWORK\_DOMAIN" VARCHAR2(32 BYTE),

"PROJECT\_REFERENCE\_ID" VARCHAR2(32 BYTE),

"INSTANCE\_TYPE" VARCHAR2(32 BYTE),

"PHYSICAL\_EQ\_REFERENCE\_ID" NUMBER,

"MGMT\_TELNET\_PORT" VARCHAR2(32 BYTE),

"MGMT\_SSH\_PORT" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NG\_FW\_EQUIP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NG\_FW\_EQUIP"

( "NE\_INST\_ID" NUMBER,

"EQP\_REFERENCE\_ID" NUMBER(9,0),

"SITE\_REFERENCE\_ID" NUMBER(9,0),

"PARENT\_EQP\_REFERENCE\_ID" NUMBER(9,0),

"EQ\_CLASS\_TYPE" VARCHAR2(5 CHAR),

"EQP\_NAME" VARCHAR2(100 CHAR),

"EQP\_MODEL" VARCHAR2(30 CHAR),

"EQP\_TYPE" VARCHAR2(30 CHAR),

"INV\_STATUS" VARCHAR2(20 CHAR),

"EQP\_VENDOR" VARCHAR2(40 CHAR),

"CONTAINER" NVARCHAR2(32),

"PARSE\_STATUS" VARCHAR2(250 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NG\_NGMLS\_VLAN\_PATHS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NG\_NGMLS\_VLAN\_PATHS\_WK"

( "EQP\_REFERENCE\_ID" NUMBER,

"CARD\_REFERENCE\_ID" NUMBER,

"CKT\_PATH\_REFERENCE\_ID" VARCHAR2(50 BYTE),

"PORT\_BANDWIDTH" VARCHAR2(50 BYTE),

"PATH\_TYPE" VARCHAR2(50 BYTE),

"VLAN\_REFERENCE\_ID" NUMBER,

"VLAN\_NAME" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER,

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NIS\_DB\_CONNECTIONS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NIS\_DB\_CONNECTIONS"

( "ID" NUMBER(\*,0),

"CONNECTION\_NAME" VARCHAR2(50 BYTE),

"DB\_HOSTNAME" VARCHAR2(50 BYTE),

"DB\_NAME" VARCHAR2(50 BYTE),

"DB\_PORT" NUMBER(\*,0),

"DB\_USER\_NAME" VARCHAR2(50 BYTE),

"DB\_PASSWORD" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NIS\_EXTRACTOR

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NIS\_EXTRACTOR"

( "RULE\_ID" NUMBER(9,0),

"RULE\_NAME" VARCHAR2(500 BYTE),

"RULE\_TYPE" VARCHAR2(15 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE),

"DESCRIPTION" VARCHAR2(4000 BYTE),

"RULE\_DEFINITION" CLOB,

"FILE\_PATH" VARCHAR2(4000 BYTE),

"FILENAME" VARCHAR2(4000 BYTE),

"FILE\_TYPE" VARCHAR2(15 BYTE),

"ORDER\_IND" CHAR(1 BYTE),

"COMPRESS\_IND" CHAR(1 BYTE),

"REMOTE\_HOST" VARCHAR2(500 BYTE),

"REMOTE\_USER" VARCHAR2(50 BYTE),

"REMOTE\_PATH" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS"

LOB ("RULE\_DEFINITION") STORE AS SECUREFILE (

TABLESPACE "TS\_NGREPORTS" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE LOGGING NOCOMPRESS KEEP\_DUPLICATES

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)) ;

--------------------------------------------------------

-- DDL for Table NIS\_EXTRACTOR\_CONTROL

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL"

( "PARTITION\_ID" NUMBER(2,0),

"CONTROL\_ID" NUMBER(15,0),

"RULE\_ID" NUMBER(9,0),

"NODE\_ID" NUMBER(9,0),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE),

"PARENT\_CONTROL\_ID" NUMBER(15,0),

"SUBMIT\_DATE" DATE,

"REQUESTED\_EXEC\_DATE" DATE,

"RUN\_DATE" DATE,

"RUN\_STATUS" VARCHAR2(50 BYTE),

"EXEC\_HOST" VARCHAR2(250 BYTE),

"FILE\_PATH" VARCHAR2(4000 BYTE),

"FILENAME" VARCHAR2(4000 BYTE),

"EXEC\_MESSAGE" CLOB,

"EXEC\_END\_TIME" DATE,

"ELAPSED\_MS" NUMBER(18,0)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS"

LOB ("EXEC\_MESSAGE") STORE AS SECUREFILE (

TABLESPACE "TS\_NGREPORTS" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE LOGGING NOCOMPRESS KEEP\_DUPLICATES ) ;

--------------------------------------------------------

-- DDL for Table NIS\_EXTRACTOR\_GRAPH

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH"

( "NODE\_ID" NUMBER(9,0),

"RULE\_ID" NUMBER(9,0),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE),

"PARENT\_NODE\_ID" NUMBER(9,0),

"IS\_RECUR" CHAR(1 BYTE),

"RUN\_FREQ" CHAR(1 BYTE),

"RUN\_HOUR" NUMBER(2,0),

"RUN\_MIN" NUMBER(2,0),

"RUN\_DAY\_OF\_MONTH" NUMBER(2,0),

"RUN\_MONTH" NUMBER(2,0),

"RUN\_DAY\_OF\_WEEK" NUMBER(2,0)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NIS\_EXTRACTOR\_PARAM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM"

( "RULE\_ID" NUMBER(9,0),

"PARAM\_INST\_ID" NUMBER(9,0),

"PARAM\_ID" NUMBER(9,0),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE),

"PARAM\_TYPE" VARCHAR2(100 BYTE),

"PARAM\_NAME" VARCHAR2(500 BYTE),

"PARAM\_VALUE" VARCHAR2(4000 BYTE)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NOCC\_LAST\_TEST\_DETAILS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_DETAILS"

( "DOMAIN\_ID" NUMBER(9,0),

"DOMAIN\_NAME" VARCHAR2(50 BYTE),

"SITE\_INST\_ID" NUMBER(9,0),

"SITE\_NAME" VARCHAR2(100 BYTE),

"SITE\_ID" VARCHAR2(30 BYTE),

"SITE\_TECH\_NAME" VARCHAR2(3100 BYTE),

"INVENTORIED\_CELLS" VARCHAR2(3100 BYTE),

"SITE\_TYPE" VARCHAR2(30 BYTE),

"NOCC\_MONITORED" VARCHAR2(30 BYTE),

"VZW\_OWNED" VARCHAR2(30 BYTE),

"LIGHTS\_REQD" VARCHAR2(30 BYTE),

"LIGHT\_LAST\_TESTED" VARCHAR2(30 BYTE),

"LIGHT\_NEXT\_TEST" VARCHAR2(30 BYTE),

"LIGHT\_TEST\_OVERDUE" VARCHAR2(30 BYTE),

"ENV\_LAST\_TESTED" VARCHAR2(30 BYTE),

"ENV\_NEXT\_TEST" VARCHAR2(30 BYTE),

"ENV\_TEST\_OVERDUE" VARCHAR2(30 BYTE),

"RUN\_DATE" DATE,

"EQUIP\_INST\_ID" NUMBER,

"LIGHTS\_VOLUNTARY" VARCHAR2(30 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NOCC\_LAST\_TEST\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY"

( "TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(100 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"DOMAIN\_ID" VARCHAR2(80 BYTE),

"DOMAIN\_NAME" VARCHAR2(100 BYTE),

"SITES" NUMBER(9,0),

"ANTENNAS" NUMBER(9,0),

"VZW\_OWNED" NUMBER(9,0),

"LIGHTS\_REQD" NUMBER(9,0),

"LIGHTS\_TESTED" NUMBER(9,0),

"LIGHTS\_TEST\_OVERDUE" NUMBER(9,0),

"LIGHTS\_TEST\_OK" NUMBER(9,0),

"NOCC\_MONITORED\_SITES" NUMBER(9,0),

"ENV\_TESTED" NUMBER(9,0),

"ENV\_TEST\_OVERDUE" NUMBER(9,0),

"ENV\_TEST\_OK" NUMBER(9,0),

"RUN\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NORTEL\_CDMA\_AUDIT\_DEV\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NORTEL\_CDMA\_AUDIT\_DEV\_SUMM"

( "LEAF\_DOMAIN" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_MTX" VARCHAR2(100 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER DEFAULT 0.00,

"EBH\_COMP" NUMBER DEFAULT 0.00,

"OVERALL\_COMP" NUMBER DEFAULT 0.00,

"EBH\_CSR\_MATCHED\_LIVE" NUMBER,

"EBH\_CSR\_MATCHED\_NON\_LIVE" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NORTEL\_CDMA\_AUDIT\_REG\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NORTEL\_CDMA\_AUDIT\_REG\_SUMM"

( "T1\_DISCOVERED" NUMBER,

"T1\_MATCHED\_LIVE" NUMBER,

"T1\_MATCHED\_NONLIVE" NUMBER,

"EBH\_DISCOVERED" NUMBER,

"EBH\_MATCHED\_LIVE" NUMBER,

"EBH\_MATCHED\_NONLIVE" NUMBER,

"T1\_COMP" NUMBER,

"EBH\_COMP" NUMBER,

"OVERALL\_COMP" NUMBER,

"T1\_REPORTED" NUMBER,

"EBH\_REPORTED" NUMBER,

"EBH\_CSR\_MATCHED\_LIVE" NUMBER,

"EBH\_CSR\_MATCHED\_NON\_LIVE" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NORTEL\_DOM\_INVENTORY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NORTEL\_DOM\_INVENTORY"

( "SEQ\_NUM" NUMBER,

"EMS\_NAME" VARCHAR2(100 BYTE),

"DOM\_IP" VARCHAR2(100 BYTE),

"BSM\_RAW\_NAME" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONG\_NAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"SLOT\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE,

"SANE\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NORTEL\_DOM\_INVENTORY\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NORTEL\_DOM\_INVENTORY\_WRK"

( "SEQ\_NUM" NUMBER,

"EMS\_NAME" VARCHAR2(100 BYTE),

"DOM\_IP" VARCHAR2(100 BYTE),

"BSM\_RAW\_NAME" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"DCG\_LONG\_NAME" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"SLOT\_NUMBER" NUMBER,

"EXTRACT\_DATE" DATE,

"SANE\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NORTEL\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT"

( "VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_EMS" VARCHAR2(100 BYTE),

"VSM\_DEVICE\_NAME\_MTX" VARCHAR2(100 BYTE),

"NE\_DCG\_NUMBER" NUMBER,

"NE\_SLOT\_NUMBER" NUMBER,

"NE\_SPAN\_NUMBER" NUMBER,

"NE\_DOM\_IP" VARCHAR2(100 BYTE),

"NE\_STATUS" VARCHAR2(50 BYTE),

"TERMINATION\_TYPE" VARCHAR2(8 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"TRAIL\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"NG\_DCG\_NUMBER" NUMBER,

"NG\_SLOT\_NUMBER" NUMBER,

"NG\_SPAN\_NUMBER" NUMBER,

"NG\_DOM\_IP" VARCHAR2(100 BYTE),

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"TRAIL\_TYPE" VARCHAR2(100 BYTE),

"MATCH\_CODE" VARCHAR2(8 BYTE),

"MATCH\_STATUS" VARCHAR2(1000 BYTE),

"NE\_SPAN\_TYPE" CHAR(10 BYTE),

"EXTRACT\_DATE" DATE,

"AUDIT\_DATE" DATE,

"NG\_BANDWIDTH" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NORTEL\_PORT\_INVENTORY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NORTEL\_PORT\_INVENTORY"

( "DOM\_SEQ\_NUM" NUMBER,

"PORT\_TYPE" VARCHAR2(100 BYTE),

"PORT\_NUM" NUMBER,

"STATUS" VARCHAR2(50 BYTE),

"PRIMARY\_IP" VARCHAR2(100 BYTE),

"PRIMARY\_MASK" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NORTEL\_PORT\_INVENTORY\_WRK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NORTEL\_PORT\_INVENTORY\_WRK"

( "DOM\_SEQ\_NUM" NUMBER,

"PORT\_TYPE" VARCHAR2(100 BYTE),

"PORT\_NUM" NUMBER,

"STATUS" VARCHAR2(50 BYTE),

"PRIMARY\_IP" VARCHAR2(100 BYTE),

"PRIMARY\_MASK" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_CSR\_GIGE\_DELIVERY\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_CSR\_GIGE\_DELIVERY\_WK"

( "TRAIL\_ID" NUMBER(11,0),

"SEGMENT\_ID" NUMBER(10,0),

"SEG\_ORDER" NUMBER(6,0),

"DELIVERED\_ON" VARCHAR2(90 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_CSR\_GIGE\_PARENTS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_CSR\_GIGE\_PARENTS\_WK"

( "GIGE\_TRAIL\_ID" NUMBER(9,0),

"PARENT\_TRAIL\_ID" NUMBER(9,0),

"TRAIL\_COMPONENT\_ID" VARCHAR2(300 BYTE),

"TRAIL\_COMPONENT\_NAME" VARCHAR2(150 BYTE),

"SYS\_CONN\_BY\_TRAIL" VARCHAR2(1000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_CSR\_GIGE\_TRAILS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_CSR\_GIGE\_TRAILS\_WK"

( "EQUIPMENT\_ID" NUMBER(9,0),

"PORT\_ID" NUMBER(10,0),

"TRAIL\_ID" NUMBER(11,0),

"NEXT\_REVISION\_TRAIL\_ID" NUMBER(9,0),

"CSR\_LIVE\_IN\_XNG" CHAR(1 BYTE),

"GIGE\_LIVE\_IN\_XNG" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_CSR\_PARSED\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_CSR\_PARSED\_WK"

( "EQP\_REFERENCE\_ID" NUMBER,

"EQP\_NAME" VARCHAR2(100 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"CSR\_DEVICE\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_CSR\_VLAN\_PATHS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS"

( "EQP\_REFERENCE\_ID" NUMBER(9,0),

"TRAIL\_ID" NUMBER(9,0),

"PORT\_BANDWIDTH" VARCHAR2(30 BYTE),

"PATH\_TYPE" VARCHAR2(30 BYTE),

"VLAN\_INST\_ID" NUMBER(9,0),

"VLAN\_NAME" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER,

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate,

"CARD\_REFERENCE\_ID" NUMBER,

"CARD\_TYPE" VARCHAR2(50 BYTE),

"TEST\_HEAD\_NAME" VARCHAR2(50 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_CSR\_VLAN\_PATHS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS\_WK"

( "EQP\_REFERENCE\_ID" NUMBER(9,0),

"TRAIL\_ID" NUMBER(9,0),

"PORT\_BANDWIDTH" VARCHAR2(30 BYTE),

"PATH\_TYPE" VARCHAR2(30 BYTE),

"VLAN\_INST\_ID" NUMBER(9,0),

"VLAN\_NAME" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER,

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate,

"CARD\_REFERENCE\_ID" NUMBER,

"CARD\_TYPE" VARCHAR2(50 BYTE),

"TEST\_HEAD\_NAME" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_NGMLS\_VLAN\_PATHS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_NGMLS\_VLAN\_PATHS\_WK"

( "EQP\_REFERENCE\_ID" NUMBER,

"CARD\_REFERENCE\_ID" NUMBER,

"CKT\_PATH\_REFERENCE\_ID" VARCHAR2(50 BYTE),

"PORT\_BANDWIDTH" VARCHAR2(50 BYTE),

"PATH\_TYPE" VARCHAR2(50 BYTE),

"VLAN\_REFERENCE\_ID" NUMBER,

"VLAN\_NAME" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER,

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NTLS\_UPDATER

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_UPDATER"

( "UPDATE\_TYPE" VARCHAR2(40 BYTE),

"KEYS" VARCHAR2(100 BYTE),

"UDA\_GROUP" VARCHAR2(40 BYTE),

"DESCRIPTION" VARCHAR2(100 BYTE),

"AUDIT\_TYPE" VARCHAR2(20 BYTE),

"INSERT\_DATE" DATE,

"IS\_ACTIVE" VARCHAR2(20 BYTE),

"SELECTOR\_TYPE" VARCHAR2(20 BYTE),

"EMAIL\_ADDRESS" VARCHAR2(50 BYTE),

"ID" VARCHAR2(20 BYTE),

"SQL" CLOB

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS"

LOB ("SQL") STORE AS SECUREFILE (

TABLESPACE "TS\_NGREPORTS" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE LOGGING NOCOMPRESS KEEP\_DUPLICATES

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)) ;

--------------------------------------------------------

-- DDL for Table NTLS\_UPDATER\_ISSUES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NTLS\_UPDATER\_ISSUES"

( "ERROR\_CODE" NUMBER,

"ERROR\_MESSAGE" VARCHAR2(1000 BYTE),

"REQUEST\_URI" CLOB,

"UPDATED\_ON" DATE,

"REQUEST\_JSON" CLOB

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS"

LOB ("REQUEST\_URI") STORE AS SECUREFILE (

TABLESPACE "TS\_NGREPORTS" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE LOGGING NOCOMPRESS KEEP\_DUPLICATES

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT))

LOB ("REQUEST\_JSON") STORE AS SECUREFILE (

TABLESPACE "TS\_NGREPORTS" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE LOGGING NOCOMPRESS KEEP\_DUPLICATES

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)) ;

--------------------------------------------------------

-- DDL for Table NT\_EVDO\_BTS\_TERM\_EMS\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NT\_EVDO\_BTS\_TERM\_EMS\_SUMM"

( "LEAF\_DOMAIN\_ID" NUMBER,

"VSM\_DEVICE\_NAME\_EMS" VARCHAR2(50 BYTE),

"TOT\_UP\_T1\_SPANS" NUMBER,

"TOT\_UP\_T1\_SPANS\_W\_L\_PATHS" NUMBER,

"TOT\_UP\_T1\_SPANS\_W\_NL\_PATHS" NUMBER,

"TOT\_UP\_EBH\_SPANS" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_L\_PATHS" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_NL\_PATHS" NUMBER,

"T1\_COMP" NUMBER,

"EBH\_COMP" NUMBER,

"OVERALL\_COMP" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_L\_PATHSCSR" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_NL\_PATHSCSR" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NT\_EVDO\_BTS\_TERM\_REG\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NT\_EVDO\_BTS\_TERM\_REG\_SUMM"

( "TOT\_UP\_T1\_SPANS" NUMBER,

"TOT\_UP\_T1\_SPANS\_W\_L\_PATHS" NUMBER,

"TOT\_UP\_T1\_SPANS\_W\_NL\_PATHS" NUMBER,

"TOT\_UP\_EBH\_SPANS" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_L\_PATHS" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_NL\_PATHS" NUMBER,

"T1\_COMP" NUMBER,

"EBH\_COMP" NUMBER,

"OVERALL\_COMP" NUMBER,

"T1\_REPORTED" NUMBER,

"EBH\_REPORTED" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_L\_PATHSCSR" NUMBER,

"TOT\_UP\_EBH\_SPANS\_W\_NL\_PATHSCSR" NUMBER,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NT\_SITERRA\_ANT\_EXTRACT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT"

( "SITE\_INST\_ID" NUMBER(9,0),

"SITE\_DESCRIPTION" VARCHAR2(150 BYTE),

"EQUIP\_INST\_ID" NUMBER(9,0),

"ANTENNA\_ID" VARCHAR2(250 BYTE),

"NAURU\_ID" VARCHAR2(25 BYTE),

"TOWER\_OWNER" VARCHAR2(4000 BYTE),

"TOWER\_OWNER\_CONTACT\_INFO" VARCHAR2(4000 BYTE),

"NEAREST\_AIRPORT\_DIRECTION" VARCHAR2(4000 BYTE),

"NEAREST\_AIRPORT\_DISTANCE" VARCHAR2(4000 BYTE),

"NEAREST\_AIRPORT\_NAME" VARCHAR2(4000 BYTE),

"NEAREST\_AIRPORT\_CONTACT\_PHONE" VARCHAR2(4000 BYTE),

"ANTENNA\_SUPPORT\_FINISH" VARCHAR2(4000 BYTE),

"ANTENNA\_SUPPORT\_MAINTAINED\_BY" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NT\_SITERRA\_SEG\_EXTRACT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT"

( "SITE\_INST\_ID" NUMBER(9,0),

"SITE\_DESCRIPTION" VARCHAR2(150 BYTE),

"CIRC\_INST\_ID" NUMBER(9,0),

"NAURU\_ID" VARCHAR2(25 BYTE),

"CIRCUIT\_ID" VARCHAR2(150 BYTE),

"SEGMENT\_TYPE" VARCHAR2(50 BYTE),

"SEGMENT\_BANDWIDTH" VARCHAR2(50 BYTE),

"SEGMENT\_BPS" NUMBER(18,0),

"TELCO\_PROVIDER" VARCHAR2(4000 BYTE),

"TELCO\_VENDOR\_COMMENTS" VARCHAR2(4000 BYTE),

"TELCO\_VENDOR\_TECHNICAL\_CONTACT" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table NT\_SITERRA\_SITE\_EXTRACT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT"

( "SITE\_INST\_ID" NUMBER(9,0),

"SITE\_DESCRIPTION" VARCHAR2(150 BYTE),

"NAURU\_ID" VARCHAR2(25 BYTE),

"ACCESS\_RESTRICTIONS" VARCHAR2(4000 BYTE),

"MSC\_CSR\_PORT" VARCHAR2(4000 BYTE),

"CSR\_MAC\_ADDRESS" VARCHAR2(4000 BYTE),

"DRIVE\_DIRECTIONS" VARCHAR2(4000 BYTE),

"LOCAL\_FIRE\_PHONE" VARCHAR2(4000 BYTE),

"LOCAL\_FIRE\_PHONE\_COMMENTS" VARCHAR2(4000 BYTE),

"LOCAL\_POLICE\_PHONE" VARCHAR2(4000 BYTE),

"LOCAL\_POLICE\_PHONE\_COMMENTS" VARCHAR2(4000 BYTE),

"ENV\_ALARM\_TEST\_DATE" VARCHAR2(4000 BYTE),

"HVAC\_CONTACT\_PHONE" VARCHAR2(4000 BYTE),

"HVAC\_MAINT\_VENDOR" VARCHAR2(4000 BYTE),

"POWER\_ACCOUNT\_NUMBER" VARCHAR2(4000 BYTE),

"POWER\_PHONE" VARCHAR2(4000 BYTE),

"POWER\_COMPANY" VARCHAR2(4000 BYTE),

"POWER\_METER\_NUM" VARCHAR2(4000 BYTE),

"BUILDING\_TYPE" VARCHAR2(4000 BYTE),

"FIRE\_SUPRESSION\_TYPE" VARCHAR2(4000 BYTE),

"XNG\_PATH\_HYPERLINK" VARCHAR2(4000 BYTE),

"EXTERNAL\_TELCO\_NIU\_FLAG" VARCHAR2(4000 BYTE),

"GAS\_COMPANY" VARCHAR2(4000 BYTE),

"GAS\_COMPANY\_ACCT\_NUM" VARCHAR2(4000 BYTE),

"GAS\_COMPANY\_METER\_NUM" VARCHAR2(4000 BYTE),

"GAS\_COMPANY\_PHONE" VARCHAR2(4000 BYTE),

"HIGH\_VOLTAGE\_PROTECTION" VARCHAR2(4000 BYTE),

"POTS\_COMPANY" VARCHAR2(4000 BYTE),

"POSTS\_COMPANY\_ACCT\_NUM" VARCHAR2(4000 BYTE),

"POTS\_PHONE\_NUM" VARCHAR2(4000 BYTE),

"POTS\_COMPANY\_PHONE\_NUMBER" VARCHAR2(4000 BYTE),

"ENODEB\_IPV4" VARCHAR2(4000 BYTE),

"ENODEB\_IPV6" VARCHAR2(4000 BYTE),

"COPPER\_DEMARC\_LOCATION" VARCHAR2(4000 BYTE),

"FIBER\_DEMARC\_LOCATION" VARCHAR2(4000 BYTE),

"MICROWAVE\_DEMARC\_LOCATION" VARCHAR2(4000 BYTE),

"EBH\_DEMARC\_LOCATION" VARCHAR2(4000 BYTE),

"T1\_DEMARC\_LOCATION" VARCHAR2(4000 BYTE),

"XNG\_COMMENTS" VARCHAR2(4000 BYTE),

"CSR\_DEVICE\_IP" VARCHAR2(4000 BYTE),

"CSR\_SYSTEM\_IP" VARCHAR2(4000 BYTE),

"CSR\_MGMT\_IP" VARCHAR2(4000 BYTE),

"CSR\_LTE\_LOOPBACK\_IPV4" VARCHAR2(4000 BYTE),

"CSR\_LTE\_LOOPBACK\_IPV6" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table PORT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."PORT"

( "PORT\_REFERENCE\_ID" NUMBER(38,0),

"PORT\_NAME" VARCHAR2(100 CHAR),

"PORT\_NUMBER" NUMBER(38,0),

"DISPATCH\_PORT\_NAME" VARCHAR2(32 CHAR),

"DESCRIPTION" VARCHAR2(100 BYTE),

"VIRTUAL\_PORT" VARCHAR2(32 CHAR),

"TP\_TYPE" VARCHAR2(32 CHAR),

"PORT\_TYPE" VARCHAR2(32 CHAR),

"FUNCTIONAL\_TYPE" VARCHAR2(32 CHAR),

"BANDWIDTH\_NAME" VARCHAR2(32 CHAR),

"BANDWIDTH\_REFRENCE\_ID" NUMBER(38,0),

"DIRECTION" VARCHAR2(32 CHAR),

"PORT\_ROLE" VARCHAR2(32 CHAR),

"PORT\_STATUS" VARCHAR2(32 CHAR),

"PORT\_SPEC\_REFERENCE\_ID" NUMBER(38,0),

"AID" VARCHAR2(100 CHAR),

"ALIAS\_AID" VARCHAR2(100 CHAR),

"SITE\_REFERENCE\_ID" NUMBER(38,0),

"EQP\_REFERENCE\_ID" NUMBER(38,0),

"CARD\_REFERENCE\_ID" NUMBER(38,0),

"SLOT\_REFERENCE\_ID" VARCHAR2(32 CHAR),

"PARENT\_PORT\_REF\_ID" NUMBER(38,0),

"TID\_LOGICAL" VARCHAR2(150 BYTE),

"TID\_PHYSICAL" VARCHAR2(100 CHAR),

"DEVICE\_IP" VARCHAR2(32 CHAR),

"RELATION" VARCHAR2(32 CHAR),

"RELATED\_PORT\_REF\_ID" NUMBER(38,0),

"RELATED\_SPEC\_REF\_ID" NUMBER(38,0),

"CONNECTOR\_TYPE" VARCHAR2(32 CHAR),

"A\_WIRED\_PORT\_REF\_ID" NUMBER(38,0),

"Z\_WIRED\_PORT\_REF\_ID" NUMBER(38,0),

"CABLE\_NAME" VARCHAR2(32 CHAR),

"CABLE\_STRAND" VARCHAR2(32 CHAR),

"CABLE\_REFERENCE\_ID" NUMBER(38,0),

"CKT\_NAME" VARCHAR2(100 CHAR),

"CKT\_REFRENCE\_ID" NUMBER(38,0),

"CKT\_PATH\_REFERENCE\_ID" NUMBER(38,0),

"WAVELENGTH" VARCHAR2(32 CHAR),

"DATE\_PROVISIONED" VARCHAR2(100 CHAR),

"NEXT\_CHANNEL\_REFERENCE\_ID" NUMBER(38,0),

"NEXT\_PATH\_REFERENCE\_ID" NUMBER(38,0),

"PARENT\_PORT\_CHANNEL" NUMBER(38,0),

"PARENT\_PORT\_CHANNEL\_NAME" VARCHAR2(32 CHAR),

"PATH\_CHANNEL\_REF\_ID" NUMBER(38,0),

"PORT\_RESVATION\_REF\_ID" NUMBER(38,0),

"LAST\_MODIFIED\_BY" VARCHAR2(100 CHAR),

"LAST\_MODIFIED\_TIME\_STAMP" VARCHAR2(100 CHAR),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 CHAR),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 CHAR),

"BLOCK\_REASON\_CODE" VARCHAR2(20 CHAR),

"DIST\_LENG\_FP" NUMBER(38,2),

"DIST\_BASE\_FP" NUMBER(38,2),

"PORT\_WIDTH" NUMBER(38,2),

"PORT\_HEIGHT" NUMBER(38,2),

"PEER\_BANDWIDTH" VARCHAR2(100 CHAR),

"BLOCK\_KEY\_NAME" VARCHAR2(60 CHAR),

"BLOCK\_KEY\_VALUE" VARCHAR2(100 CHAR),

"PHYSICAL\_PORT\_REFERENCE\_ID" NUMBER(38,0),

"INSTANCE\_TYPE" VARCHAR2(50 CHAR),

"ADMIN\_STATUS" VARCHAR2(80 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table PORT\_ATTRIBUTES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."PORT\_ATTRIBUTES"

( "PORT\_ATTRIBUTES\_ID" NUMBER(18,0),

"PORT\_REFERENCE\_ID" NUMBER(18,0),

"CARD\_REFERENCE\_ID" NUMBER(18,0),

"PORT\_DESCPT\_ID" NUMBER(18,0),

"PORT\_GROUP\_NAME" VARCHAR2(50 BYTE),

"PORT\_NAME" VARCHAR2(50 BYTE),

"PORT\_VALUE" VARCHAR2(4000 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(4000 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table PORT\_OLD

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."PORT\_OLD"

( "PORT\_REFERENCE\_ID" NUMBER(38,0),

"PORT\_NAME" VARCHAR2(100 CHAR),

"PORT\_NUMBER" NUMBER(38,0),

"DISPATCH\_PORT\_NAME" VARCHAR2(32 CHAR),

"DESCRIPTION" VARCHAR2(32 CHAR),

"VIRTUAL\_PORT" VARCHAR2(32 CHAR),

"TP\_TYPE" VARCHAR2(32 CHAR),

"PORT\_TYPE" VARCHAR2(32 CHAR),

"FUNCTIONAL\_TYPE" VARCHAR2(32 CHAR),

"BANDWIDTH\_NAME" VARCHAR2(32 CHAR),

"BANDWIDTH\_REFRENCE\_ID" NUMBER(38,0),

"DIRECTION" VARCHAR2(32 CHAR),

"PORT\_ROLE" VARCHAR2(32 CHAR),

"PORT\_STATUS" VARCHAR2(32 CHAR),

"PORT\_SPEC\_REFERENCE\_ID" NUMBER(38,0),

"AID" VARCHAR2(100 CHAR),

"ALIAS\_AID" VARCHAR2(100 CHAR),

"SITE\_REFERENCE\_ID" NUMBER(38,0),

"EQP\_REFERENCE\_ID" NUMBER(38,0),

"CARD\_REFERENCE\_ID" NUMBER(38,0),

"SLOT\_REFERENCE\_ID" VARCHAR2(32 CHAR),

"PARENT\_PORT\_REF\_ID" NUMBER(38,0),

"TID\_LOGICAL" VARCHAR2(100 CHAR),

"TID\_PHYSICAL" VARCHAR2(100 CHAR),

"DEVICE\_IP" VARCHAR2(32 CHAR),

"RELATION" VARCHAR2(32 CHAR),

"RELATED\_PORT\_REF\_ID" NUMBER(38,0),

"RELATED\_SPEC\_REF\_ID" NUMBER(38,0),

"CONNECTOR\_TYPE" VARCHAR2(32 CHAR),

"A\_WIRED\_PORT\_REF\_ID" NUMBER(38,0),

"Z\_WIRED\_PORT\_REF\_ID" NUMBER(38,0),

"CABLE\_NAME" VARCHAR2(32 CHAR),

"CABLE\_STRAND" VARCHAR2(32 CHAR),

"CABLE\_REFERENCE\_ID" NUMBER(38,0),

"CKT\_NAME" VARCHAR2(100 CHAR),

"CKT\_REFRENCE\_ID" NUMBER(38,0),

"CKT\_PATH\_REFERENCE\_ID" NUMBER(38,0),

"WAVELENGTH" VARCHAR2(32 CHAR),

"DATE\_PROVISIONED" DATE,

"NEXT\_CHANNEL\_REFERENCE\_ID" NUMBER(38,0),

"NEXT\_PATH\_REFERENCE\_ID" NUMBER(38,0),

"PARENT\_PORT\_CHANNEL" NUMBER(38,0),

"PARENT\_PORT\_CHANNEL\_NAME" VARCHAR2(32 CHAR),

"PATH\_CHANNEL\_REF\_ID" NUMBER(38,0),

"PORT\_RESVATION\_REF\_ID" NUMBER(38,0),

"LAST\_MODIFIED\_BY" VARCHAR2(100 CHAR),

"LAST\_MODIFIED\_TIME\_STAMP" DATE,

"FR\_REF\_KEY\_NAME" VARCHAR2(32 CHAR),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 CHAR),

"BLOCK\_REASON\_CODE" VARCHAR2(20 CHAR),

"DIST\_LENG\_FP" NUMBER(38,2),

"DIST\_BASE\_FP" NUMBER(38,2),

"PORT\_WIDTH" NUMBER(38,2),

"PORT\_HEIGHT" NUMBER(38,2),

"PEER\_BANDWIDTH" VARCHAR2(100 CHAR),

"BLOCK\_KEY\_NAME" VARCHAR2(60 CHAR),

"BLOCK\_KEY\_VALUE" VARCHAR2(100 CHAR),

"PHYSICAL\_PORT\_REFERENCE\_ID" NUMBER(38,0),

"INSTANCE\_TYPE" VARCHAR2(50 CHAR),

"ADMIN\_STATUS" VARCHAR2(80 CHAR)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table PROSPECT\_SERVERS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."PROSPECT\_SERVERS"

( "SERVER" VARCHAR2(50 BYTE),

"IP\_ADDRESS" VARCHAR2(50 BYTE),

"ENABLED\_YN" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table PROS\_ECP\_CELL\_DS1\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."PROS\_ECP\_CELL\_DS1\_MAP"

( "SYS\_ID" NUMBER,

"ECP\_SID" NUMBER,

"CELL\_NUMBER" NUMBER,

"URC\_NUM" NUMBER,

"DS1\_NUM" NUMBER,

"BTS\_NAME" VARCHAR2(50 BYTE),

"CELL\_TYPE" VARCHAR2(10 BYTE),

"VERSION" VARCHAR2(50 BYTE),

"CELL\_STAT" VARCHAR2(10 BYTE),

"DS1\_STAT" VARCHAR2(10 BYTE),

"SERVER" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE,

"SERVICE" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table PROS\_ECP\_CELL\_EBH\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."PROS\_ECP\_CELL\_EBH\_MAP"

( "SYS\_ID" NUMBER,

"ECP\_SID" NUMBER,

"CELL\_NUMBER" NUMBER,

"URC\_NUM" NUMBER,

"BTS\_NAME" VARCHAR2(50 BYTE),

"CELL\_TYPE" VARCHAR2(10 BYTE),

"VERSION" VARCHAR2(50 BYTE),

"CELL\_STAT" VARCHAR2(10 BYTE),

"URC\_STAT" VARCHAR2(10 BYTE),

"SERVER" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE,

"SERVICE" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table RRH\_CPRI

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."RRH\_CPRI"

( "REQUEST\_ID" NUMBER(10,0),

"REQUEST\_TIME" DATE,

"SITE\_ID" VARCHAR2(100 BYTE),

"CPRI\_ID" VARCHAR2(50 BYTE),

"CPRI\_FRU\_ID" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table RRH\_LTE\_CELL

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."RRH\_LTE\_CELL"

( "SITE\_ID" VARCHAR2(100 BYTE),

"LTE\_CELL\_ID" VARCHAR2(100 BYTE),

"ANT\_PORT\_ID" VARCHAR2(500 BYTE),

"ANT\_FULL\_NAME" VARCHAR2(500 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table RRH\_RFM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."RRH\_RFM"

( "SITE\_ID" VARCHAR2(100 BYTE),

"RFM\_ID" VARCHAR2(100 BYTE),

"HW\_NUMBER" VARCHAR2(100 BYTE),

"RFM\_DISPLAY\_NAME" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_CSR\_VLAN\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_CSR\_VLAN\_AUDIT"

( "CSR\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"CSR\_VENDOR" VARCHAR2(50 BYTE),

"NE\_HOSTNAME" VARCHAR2(100 BYTE),

"DISPLAYED\_NAME" VARCHAR2(50 BYTE),

"NE\_VLAN\_NUMBER" NUMBER,

"NODE\_ID" VARCHAR2(50 BYTE),

"MATCH\_CODE" VARCHAR2(18 BYTE),

"MATCH\_STATUS" VARCHAR2(250 BYTE),

"NG\_VLAN\_NUMBER" NUMBER,

"VLAN\_INST\_ID" NUMBER(9,0),

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"AUDIT\_DATE" DATE DEFAULT sysdate,

"CLLI" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"PORT\_NAME" VARCHAR2(50 BYTE),

"TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_CSR\_VLAN\_CLLI\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_CSR\_VLAN\_CLLI\_SUMM"

( "MARKET" VARCHAR2(50 BYTE),

"CLLI" VARCHAR2(50 BYTE),

"NE\_TOTAL\_CSR\_VLAN" NUMBER,

"MATCHED\_L\_CSR\_VLAN" NUMBER,

"MATCHED\_NL\_CSR\_VLAN" NUMBER,

"MISMATCHED\_CSR\_VLAN" NUMBER,

"VLAN\_COMPLIANCE" NUMBER,

"NT\_NL\_2\_TOTAL\_VLAN\_PCT" NUMBER,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_CSR\_VLAN\_REGION\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_CSR\_VLAN\_REGION\_SUMM"

( "NE\_TOTAL\_CSR\_VLAN" NUMBER,

"MATCHED\_L\_CSR\_VLAN" NUMBER,

"MATCHED\_NL\_CSR\_VLAN" NUMBER,

"MISMATCHED\_CSR\_VLAN" NUMBER,

"VLAN\_COMPLIANCE" NUMBER,

"GI\_NL\_2\_TOTAL\_VLAN\_PCT" NUMBER,

"AUDIT\_DATE" DATE DEFAULT sysdate,

"TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_EBH\_HEALTH

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_EBH\_HEALTH"

( "MAC\_ADDRESS" VARCHAR2(100 BYTE),

"OUT\_OF\_BAND\_ADDR" VARCHAR2(100 BYTE),

"LAST\_RESYN\_START" VARCHAR2(100 BYTE),

"LAST\_RESYN\_END" VARCHAR2(100 BYTE),

"RESYNC\_STATUS" VARCHAR2(100 BYTE),

"RESYNC\_STATE" VARCHAR2(50 BYTE),

"REACHABILITY" VARCHAR2(50 BYTE),

"CHASSIS\_TYPE" VARCHAR2(100 BYTE),

"SITE\_NAME" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_EBH\_PORT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_EBH\_PORT"

( "CARD\_TYPE\_BIT" VARCHAR2(200 BYTE),

"EBH\_PORT\_MODE" VARCHAR2(100 BYTE),

"MAC\_ADDRESS" VARCHAR2(100 BYTE),

"ENCAP\_TYPE" VARCHAR2(50 BYTE),

"PORT\_NAME" VARCHAR2(100 BYTE),

"SITE\_ID" VARCHAR2(100 BYTE),

"SITE\_NAME" VARCHAR2(100 BYTE),

"OPR\_STATE" VARCHAR2(20 BYTE),

"ADM\_STATE" VARCHAR2(20 BYTE),

"CFM\_LB\_MODE" VARCHAR2(20 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_EBH\_RTR

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_EBH\_RTR"

( "PHYSICAL\_ADDRESS" VARCHAR2(50 BYTE),

"IP" VARCHAR2(100 BYTE),

"OUT\_ENCAP\_VALUE" VARCHAR2(20 BYTE),

"PORT\_NAME" VARCHAR2(100 BYTE),

"ENCAP\_TYPE" VARCHAR2(20 BYTE),

"NODE\_NAME" VARCHAR2(50 BYTE),

"NODE\_ID" VARCHAR2(100 BYTE),

"DISPLAYED\_NAME" VARCHAR2(50 BYTE),

"DESCRIPTION" VARCHAR2(100 BYTE),

"EXTRACT\_DATE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_EH\_NT\_CSR\_CLLI\_SUMM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_EH\_NT\_CSR\_CLLI\_SUMM"

( "MARKET" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"CLLI\_6" VARCHAR2(50 BYTE),

"EH\_CSRS" NUMBER,

"NT\_L\_CSRS" NUMBER,

"SAM\_CSRS" NUMBER,

"SAM\_NT\_L\_MATCH" NUMBER,

"SAM\_NT\_PER" NUMBER,

"EH\_NT\_L\_MATCH" NUMBER,

"EH\_NT\_PER" NUMBER,

"SAM\_EH\_NT\_ISSUES" NUMBER,

"SAM\_NT\_NL\_MATCH" NUMBER,

"SAM\_NT\_MATCH" NUMBER,

"EH\_NT\_NL\_MATCH" NUMBER,

"EH\_NT\_CSRS" NUMBER,

"EH\_NT\_MATCH" NUMBER,

"NT\_NL\_CSRS" NUMBER,

"NT\_CSRS" NUMBER,

"SAM\_NT\_MIS\_MATCH" NUMBER,

"EH\_NT\_MIS\_MATCH" NUMBER,

"BRIX\_CSRS" NUMBER DEFAULT 0,

"BRIX\_NT\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_L\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_NL\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_MIS\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_PER" NUMBER DEFAULT 0,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE"

( "MARKET" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"CLLI\_6" VARCHAR2(50 BYTE),

"EH\_CSRS" NUMBER,

"NT\_L\_CSRS" NUMBER,

"SAM\_CSRS" NUMBER,

"SAM\_NT\_L\_MATCH" NUMBER,

"SAM\_NT\_PER" NUMBER,

"EH\_NT\_L\_MATCH" NUMBER,

"EH\_NT\_PER" NUMBER,

"SAM\_EH\_NT\_ISSUES" NUMBER,

"SAM\_NT\_NL\_MATCH" NUMBER,

"SAM\_NT\_MATCH" NUMBER,

"EH\_NT\_NL\_MATCH" NUMBER,

"EH\_NT\_CSRS" NUMBER,

"EH\_NT\_MATCH" NUMBER,

"NT\_NL\_CSRS" NUMBER,

"NT\_CSRS" NUMBER,

"SAM\_NT\_MIS\_MATCH" NUMBER,

"EH\_NT\_MIS\_MATCH" NUMBER,

"BRIX\_CSRS" NUMBER DEFAULT 0,

"BRIX\_NT\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_L\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_NL\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_MIS\_MATCH" NUMBER DEFAULT 0,

"BRIX\_NT\_PER" NUMBER DEFAULT 0,

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_ENB\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_ENB\_WK"

( "CHASSIS\_TYPE" VARCHAR2(100 BYTE),

"STATUS" VARCHAR2(50 BYTE),

"SITE\_ID" VARCHAR2(100 BYTE),

"CONTROLLER\_CARD\_TYPE" VARCHAR2(100 BYTE),

"REMOTE\_IP" VARCHAR2(100 BYTE),

"IPV6\_ADDR" VARCHAR2(100 BYTE),

"ENUM" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SAM\_NT\_AL\_CSR\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SAM\_NT\_AL\_CSR\_AUDIT"

( "CSR\_VENDOR" VARCHAR2(20 BYTE),

"CSR\_DEVICE\_NAME" VARCHAR2(200 BYTE),

"SAM\_CSR\_DEVICE\_NAME" VARCHAR2(50 BYTE),

"SAM\_REACHABILITY" VARCHAR2(50 BYTE),

"SAM\_RESYNC\_STATUS" VARCHAR2(50 BYTE),

"SAM\_RESYNC\_STATE" VARCHAR2(50 BYTE),

"SAM\_SERVER\_NAME" VARCHAR2(50 BYTE),

"SAM\_LAST\_RESYNC\_START" VARCHAR2(50 BYTE),

"SAM\_LAST\_RESYNC\_END" VARCHAR2(50 BYTE),

"NT\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"LIVE\_IN\_NG" VARCHAR2(1 BYTE),

"EQ\_STATUS" VARCHAR2(20 BYTE),

"MATCH\_CODE" VARCHAR2(20 BYTE),

"MATCH\_STATUS" VARCHAR2(250 BYTE),

"EH\_CSR\_HOSTNAME" VARCHAR2(50 BYTE),

"EH\_CSR\_IPADDRESS" VARCHAR2(50 BYTE),

"EQP\_REFERENCE\_ID" NUMBER,

"MARKET" VARCHAR2(50 BYTE),

"LEAF\_DOMAIN\_ID" NUMBER,

"CLLI" VARCHAR2(50 BYTE),

"BRIX\_CSR\_SITE" VARCHAR2(50 BYTE),

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SEGMENTS\_MISSING\_EST\_COST

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST"

( "CIRC\_INST\_ID" NUMBER(\*,0),

"CIRC\_HUM\_ID" VARCHAR2(60 BYTE),

"INSERTED\_ON" DATE,

"IS\_COST\_MISSING" NUMBER(\*,0) DEFAULT 0,

"AREA" VARCHAR2(50 BYTE),

"REGION" VARCHAR2(50 BYTE),

"DOMAIN" VARCHAR2(50 BYTE)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE( INITIAL 8388608 NEXT 8388608 MINEXTENTS 1

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SEGMENT\_TERM\_EXP\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SEGMENT\_TERM\_EXP\_WK"

( "SEGMENT\_ID" NUMBER(10,0),

"SEGMENT\_NAME" VARCHAR2(200 BYTE),

"IS\_TERM\_EXPIRED" NUMBER(\*,0) DEFAULT 0,

"TERM\_END" DATE,

"TERRITORY" VARCHAR2(100 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"DOMAIN" VARCHAR2(50 BYTE),

"RECUR\_PERIOD" VARCHAR2(100 BYTE),

"ORDERED\_DATE" DATE,

"RECUR\_COSTS" NUMBER(10,2),

"TERM\_REASON" VARCHAR2(200 BYTE),

"TERM\_START" VARCHAR2(20 BYTE),

"TERM\_PLAN\_EXP" VARCHAR2(20 BYTE),

"LAST\_BILL\_DATE" DATE,

"TERM\_DURATION" NUMBER(13,3),

"ORDER\_NUMBER" VARCHAR2(100 BYTE),

"M2M\_YN" CHAR(1 BYTE),

"DATE\_BEGIN" DATE,

"RUN\_DATE" DATE,

"A\_SITE\_ID" NUMBER(9,0),

"A\_SITE\_NAME" VARCHAR2(300 BYTE),

"Z\_SITE\_ID" NUMBER(9,0),

"Z\_SITE\_NAME" VARCHAR2(300 BYTE),

"VENDOR" VARCHAR2(330 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SEVONE\_CSR

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SEVONE\_CSR"

( "DEVICE\_NAME" VARCHAR2(50 BYTE),

"SOURCE" VARCHAR2(50 BYTE),

"DEVICE\_IP" VARCHAR2(50 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE),

"VENDOR" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SEVONE\_PERFORMANCE\_DATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SEVONE\_PERFORMANCE\_DATA"

( "DEVICE\_ID" NUMBER(10,0),

"DEVICE\_NAME" VARCHAR2(50 BYTE),

"DEVICE\_IP" VARCHAR2(256 BYTE),

"OBJECT\_NAME" VARCHAR2(50 BYTE),

"INDICATOR\_NAME" VARCHAR2(50 BYTE),

"INDICATOR\_TYPE" VARCHAR2(50 BYTE),

"TIMESTAMP" VARCHAR2(50 BYTE),

"VALUE" NUMBER(30,3),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SITE

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SITE"

( "SITE\_REFERENCE\_ID" NUMBER,

"PARENT\_SITE\_REF\_ID" NUMBER DEFAULT NULL,

"SITE\_NAME" VARCHAR2(100 BYTE),

"STATUS" VARCHAR2(32 BYTE) DEFAULT NULL,

"CLLI" VARCHAR2(60 BYTE) DEFAULT NULL,

"TYPE" VARCHAR2(50 BYTE) DEFAULT NULL,

"BASE\_NUM" VARCHAR2(32 BYTE) DEFAULT NULL,

"LATITUDE" VARCHAR2(32 BYTE) DEFAULT NULL,

"LONGITUDE" VARCHAR2(32 BYTE) DEFAULT NULL,

"CUSTOMER\_ID" VARCHAR2(32 BYTE) DEFAULT NULL,

"RESTRICTIONS" VARCHAR2(4000 BYTE) DEFAULT NULL,

"CONTACTS" VARCHAR2(4000 BYTE) DEFAULT NULL,

"COMMENTS" VARCHAR2(4000 BYTE) DEFAULT NULL,

"INSTALL\_DATE" VARCHAR2(30 BYTE) DEFAULT NULL,

"ADDRESS" VARCHAR2(2000 BYTE) DEFAULT NULL,

"ZIP\_CODE" VARCHAR2(25 BYTE) DEFAULT NULL,

"CITY" VARCHAR2(100 BYTE) DEFAULT NULL,

"STATE" VARCHAR2(100 BYTE) DEFAULT NULL,

"COUNTY" VARCHAR2(100 BYTE) DEFAULT NULL,

"COUNTRY" VARCHAR2(100 BYTE) DEFAULT NULL,

"ROOM" VARCHAR2(50 BYTE) DEFAULT NULL,

"FLOOR" VARCHAR2(50 BYTE) DEFAULT NULL,

"FLOOR\_PLAN\_DEPTH" NUMBER(11,3) DEFAULT NULL,

"FLOOR\_PLAN\_WIDTH" NUMBER(11,3) DEFAULT NULL,

"FLOOR\_PLAN\_HEIGHT" NUMBER(11,3) DEFAULT NULL,

"FLOOR\_PLAN\_NAME" VARCHAR2(100 BYTE) DEFAULT NULL,

"FLOOR\_PLAN\_ORIGIN\_X" NUMBER(11,3) DEFAULT NULL,

"FLOOR\_PLAN\_ORIGIN\_Y" NUMBER(11,3) DEFAULT NULL,

"OWNER" VARCHAR2(100 BYTE) DEFAULT NULL,

"REGION" VARCHAR2(100 BYTE) DEFAULT NULL,

"LATA" VARCHAR2(10 BYTE) DEFAULT NULL,

"LAST\_MODIFIED\_TIME\_STAMP" VARCHAR2(30 BYTE) DEFAULT NULL,

"LAST\_MODIFIED\_BY" VARCHAR2(100 BYTE) DEFAULT NULL,

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE) DEFAULT NULL,

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE) DEFAULT NULL,

"CO\_SITE\_ID" NUMBER DEFAULT NULL

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SITE\_ATTRIBUTES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SITE\_ATTRIBUTES"

( "SITE\_ATTRIBUTES\_ID" NUMBER(18,0),

"SITE\_REFERENCE\_ID" NUMBER(18,0),

"SITE\_DESCPT\_ID" NUMBER(18,0),

"SITE\_GROUP\_NAME" VARCHAR2(50 BYTE),

"SITE\_NAME" VARCHAR2(50 BYTE),

"SITE\_VALUE" VARCHAR2(4000 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SITE\_ATTRIBUTES\_TEST

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SITE\_ATTRIBUTES\_TEST"

( "SITE\_ATTRIBUTES\_ID" NUMBER(18,0),

"SITE\_REFERENCE\_ID" NUMBER(18,0),

"SITE\_DESCPT\_ID" NUMBER(18,0),

"SITE\_GROUP\_NAME" VARCHAR2(50 BYTE),

"SITE\_NAME" VARCHAR2(50 BYTE),

"SITE\_VALUE" VARCHAR2(4000 BYTE),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SITE\_DOMAIN\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SITE\_DOMAIN\_MAP"

( "SITE\_REFERENCE\_ID" NUMBER(18,0),

"DOMAIN\_ID" NUMBER(10,0),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SITE\_PORTAL\_DOMAIN\_MAPPING

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SITE\_PORTAL\_DOMAIN\_MAPPING"

( "SITE\_PORTAL\_DOMAIN\_CODE" VARCHAR2(20 CHAR),

"DOMAIN\_INST\_ID" NUMBER(10,0),

"SITE\_PORTAL\_DOMAIN\_MAPPING\_ID" NUMBER(10,0),

"DOMAIN\_ID" NUMBER(10,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SITE\_PORTAL\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SITE\_PORTAL\_WK"

( "REGION" VARCHAR2(20 BYTE),

"DATBASE\_NAME" VARCHAR2(20 BYTE),

"SERVER\_IP" VARCHAR2(60 BYTE),

"GROUP\_NAME" VARCHAR2(100 BYTE),

"VENDOR" VARCHAR2(100 BYTE),

"MODEL" VARCHAR2(100 BYTE),

"DEVICE\_NAME" VARCHAR2(100 BYTE),

"IP\_ADDRESS" VARCHAR2(60 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table SLOT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."SLOT"

( "SLOT\_REFERENCE\_ID" NUMBER(18,0),

"SLOT\_NAME" VARCHAR2(50 BYTE),

"LOGICAL\_SLOT\_NAME" VARCHAR2(120 BYTE),

"SLOT\_NUMBER" VARCHAR2(50 BYTE),

"EQP\_REFERENCE\_ID" NUMBER(18,0),

"SLOT\_SPEC\_REF\_ID" NUMBER(18,0),

"PARENT\_CARD\_REFERENCE\_ID" NUMBER(18,0),

"AID" VARCHAR2(100 BYTE),

"SLOT\_POSITION" VARCHAR2(15 BYTE),

"LAST\_MODIFIED\_TIME\_STAMP" VARCHAR2(30 BYTE),

"LAST\_MODIFIED\_BY" VARCHAR2(100 BYTE),

"SLOT\_STATUS" VARCHAR2(32 BYTE),

"CARD\_REFERENCE\_ID" NUMBER(18,0),

"FR\_REF\_KEY\_NAME" VARCHAR2(32 BYTE),

"FR\_REF\_KEY\_VALUE" VARCHAR2(32 BYTE),

"TRAFFIC\_BEARING" NUMBER(9,0),

"INSTANCE\_TYPE" VARCHAR2(50 BYTE),

"PHYSICAL\_SLOT\_REFERENCE\_ID" NUMBER(18,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table STRIPPED\_SEGMENT\_DATA

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."STRIPPED\_SEGMENT\_DATA"

( "NTLS\_SEGMENT\_ID" NUMBER(10,0),

"NTLS\_SEGMENT\_NAME" VARCHAR2(200 BYTE),

"NTLS\_SEG\_NAME\_STRIP" VARCHAR2(200 BYTE),

"NTLS\_VENDOR" VARCHAR2(330 BYTE),

"NTLS\_VENDOR\_STRIP" VARCHAR2(330 BYTE),

"NTLS\_STATUS" VARCHAR2(100 BYTE),

"NTLS\_BAN" VARCHAR2(100 BYTE),

"NTLS\_BAN\_STRIP" VARCHAR2(100 BYTE),

"NTLS\_RECUR\_COSTS" NUMBER,

"NTLS\_RECUR\_COSTS\_STRIP" NUMBER,

"NTLS\_SEG\_UNQ" CHAR(1 BYTE),

"NTLS\_SEG\_VENDOR\_UNQ" CHAR(1 BYTE),

"NTLS\_SEG\_VENDOR\_BAN\_UNQ" CHAR(1 BYTE),

"NTLS\_BILL\_DATE" DATE,

"NTLS\_BILL\_DATE\_CHAR" VARCHAR2(20 BYTE),

"NTLS\_TYPE" VARCHAR2(500 BYTE),

"NTLS\_BANDWIDTH" VARCHAR2(90 BYTE),

"NTLS\_CURR\_PATH\_INST\_ID" NUMBER(9,0),

"NTLS\_NEXT\_PATH\_INST\_ID" NUMBER(9,0),

"EXTRACT\_DATE" VARCHAR2(50 BYTE) DEFAULT TO\_CHAR(SYSDATE, 'yyyymmdd')

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table TEMP\_NT\_SITERRA\_CSR

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."TEMP\_NT\_SITERRA\_CSR"

( "SITE\_INST\_ID" NUMBER(9,0),

"CSR\_ID" VARCHAR2(110 BYTE),

"CSR\_DEVICE\_IP" VARCHAR2(4000 BYTE),

"CSR\_SYSTEM\_IP" VARCHAR2(4000 BYTE),

"CSR\_MGMT\_IP" VARCHAR2(4000 BYTE),

"CSR\_LTE\_LOOPBACK\_IPV4" VARCHAR2(4000 BYTE),

"CSR\_LTE\_LOOPBACK\_IPV6" VARCHAR2(4000 BYTE),

"CSR\_MAC\_ADDRESS" VARCHAR2(4000 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table TEMP\_NT\_SITERRA\_ENB

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."TEMP\_NT\_SITERRA\_ENB"

( "SITE\_INST\_ID" NUMBER(9,0),

"SITE\_DESCRIPTION" VARCHAR2(150 BYTE),

"EQUIP\_INST\_ID" NUMBER(18,0),

"DESCR" VARCHAR2(110 BYTE),

"NODE" CHAR(1 BYTE),

"IP" CHAR(1 BYTE),

"STATUS" VARCHAR2(32 BYTE),

"XNG\_EQUIP\_NAME" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table TEST\_HEAD\_XREF

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."TEST\_HEAD\_XREF"

( "TEST\_HEAD\_NAME" VARCHAR2(100 BYTE),

"STATUS" VARCHAR2(10 BYTE),

"DOMAIN\_INST\_ID" NUMBER,

"ALLTEL\_DO\_NOT\_REMOVE" VARCHAR2(1 BYTE),

"ALLTEL\_CENTEST\_TID" VARCHAR2(50 BYTE),

"ALLTEL\_CENTEST\_IP" VARCHAR2(15 BYTE),

"TERRITORY" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"SUB\_MARKET" VARCHAR2(50 BYTE),

"TERRITORY\_MARKET\_SUB" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VDDM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VDDM"

( "VSM\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"VSM\_CLASS" VARCHAR2(100 BYTE),

"AS\_OF\_DATE" DATE,

"ALLTEL\_DO\_NOT\_REMOVE" VARCHAR2(1 BYTE),

"LEAF\_DOMAIN\_INST\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VSM\_DEVICES\_DOMAIN\_MAP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VSM\_DEVICES\_DOMAIN\_MAP"

( "VSM\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"VSM\_CLASS" VARCHAR2(100 BYTE),

"AS\_OF\_DATE" DATE,

"ALLTEL\_DO\_NOT\_REMOVE" VARCHAR2(1 BYTE),

"LEAF\_DOMAIN\_INST\_ID" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VSM\_DEVICES\_IPADDRESSES

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VSM\_DEVICES\_IPADDRESSES"

( "VSM\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"VSM\_IP" VARCHAR2(254 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VSM\_DEVICES\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VSM\_DEVICES\_WK"

( "VSM\_DEVICE\_NAME" VARCHAR2(100 BYTE),

"VSM\_CLASS" VARCHAR2(100 BYTE),

"VSM\_GATEWAY\_NAME" VARCHAR2(100 BYTE),

"VSM\_IP" VARCHAR2(100 BYTE),

"SYS\_CREATION\_DATE" DATE,

"SYS\_UPDATE\_DATE" DATE,

"OPERATOR\_ID" VARCHAR2(15 BYTE),

"APPLICATION\_ID" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VSM\_SANE\_NAME\_EMS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VSM\_SANE\_NAME\_EMS"

( "VSM\_DEVICE\_NAME\_EMS" VARCHAR2(100 BYTE),

"SANE\_NAME\_EMS" VARCHAR2(100 BYTE),

"EMS\_PATH" VARCHAR2(100 BYTE),

"EMS\_STATUS" VARCHAR2(100 BYTE),

"CREATE\_DATE" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VSM\_SANE\_RAW\_NAME\_BSM

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VSM\_SANE\_RAW\_NAME\_BSM"

( "VSM\_DEVICE\_NAME\_BSM" VARCHAR2(100 BYTE),

"SANE\_NAME\_BSM" VARCHAR2(100 BYTE),

"RAW\_NAME\_BSM" VARCHAR2(100 BYTE),

"BSM\_PATH" VARCHAR2(100 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_ERROR\_DETAILS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_ERROR\_DETAILS"

( "PROCESS\_ID" NUMBER,

"ERROR\_ID" NUMBER,

"ERROR\_MSG" VARCHAR2(3000 BYTE),

"LOG\_DATE" DATE DEFAULT sysdate,

"NOTIFY" CHAR(1 BYTE) DEFAULT 'N',

"EMAIL\_SENT" CHAR(1 BYTE) DEFAULT 'N',

"EMAIL\_SENT\_DATE" DATE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_LUCENT\_CELL\_PATHS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_LUCENT\_CELL\_PATHS"

( "LOCALE" VARCHAR2(8 BYTE),

"PATH\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"SWITCH\_ID" VARCHAR2(8 BYTE),

"CELL\_NAME" VARCHAR2(100 BYTE),

"CELL\_NUMBER" NUMBER,

"URC\_CRC" NUMBER,

"DS1\_NUMBER" NUMBER,

"NG\_PATH\_STATUS" VARCHAR2(20 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(10 BYTE),

"PARSE\_STATUS" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_MOTO\_EVDO\_CELL\_PATHS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_MOTO\_EVDO\_CELL\_PATHS"

( "PATH\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"IPBSCDO\_NUMBER" NUMBER,

"MCCDO\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_MOTO\_EVDO\_CELL\_PATHS\_TEMP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_MOTO\_EVDO\_CELL\_PATHS\_TEMP"

( "PATH\_NAME" VARCHAR2(100 BYTE),

"CIRC\_PATH\_INST\_ID" NUMBER,

"IPBSCDO\_NUMBER" NUMBER,

"MCCDO\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"XNG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"XNG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_MOTO\_UBS\_CELL\_PATHS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_MOTO\_UBS\_CELL\_PATHS"

( "TRAIL\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"LOCALE" VARCHAR2(10 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"BTS\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(10 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_MOTO\_UBS\_CELL\_PATHS\_TEMP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_MOTO\_UBS\_CELL\_PATHS\_TEMP"

( "TRAIL\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"LOCALE" VARCHAR2(10 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"IPBSCDO\_NUMBER" NUMBER,

"BTS\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(10 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_MOTO\_VOICE\_CELL\_PATHS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_MOTO\_VOICE\_CELL\_PATHS"

( "TRAIL\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"LOCALE" VARCHAR2(10 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"BTS\_NUMBER" NUMBER,

"CLUSTER\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_MOTO\_VOICE\_CELL\_PATHS\_TEMP

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_MOTO\_VOICE\_CELL\_PATHS\_TEMP"

( "TRAIL\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"LOCALE" VARCHAR2(10 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"BTS\_NUMBER" NUMBER,

"CLUSTER\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_NETWORK\_ORG

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_NETWORK\_ORG"

( "INST\_ID" NUMBER,

"AREA" VARCHAR2(50 BYTE),

"REGION" VARCHAR2(50 BYTE),

"MARKET" VARCHAR2(50 BYTE),

"REGION\_DOMAIN\_NAME" VARCHAR2(50 BYTE),

"MARKET\_DOMAIN\_NAME" VARCHAR2(50 BYTE),

"MARKET\_TERRITORY" VARCHAR2(50 BYTE),

"TERRITORY" VARCHAR2(100 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"SUB\_MARKET\_LEAF" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_NT\_EVDO\_CELL\_PATHS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_NT\_EVDO\_CELL\_PATHS"

( "TRAIL\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"LOCALE" VARCHAR2(10 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"SLOT\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"DOM\_NODE\_IP" VARCHAR2(100 BYTE),

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table VZW\_NT\_VOICE\_CELL\_PATHS

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."VZW\_NT\_VOICE\_CELL\_PATHS"

( "TRAIL\_NAME" VARCHAR2(100 BYTE),

"TRAIL\_ID" NUMBER,

"LOCALE" VARCHAR2(10 BYTE),

"SWITCH\_ID" VARCHAR2(100 BYTE),

"DCG\_NUMBER" NUMBER,

"SPAN\_NUMBER" NUMBER,

"NG\_PATH\_STATUS" VARCHAR2(100 BYTE),

"NG\_PATH\_TYPE" VARCHAR2(100 BYTE),

"PARSE\_STATUS" VARCHAR2(100 BYTE),

"BANDWIDTH" VARCHAR2(50 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table XCONNECT\_SUMMARY

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."XCONNECT\_SUMMARY"

( "TERRITORY" VARCHAR2(100 BYTE),

"MARKET" VARCHAR2(100 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"AUDITABLE" NUMBER,

"NOT\_AUDITABLE" NUMBER,

"TOTAL\_XCONN\_COUNT" NUMBER,

"MATCHED\_XCONN\_COUNT" NUMBER,

"COMPLIANCE\_PCT" NUMBER

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table XCONN\_AUDIT

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."XCONN\_AUDIT"

( "TERRITORY" VARCHAR2(100 BYTE),

"MARKET" VARCHAR2(100 BYTE),

"SUB\_MARKET" VARCHAR2(100 BYTE),

"NG\_SITE\_INST\_ID" NUMBER(9,0),

"NG\_SITE\_NAME" VARCHAR2(50 BYTE),

"NG\_EQUIP\_INST\_ID" NUMBER(9,0),

"NG\_EQUIP\_CONTAINER" VARCHAR2(128 BYTE),

"NG\_DEVICE\_TYPE" VARCHAR2(20 BYTE),

"NG\_TARGET\_ID" VARCHAR2(30 BYTE),

"NG\_DOMAIN\_INST\_ID" NUMBER(9,0),

"AUDIT\_STATUS" VARCHAR2(20 BYTE),

"AUDIT\_DATE" DATE,

"LAST\_SUCCESSFUL\_AUDIT\_DATE" DATE,

"PERCENT\_CORRECT\_IN\_NG" VARCHAR2(10 BYTE),

"MATCHED\_XCONN\_COUNT" NUMBER(9,0),

"DISCREPANCY\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_XCONN\_COUNT" NUMBER(9,0),

"TOTAL\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"MATCHED\_CIRC\_PATH\_COUNT" NUMBER(9,0),

"PERCENT\_CORRECT\_FROM\_NETWORK" VARCHAR2(10 BYTE),

"DESCRIPTION" VARCHAR2(300 BYTE),

"NE\_CLLI" VARCHAR2(50 BYTE),

"STATUS" VARCHAR2(30 BYTE),

"METROWATCH\_DB" VARCHAR2(50 BYTE),

"METROWATCH\_NE\_NAME" VARCHAR2(40 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Table XNG\_CSR\_VLAN\_PATHS\_WK

--------------------------------------------------------

CREATE TABLE "NG\_REPORTS"."XNG\_CSR\_VLAN\_PATHS\_WK"

( "EQP\_REFERENCE\_ID" NUMBER(9,0),

"trail\_id" NUMBER(9,0),

"PORT\_BANDWIDTH" VARCHAR2(30 BYTE),

"PATH\_TYPE" VARCHAR2(30 BYTE),

"VLAN\_INST\_ID" NUMBER(9,0),

"VLAN\_HUM\_ID" VARCHAR2(100 BYTE),

"VLAN\_NUMBER" NUMBER,

"VLAN\_STATUS" VARCHAR2(50 BYTE),

"EXTRACT\_DATE" DATE DEFAULT sysdate,

"card\_reference\_id" NUMBER,

"CARD\_TYPE" VARCHAR2(50 BYTE),

"TEST\_HEAD\_NAME" VARCHAR2(50 BYTE)

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for View BTOD\_NTLS\_PP\_V

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."BTOD\_NTLS\_PP\_V" ("BAN\_CATEGORY\_1", "BAN\_CATEGORY\_2", "BAN\_CUSTOM\_FIELD\_1", "MASTER\_BAN", "BAN", "BAN\_CUSTOM\_FIELD\_4", "STRIPPED\_EC\_CIRCUIT\_ID", "INVENTORY\_VENDOR\_CODE", "CHARGE\_AMOUNT", "BTOD\_EXTRACT\_DATE", "NTLS\_TERRITORY", "NTLS\_MARKET\_TERRITORY", "NTLS\_SUB\_MARKET", "BTOD\_NTLS\_MATCH", "NTLS\_SEGMENT\_ID", "NTLS\_SEGMENT\_NAME", "NTLS\_TYPE", "NTLS\_VENDOR", "NTLS\_BANDWIDTH", "NTLS\_STATUS", "NTLS\_EXTRACT\_DATE", "BAN\_CUSTOM\_FIELD\_3", "BTOD\_IN\_SERVICE\_DATE", "NTLS\_IN\_SERVICE\_DATE", "ORIG\_COMPANY\_NUM", "ORIG\_COMPANY\_NAME", "INVOICE", "INVOICE\_STATUS", "LATEST\_BILL\_DATE", "Z\_SIDE\_SITE", "Z\_SIDE\_LATITUDE", "Z\_SIDE\_LONGITUDE", "ESTIMATED\_COST", "RENEWAL\_DATE", "TERM\_DURATION", "PATH911") AS

WITH bdp

AS (SELECT bdp.BAN\_CATEGORY\_1,

bdp.BAN\_CATEGORY\_2,

bdp.BAN\_CUSTOM\_FIELD\_1,

bdp.MASTER\_BAN,

bdp.BAN,

bdp.BAN\_CUSTOM\_FIELD\_4,

bdp.STRIPPED\_EC\_CIRCUIT\_ID,

bdp.INVENTORY\_VENDOR\_CODE,

NVL (TO\_NUMBER (bdp.CHARGE\_AMOUNT), 0) charge\_amount,

TO\_DATE (bdp.BTOD\_EXTRACT\_DATE, 'yyyymmdd')

BTOD\_EXTRACT\_DATE,

'Y' BTOD\_NTLS\_MATCH,

n.NTLS\_SEGMENT\_ID,

n.NTLS\_SEGMENT\_NAME,

n.NTLS\_TYPE,

n.NTLS\_VENDOR,

n.NTLS\_BANDWIDTH,

n.NTLS\_STATUS,

bdp.BAN\_CUSTOM\_FIELD\_3,

TO\_DATE (

SUBSTR (in\_service\_date,

1,

INSTR (bdp.in\_service\_date, ' ') - 1),

'yyyy/mm/dd')

in\_service\_date,

bdp.orig\_company\_num,

bdp.orig\_company\_name,

REPLACE (bdp.invoice, CHR (13), '') invoice,

bdp.latest\_bill\_date,

1 AS match\_type

FROM (SELECT a.\*,

TO\_DATE (

SUBSTR (a.bill\_date,

1,

INSTR (a.bill\_date, ' ')),

'yyyy/mm/dd')

latest\_bill\_date

FROM ng\_reports.btod\_pp\_data a) bdp,

ng\_reports.stripped\_segment\_data n

WHERE bdp.stripped\_ec\_circuit\_id = n.ntls\_seg\_name\_strip

AND bdp.inventory\_vendor\_code = n.ntls\_vendor\_strip

AND bdp.ban = n.ntls\_ban\_strip

UNION

SELECT bdp.BAN\_CATEGORY\_1,

bdp.BAN\_CATEGORY\_2,

bdp.BAN\_CUSTOM\_FIELD\_1,

bdp.MASTER\_BAN,

bdp.BAN,

bdp.BAN\_CUSTOM\_FIELD\_4,

bdp.STRIPPED\_EC\_CIRCUIT\_ID,

bdp.INVENTORY\_VENDOR\_CODE,

NVL (TO\_NUMBER (bdp.CHARGE\_AMOUNT), 0) charge\_amount,

TO\_DATE (bdp.BTOD\_EXTRACT\_DATE, 'yyyymmdd')

BTOD\_EXTRACT\_DATE,

'Y' BTOD\_NTLS\_MATCH,

n.NTLS\_SEGMENT\_ID,

n.NTLS\_SEGMENT\_NAME,

n.NTLS\_TYPE,

n.NTLS\_VENDOR,

n.NTLS\_BANDWIDTH,

n.NTLS\_STATUS,

bdp.BAN\_CUSTOM\_FIELD\_3,

TO\_DATE (

SUBSTR (in\_service\_date,

1,

INSTR (bdp.in\_service\_date, ' ') - 1),

'yyyy/mm/dd')

in\_service\_date,

bdp.orig\_company\_num,

bdp.orig\_company\_name,

REPLACE (bdp.invoice, CHR (13), '') invoice,

bdp.latest\_bill\_date,

2 AS match\_type

FROM (SELECT a.\*,

TO\_DATE (

SUBSTR (a.bill\_date,

1,

INSTR (a.bill\_date, ' ')),

'yyyy/mm/dd')

latest\_bill\_date

FROM ng\_reports.btod\_pp\_data a) bdp,

ng\_reports.stripped\_segment\_data n

WHERE bdp.stripped\_ec\_circuit\_id = n.ntls\_seg\_name\_strip

AND bdp.inventory\_vendor\_code = n.ntls\_vendor\_strip

UNION

SELECT bdp.BAN\_CATEGORY\_1,

bdp.BAN\_CATEGORY\_2,

bdp.BAN\_CUSTOM\_FIELD\_1,

bdp.MASTER\_BAN,

bdp.BAN,

bdp.BAN\_CUSTOM\_FIELD\_4,

bdp.STRIPPED\_EC\_CIRCUIT\_ID,

bdp.INVENTORY\_VENDOR\_CODE,

NVL (TO\_NUMBER (bdp.CHARGE\_AMOUNT), 0) charge\_amount,

TO\_DATE (bdp.BTOD\_EXTRACT\_DATE, 'yyyymmdd')

BTOD\_EXTRACT\_DATE,

'Y' BTOD\_NTLS\_MATCH,

n.NTLS\_SEGMENT\_ID,

n.NTLS\_SEGMENT\_NAME,

n.NTLS\_TYPE,

n.NTLS\_VENDOR,

n.NTLS\_BANDWIDTH,

n.NTLS\_STATUS,

bdp.BAN\_CUSTOM\_FIELD\_3,

TO\_DATE (

SUBSTR (in\_service\_date,

1,

INSTR (bdp.in\_service\_date, ' ') - 1),

'yyyy/mm/dd')

in\_service\_date,

bdp.orig\_company\_num,

bdp.orig\_company\_name,

REPLACE (bdp.invoice, CHR (13), '') invoice,

bdp.latest\_bill\_date,

3 AS match\_type

FROM (SELECT a.\*,

TO\_DATE (

SUBSTR (a.bill\_date,

1,

INSTR (a.bill\_date, ' ')),

'yyyy/mm/dd')

latest\_bill\_date

FROM ng\_reports.btod\_pp\_data a) bdp,

ng\_reports.stripped\_segment\_data n

WHERE bdp.stripped\_ec\_circuit\_id = n.ntls\_seg\_name\_strip),

mt

AS (SELECT bdp.STRIPPED\_EC\_CIRCUIT\_ID,

bdp.INVENTORY\_VENDOR\_CODE,

bdp.BAN,

1 match\_type

FROM ng\_reports.btod\_pp\_data bdp,

ng\_reports.stripped\_segment\_data n

WHERE bdp.stripped\_ec\_circuit\_id = n.ntls\_seg\_name\_strip

AND bdp.inventory\_vendor\_code = n.ntls\_vendor\_strip

AND bdp.ban = n.ntls\_ban\_strip

UNION

SELECT bdp.STRIPPED\_EC\_CIRCUIT\_ID,

bdp.INVENTORY\_VENDOR\_CODE,

bdp.BAN,

2 match\_type

FROM ng\_reports.btod\_pp\_data bdp,

ng\_reports.stripped\_segment\_data n

WHERE bdp.stripped\_ec\_circuit\_id = n.ntls\_seg\_name\_strip

AND bdp.inventory\_vendor\_code = n.ntls\_vendor\_strip

UNION

SELECT bdp.STRIPPED\_EC\_CIRCUIT\_ID,

bdp.INVENTORY\_VENDOR\_CODE,

bdp.BAN,

3 match\_type

FROM ng\_reports.btod\_pp\_data bdp,

ng\_reports.stripped\_segment\_data n

WHERE bdp.stripped\_ec\_circuit\_id = n.ntls\_seg\_name\_strip),

mt\_min

AS ( SELECT stripped\_ec\_circuit\_id,

inventory\_vendor\_code,

ban,

MIN (match\_type) match\_type

FROM mt

GROUP BY stripped\_ec\_circuit\_id, inventory\_vendor\_code, ban),

circ

AS (SELECT bdp.BAN\_CATEGORY\_1,

bdp.BAN\_CATEGORY\_2,

bdp.BAN\_CUSTOM\_FIELD\_1,

bdp.MASTER\_BAN,

bdp.BAN,

bdp.BAN\_CUSTOM\_FIELD\_4,

bdp.STRIPPED\_EC\_CIRCUIT\_ID,

bdp.INVENTORY\_VENDOR\_CODE,

bdp.charge\_amount,

bdp.BTOD\_EXTRACT\_DATE,

bdp.BTOD\_NTLS\_MATCH,

bdp.NTLS\_segment\_name,

bdp.NTLS\_segment\_id,

bdp.NTLS\_TYPE,

bdp.NTLS\_VENDOR,

bdp.NTLS\_BANDWIDTH,

bdp.NTLS\_STATUS,

bdp.BAN\_CUSTOM\_FIELD\_3,

bdp.IN\_SERVICE\_DATE,

bdp.orig\_company\_num,

bdp.orig\_company\_name,

bdp.invoice,

bdp.latest\_bill\_date,

1 extm

FROM bdp, mt\_min

WHERE bdp.stripped\_ec\_circuit\_id =

mt\_min.stripped\_ec\_circuit\_id

AND bdp.inventory\_vendor\_code =

mt\_min.inventory\_vendor\_code

AND bdp.ban = mt\_min.ban

AND bdp.match\_type = mt\_min.match\_type

UNION

SELECT a.BAN\_CATEGORY\_1,

a.BAN\_CATEGORY\_2,

a.BAN\_CUSTOM\_FIELD\_1,

a.MASTER\_BAN,

a.BAN,

a.BAN\_CUSTOM\_FIELD\_4,

a.STRIPPED\_EC\_CIRCUIT\_ID,

a.INVENTORY\_VENDOR\_CODE,

NVL (TO\_NUMBER (a.CHARGE\_AMOUNT), 0) charge\_amount,

TO\_DATE (a.BTOD\_EXTRACT\_DATE, 'yyyymmdd') BTOD\_EXTRACT\_DATE,

'N' BTOD\_NTLS\_MATCH,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

a.BAN\_CUSTOM\_FIELD\_3,

TO\_DATE (

SUBSTR (in\_service\_date,

1,

INSTR (a.in\_service\_date, ' ') - 1),

'yyyy/mm/dd')

in\_service\_date,

a.orig\_company\_num,

a.orig\_company\_name,

REPLACE (a.invoice, CHR (13), '') invoice,

TO\_DATE (

SUBSTR (a.bill\_date, 1, INSTR (a.bill\_date, ' ')),

'yyyy/mm/dd')

latest\_bill\_date,

1 extm

FROM ng\_reports.btod\_pp\_data a

WHERE NOT EXISTS

(SELECT 1

FROM mt\_min b

WHERE a.stripped\_ec\_circuit\_id =

b.stripped\_ec\_circuit\_id

AND a.inventory\_vendor\_code =

b.inventory\_vendor\_code

AND a.ban = b.ban)),

dom

AS (SELECT bxd.btod\_ban\_category\_1,

drr.territory AS n\_territory,

drr.market\_territory AS n\_market\_territory,

drr.sub\_market AS n\_sub\_market,

drr.domain\_id

FROM ng\_reports.btod\_ntls\_domain\_map bxd,

ng\_reports.domains\_regional\_reporting drr

WHERE bxd.ntls\_domain\_id = drr.domain\_id(+)),

n\_dom

AS (SELECT sdm.segment\_id,

drr.territory AS n\_territory,

drr.market\_territory AS n\_market\_territory,

drr.sub\_market AS n\_sub\_market,

sdm.domain\_id

FROM ng\_topology.segment\_domain\_map sdm,

ng\_reports.domains\_regional\_reporting drr

WHERE sdm.domain\_id = drr.domain\_id),

extd

AS (SELECT 1 extm, extract\_date ntls\_extract\_date

FROM ng\_reports.stripped\_segment\_data

WHERE ROWNUM = 1),

zs

AS (SELECT ts.segment\_id,

s.site\_name,

s.latitude,

s.longitude

FROM NG\_TOPOLOGY.TRAIL\_SEGMENT ts, NG\_REPORTS.SITE s

WHERE ts.z\_site\_id = s.site\_reference\_id),

p911

AS (SELECT ts.segment\_id, 'Y' path911

FROM ng\_topology.trail\_segment ts, ng\_topology.trail t

WHERE ts.current\_trail\_id = t.trail\_id AND t.TYPE LIKE '%911')

SELECT DISTINCT

BAN\_CATEGORY\_1,

BAN\_CATEGORY\_2,

BAN\_CUSTOM\_FIELD\_1,

MASTER\_BAN,

BAN,

BAN\_CUSTOM\_FIELD\_4,

STRIPPED\_EC\_CIRCUIT\_ID,

INVENTORY\_VENDOR\_CODE,

charge\_amount,

BTOD\_EXTRACT\_DATE,

NVL (dom.n\_territory, NVL (xd.n\_territory, 'Unknown'))

AS ntls\_territory,

NVL (dom.n\_market\_territory,

NVL (xd.n\_market\_territory, 'Unknown'))

AS ntls\_market\_territory,

NVL (dom.n\_sub\_market, NVL (xd.n\_sub\_market, 'Unknown'))

AS ntls\_sub\_market,

BTOD\_NTLS\_MATCH,

NTLS\_SEGMENT\_ID,

NTLS\_SEGMENT\_NAME,

NTLS\_TYPE,

NTLS\_VENDOR,

NTLS\_BANDWIDTH,

NTLS\_STATUS,

extd.NTLS\_EXTRACT\_DATE,

BAN\_CUSTOM\_FIELD\_3,

IN\_SERVICE\_DATE btod\_in\_service\_date,

ci.in\_service ntls\_in\_service\_date,

orig\_company\_num,

orig\_company\_name,

RTRIM (LTRIM (SUBSTR (invoice,

1,

INSTR (invoice,

' - ',

-1,

1)

- 1)))

invoice,

RTRIM (LTRIM (SUBSTR (invoice,

INSTR (invoice,

' - ',

-1,

1)

+ 3)))

invoice\_status,

latest\_bill\_date,

zs.site\_name AS z\_side\_site,

zs.latitude AS z\_side\_latitude,

zs.longitude AS z\_side\_longitude,

ci.non\_recur\_costs AS estimated\_cost,

ci.renewal\_date,

ci.term\_duration,

NVL (p911.path911, 'N') AS path911

FROM circ,

dom,

n\_dom xd,

extd,

zs,

p911,

ng\_topology.trail\_segment ci

WHERE circ.ban\_category\_1 = dom.btod\_ban\_category\_1(+)

AND circ.ntls\_segment\_id = xd.segment\_id(+)

AND circ.extm = extd.extm(+)

AND circ.ntls\_segment\_id = zs.segment\_id(+)

AND circ.ntls\_segment\_id = p911.segment\_id(+)

AND circ.ntls\_segment\_id = ci.segment\_id(+)

;

--------------------------------------------------------

-- DDL for View CIRC\_AUDIT\_LOG\_VW

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."CIRC\_AUDIT\_LOG\_VW" ("LOCAL\_ID", "QUALIFIED\_NAME", "COMMIT\_DATE", "AUTHOR", "TYPE", "CHANGED\_PROPERTIES", "VERSION", "STATE") AS

SELECT g.local\_id,

cc.qualified\_name,

c.commit\_date,

c.author,

s.type,

s.changed\_properties,

s.version,

s.state

FROM jv\_global\_id g,

jv\_cdo\_class cc,

jv\_commit c,

jv\_snapshot s

WHERE cc.cdo\_class\_pk = g.cdo\_class\_fk

AND g.global\_id\_pk = s.global\_id\_fk

AND s.commit\_fk = c.commit\_pk

;

--------------------------------------------------------

-- DDL for View CIRC\_BC\_THRESHOLD\_DATA\_VW

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."CIRC\_BC\_THRESHOLD\_DATA\_VW" ("VLAN\_TRAIL\_ID", "BC\_FRAME\_DELAY", "FD\_WARN\_THRESHOLD", "FD\_CRITICAL\_THRESHOLD", "BC\_FRAME\_DELAY\_VARIATION", "FDV\_WARN\_THRESHOLD", "FDV\_CRITICAL\_THRESHOLD", "CLASS\_OF\_SERVICE") AS

SELECT

ta2."VLAN\_TRAIL\_ID",

ta2."BC\_FRAME\_DELAY",

ta2."FD\_WARN\_THRESHOLD",

ta2."FD\_CRITICAL\_THRESHOLD",

ta2."BC\_FRAME\_DELAY\_VARIATION",

ta2."FDV\_WARN\_THRESHOLD",

ta2."FDV\_CRITICAL\_THRESHOLD",

ta2."CLASS\_OF\_SERVICE"

FROM

ng\_topology.trail tr

JOIN (

SELECT

ta.trail\_id vlan\_trail\_id,

ta.attribute\_value bc\_frame\_delay,

TO\_CHAR(

CASE

WHEN ta.attribute\_value \* 1.4 > 5.0 THEN ta.attribute\_value \* 1.4

ELSE 5.0

END

) fd\_warn\_threshold,

TO\_CHAR(

CASE

WHEN ta.attribute\_value \* 1.7 > 10.0 THEN ta.attribute\_value \* 1.7

ELSE 10.0

END

) fd\_critical\_threshold,

TRIM(TO\_CHAR(ta1.attribute\_value,'999.999999999') ) bc\_frame\_delay\_variation,

TO\_CHAR(1.5) fdv\_warn\_threshold,

TO\_CHAR(2.0) fdv\_critical\_threshold,

'p5' class\_of\_service

FROM

(

SELECT

\*

FROM

ng\_topology.trail\_attributes

WHERE

attribute\_name = 'BC Frame Delay'

AND group\_name = 'Path Information'

AND attribute\_value <= 50

) ta

FULL OUTER JOIN (

SELECT

\*

FROM

ng\_topology.trail\_attributes

WHERE

attribute\_name = 'BC Frame Delay Variation'

AND group\_name = 'Path Information'

AND attribute\_value <= 2

) ta1 ON ta.trail\_id = ta1.trail\_id

) ta2 ON tr.trail\_id = ta2.vlan\_trail\_id

UNION

SELECT

ta2."VLAN\_TRAIL\_ID",

ta2."BC\_FRAME\_DELAY",

ta2."FD\_WARN\_THRESHOLD",

ta2."FD\_CRITICAL\_THRESHOLD",

ta2."BC\_FRAME\_DELAY\_VARIATION",

ta2."FDV\_WARN\_THRESHOLD",

ta2."FDV\_CRITICAL\_THRESHOLD",

ta2."CLASS\_OF\_SERVICE"

FROM

ng\_topology.trail tr

JOIN (

SELECT

ta.trail\_id vlan\_trail\_id,

ta.attribute\_value bc\_frame\_delay,

TO\_CHAR(

CASE

WHEN ta.attribute\_value \* 1.7 > 30.0 THEN ta.attribute\_value \* 1.7

ELSE 30.0

END

) fd\_warn\_threshold,

TO\_CHAR(

CASE

WHEN ta.attribute\_value \* 1.9 > 50.0 THEN ta.attribute\_value \* 1.9

ELSE 50.0

END

) fd\_critical\_threshold,

TRIM(TO\_CHAR(ta1.attribute\_value,'999.999999999') ) bc\_frame\_delay\_variation,

TO\_CHAR(7.0) fdv\_warn\_threshold,

TO\_CHAR(10.0) fdv\_critical\_threshold,

'p0' class\_of\_service

FROM

(

SELECT

\*

FROM

ng\_topology.trail\_attributes

WHERE

attribute\_name = 'BC Frame Delay'

AND group\_name = 'Path Information'

AND attribute\_value <= 50

) ta

FULL OUTER JOIN (

SELECT

\*

FROM

ng\_topology.trail\_attributes

WHERE

attribute\_name = 'BC Frame Delay Variation'

AND group\_name = 'Path Information'

AND attribute\_value <= 2

) ta1 ON ta.trail\_id = ta1.trail\_id

) ta2 ON tr.trail\_id = ta2.vlan\_trail\_id

;

--------------------------------------------------------

-- DDL for View CIRC\_SEG\_NEW\_DECOM\_DET\_TESS\_VW

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DET\_TESS\_VW" ("SEGMENT\_ID", "SEGMENT\_NAME", "STATUS", "SEGMENT\_TYPE", "BANDWIDTH", "VENDOR", "RPT\_ID", "DOMAIN\_ID", "DOMAIN\_NAME", "TERRITORY", "MARKET\_TERRITORY", "SUB\_MARKET", "Z\_SITE\_ID", "SITE\_NAME", "ACTION\_TYPE", "COMMIT\_DATE", "AUTHOR", "MRC", "ESTIMATED\_COSTS", "IN\_SERVICE", "TERM\_END", "DURATION", "DISCONNECT\_PON", "DISCONNECT\_ORDER\_NUM", "DISCONNECT\_ORDER\_TS", "DISCONNECT\_TS", "BILLING\_CODE", "LEASE\_BANDWIDTH", "CONTRACT\_DOC", "NEW", "NEW\_TOTAL", "NEW\_BY\_TESS", "NEW\_NOT\_TESS\_CONTRACT", "NEW\_NOT\_TESS\_NO\_CONTRACT", "DECOM") AS

WITH NEW\_SEGS

AS (SELECT TS.SEGMENT\_ID,

TS.NAME SEGMENT\_NAME,

TS.STATUS,

TS.TYPE SEGMENT\_TYPE,

TS.BANDWIDTH,

TS.VENDOR,

SD.RPT\_ID,

SD.DOMAIN\_ID,

SD.DOMAIN\_NAME,

SD.TERRITORY,

SD.MARKET\_TERRITORY,

SD.SUB\_MARKET,

SD.Z\_SITE\_ID,

SD.SITE\_NAME,

SD.ACTION\_TYPE,

SD.COMMIT\_DATE,

SD.AUTHOR,

--SD.MRC,

TS.RECUR\_COSTS MRC,

TS.NON\_RECUR\_COSTS ESTIMATED\_COSTS,

TS.IN\_SERVICE,

TS.RENEWAL\_DATE TERM\_END,

(CASE WHEN TS.RENEWAL\_DATE IS NOT NULL AND SD.COMMIT\_DATE IS NOT NULL AND TS.RENEWAL\_DATE > SD.COMMIT\_DATE THEN ROUND(TS.RENEWAL\_DATE - SD.COMMIT\_DATE) ELSE NULL END) AS DURATION,

SD.DISCONNECT\_PON,

SD.DISCONNECT\_ORDER\_NUM,

SD.DISCONNECT\_ORDER\_TS,

SD.DISCONNECT\_TS,

CASE

WHEN TS.BILLING\_CODE IS NULL THEN SD.BILLING\_CODE

ELSE TS.BILLING\_CODE

END

BILLING\_CODE,

CASE

WHEN ( ( TS.TYPE = 'LEASED SERVICE'

OR ( TS.TYPE != 'LEASED SERVICE'

AND TS.VENDOR NOT LIKE 'VZW%'

AND TS.VENDOR NOT LIKE

'VERIZON WIRELESS%'))

AND ( TS.BANDWIDTH IN

('OC192',

'10 Gbps',

'100 Gbps',

'DARK FIBER')

OR ( TS.BANDWIDTH LIKE '% Mbps'

AND TO\_NUMBER (

SUBSTR (

TS.BANDWIDTH,

1,

INSTR (TS.BANDWIDTH, ' '))) >=

50)))

THEN

1

ELSE

0

END

AS LEASE\_BANDWIDTH

FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL SD

JOIN

NG\_TOPOLOGY.TRAIL\_SEGMENT ts

ON SD.SEGMENT\_ID = TS.SEGMENT\_ID

WHERE SD.ACTION\_TYPE = 'NEW'),

SEG\_CA

AS (SELECT NEW\_SEGS.\*, SCD.ATTRIBUTE\_VALUE CONTRACT\_DOC

FROM NEW\_SEGS

LEFT OUTER JOIN

(SELECT CAS.\*

FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES CAS

-- JOIN

-- VZWNET.VAL\_ATTR\_NAME VAN

-- ON CAS.VAL\_ATTR\_INST\_ID =

-- VAN.VAL\_ATTR\_INST\_ID

WHERE CAS.ATTRIBUTE\_NAME = 'Contract Documentation') SCD

ON NEW\_SEGS.SEGMENT\_ID = SCD.SEGMENT\_ID)

SELECT SC."SEGMENT\_ID",SC."SEGMENT\_NAME",SC."STATUS",SC."SEGMENT\_TYPE",SC."BANDWIDTH",SC."VENDOR",SC."RPT\_ID",SC."DOMAIN\_ID",SC."DOMAIN\_NAME",SC."TERRITORY",SC."MARKET\_TERRITORY",SC."SUB\_MARKET",SC."Z\_SITE\_ID",SC."SITE\_NAME",SC."ACTION\_TYPE",SC."COMMIT\_DATE",SC."AUTHOR",SC."MRC",SC."ESTIMATED\_COSTS",SC."IN\_SERVICE",SC."TERM\_END",SC."DURATION",SC."DISCONNECT\_PON",SC."DISCONNECT\_ORDER\_NUM",SC."DISCONNECT\_ORDER\_TS",SC."DISCONNECT\_TS",SC."BILLING\_CODE",SC."LEASE\_BANDWIDTH",SC."CONTRACT\_DOC",

1 NEW,

CASE WHEN (LEASE\_BANDWIDTH = 1) THEN 1 ELSE 0 END NEW\_TOTAL,

CASE

WHEN (LEASE\_BANDWIDTH = 1)

AND ( SC.AUTHOR = 'tessUser'

OR SC.CONTRACT\_DOC LIKE

'https://erpprd-fnprd.erp.vzwcorp.com/psp/ps/EMPLOYEE/ERP/c/CONTRACT\_MGMT.CS\_DOC\_MAINT.GBL%')

THEN

1

ELSE

0

END

NEW\_BY\_TESS,

CASE

WHEN (LEASE\_BANDWIDTH = 1)

AND ( SC.AUTHOR != 'tessUser'

AND SC.CONTRACT\_DOC NOT LIKE

'https://erpprd-fnprd.erp.vzwcorp.com/psp/ps/EMPLOYEE/ERP/c/CONTRACT\_MGMT.CS\_DOC\_MAINT.GBL%')

THEN

1

ELSE

0

END

NEW\_NOT\_TESS\_CONTRACT,

CASE

WHEN (LEASE\_BANDWIDTH = 1)

AND (SC.AUTHOR != 'tessUser' AND SC.CONTRACT\_DOC IS NULL)

THEN

1

ELSE

0

END

NEW\_NOT\_TESS\_NO\_CONTRACT,

0 DECOM

FROM SEG\_CA SC

UNION

SELECT SD.SEGMENT\_ID,

SD.SEGMENT\_NAME,

SD.STATUS,

SD.SEGMENT\_TYPE,

SD.BANDWIDTH,

SD.VENDOR,

SD.RPT\_ID,

SD.DOMAIN\_ID,

SD.DOMAIN\_NAME,

SD.TERRITORY,

SD.MARKET\_TERRITORY,

SD.SUB\_MARKET,

SD.Z\_SITE\_ID,

SD.SITE\_NAME,

SD.ACTION\_TYPE,

SD.COMMIT\_DATE,

SD.AUTHOR,

SD.MRC,

null ESTIMATED\_COSTS,

null IN\_SERVICE,

null TERM\_END,

null DURATION,

SD.DISCONNECT\_PON,

SD.DISCONNECT\_ORDER\_NUM,

SD.DISCONNECT\_ORDER\_TS,

SD.DISCONNECT\_TS,

SD.BILLING\_CODE,

0 LEASE\_BANDWIDTH,

NULL CONTRACT\_DOC,

0 NEW,

0 NEW\_TOTAL,

0 NEW\_BY\_TESS,

0 NEW\_NOT\_TESS\_CONTRACT,

0 NEW\_NOT\_TESS\_NO\_CONTRACT,

1 DECOM

FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL SD

WHERE SD.ACTION\_TYPE = 'DECOM'

;

--------------------------------------------------------

-- DDL for View CLLI\_DOMAIN\_MAP\_V

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."CLLI\_DOMAIN\_MAP\_V" ("CLLI", "SUB\_MARKET\_LEAF", "SUB\_MARKET", "MARKET\_TERRITORY", "TERRITORY", "LEAF\_DOMAIN\_NAME", "LEAF\_DOMAIN\_ID", "PARENT\_DOMAIN\_ID", "COMMENTS") AS

SELECT DISTINCT v.clli,

dlr.sub\_market\_leaf,

dlr.sub\_market,

dlr.market\_territory,

DLR.TERRITORY,

dlr.domain\_name leaf\_domain\_name,

dlr.domain\_id leaf\_domain\_id,

dlr.parent\_domain\_id,

v.comments

FROM CLLI\_DOMAIN\_MAP V

LEFT OUTER JOIN DOMAINS\_LEAF\_REPORTING dlr

ON dlr.DOMAIN\_ID = v.LEAF\_DOMAIN\_ID

;

--------------------------------------------------------

-- DDL for View DOMAINS\_ALL\_LEAVES

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."DOMAINS\_ALL\_LEAVES" ("DOMAIN\_ID", "DOMAIN\_NAME", "PARENT\_DOMAIN\_ID", "PARENT\_DOMAIN\_NAME", "REGION\_DOMAIN\_NAME", "REGION\_DOMAIN\_ID", "IS\_REGION") AS

WITH leaves

AS (SELECT domain\_id FROM uteapp.dir\_domains

MINUS

-- minus the parents

SELECT parent\_domain\_id FROM ng\_reports.domain\_parents),

dom\_parents as (select dd.domain\_id, dd.domain\_name, dp.parent\_domain\_id

from uteapp.dir\_domains dd,

ng\_reports.domain\_parents dp

where dd.domain\_id = dp.domain\_id)

SELECT DISTINCT

di.domain\_id,

di.domain\_name,

di.parent\_domain\_id,

p.domain\_name parent\_domain\_name,

CASE

WHEN vno.region\_domain\_name IS NULL THEN p.domain\_name

ELSE di.domain\_name

END

region\_domain\_name,

CASE

WHEN vno.region\_domain\_name IS NULL THEN p.domain\_id

ELSE di.domain\_id

END

region\_domain\_inst\_id,

CASE WHEN vno.region\_domain\_name IS NOT NULL THEN 'Y' END is\_region

FROM dom\_parents di

JOIN leaves l ON l.domain\_id = di.domain\_id

JOIN dom\_parents p

ON p.domain\_id = di.parent\_domain\_id

LEFT OUTER JOIN ng\_reports.vzw\_network\_org vno

ON di.domain\_name = vno.region\_domain\_name

;

--------------------------------------------------------

-- DDL for View DOMAINS\_LEAF\_REPORTING

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."DOMAINS\_LEAF\_REPORTING" ("DOMAIN\_ID", "DOMAIN\_NAME", "PARENT\_DOMAIN\_ID", "PARENT\_DOMAIN\_NAME", "SUB\_MARKET\_LEAF", "SUB\_MARKET", "MARKET\_TERRITORY", "TERRITORY", "REGION\_DOMAIN\_ID", "REGION\_DOMAIN\_NAME", "IS\_REGION") AS

SELECT DISTINCT d.DOMAIN\_ID,

d.DOMAIN\_NAME,

d.PARENT\_DOMAIN\_ID,

d.PARENT\_DOMAIN\_NAME,

vno.sub\_market\_leaf,

vno.sub\_market,

vno.market\_territory,

vno.territory,

d.REGION\_DOMAIN\_ID,

d.REGION\_DOMAIN\_NAME,

d.IS\_REGION

FROM domains\_all\_leaves d

JOIN ng\_reports.vzw\_network\_org vno

ON vno.inst\_id = domain\_id

WHERE domain\_id NOT IN (47)

;

--------------------------------------------------------

-- DDL for View DOMAINS\_REGIONAL\_REPORTING

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."DOMAINS\_REGIONAL\_REPORTING" ("DOMAIN\_ID", "DOMAIN\_NAME", "MARKET\_TERRITORY", "TERRITORY", "SUB\_MARKET") AS

SELECT DISTINCT domain\_id,

domain\_name,

market\_territory,

territory,

sub\_market

FROM uteapp.dir\_domains d

JOIN vzw\_network\_org vno ON vno.region\_domain\_name = d.domain\_name

;

--------------------------------------------------------

-- DDL for View EBH\_TERM\_BTS\_REGIONAL\_SUMMARY

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."EBH\_TERM\_BTS\_REGIONAL\_SUMMARY" ("TERRITORY\_MARKET\_SUB", "TERRITORY", "MARKET", "SUB\_MARKET", "LIVE", "RPTD", "DISCOVERED", "LU\_EBH\_COMP", "NT\_EBH\_CDMA\_COMP", "NT\_EBH\_EVDO\_COMP", "MT\_EBH\_CDMA\_COMP", "MT\_EBH\_EVDO\_COMP", "TOTAL\_COMPLIANCE") AS

WITH regions

AS (SELECT territory, market\_territory market, sub\_market

FROM domains\_leaf\_reporting dlr

WHERE territory NOT IN ('OSS', 'NNO')

UNION

SELECT DISTINCT territory, NULL, NULL

FROM domains\_leaf\_reporting dlr

WHERE territory NOT IN ('OSS')

UNION

SELECT DISTINCT territory, market\_territory market, NULL

FROM domains\_leaf\_reporting dlr

WHERE territory NOT IN ('OSS', 'NNO')

-- UNION

-- SELECT 'Unknown', 'Unknown', 'Unknown' FROM DUAL

UNION

SELECT 'grand Total', NULL, NULL FROM DUAL), ---select \* from regions

lucent -- remove \_wrk

AS (SELECT territory,

market\_territory market,

sub\_market,

NVL (ebh\_CSR\_matched\_live, 0) lu\_ebh\_live,

NVL (ebh\_csr\_reported, 0) lu\_ebh\_reported,

NVL (tot.EBH\_DISCOVERED, 0) lu\_ebh\_discovered,

NVL (ebh\_csr\_comp, 0) lu\_ebh\_comp

FROM lucent\_audit\_reg\_summ tot),

nt\_cdma

AS (SELECT territory,

market\_territory market,

sub\_market,

NVL (ebh\_csr\_matched\_live, 0) nt\_ebh\_cdma\_live,

NVL (ebh\_reported, 0) nt\_ebh\_cdma\_reported,

NVL (tot.EBH\_DISCOVERED, 0) nt\_ebh\_cdma\_discovered,

NVL (ebh\_comp, 0) nt\_ebh\_cdma\_comp

FROM nortel\_cdma\_audit\_reg\_summ tot),

nt\_evdo

AS (SELECT territory,

market\_territory market,

sub\_market,

NVL (tot\_up\_ebh\_spans\_w\_l\_pathscsr, 0) nt\_ebh\_evdo\_live,

NVL (ebh\_reported, 0) nt\_ebh\_evdo\_reported,

NVL (tot.TOT\_UP\_EBH\_SPANS, 0) nt\_ebh\_evdo\_discovered,

NVL (ebh\_comp, 0) nt\_ebh\_evdo\_comp

FROM nt\_evdo\_bts\_term\_reg\_summ tot),

mt\_cdma

AS (SELECT territory,

market\_territory market,

sub\_market,

NVL (ebh\_matched\_live\_csr, 0) mt\_ebh\_cdma\_live,

NVL (ebh\_reported, 0) mt\_ebh\_cdma\_reported,

NVL (tot.EBH\_DISCOVERED, 0) mt\_ebh\_cdma\_discovered,

NVL (ebh\_comp, 0) mt\_ebh\_cdma\_comp

FROM moto\_cdma\_audit\_reg\_summ tot),

mt\_evdo

AS (SELECT territory,

territory market,

sub\_market,

NVL (ebh\_matched\_live\_csr, 0) mt\_ebh\_evdo\_live,

NVL (ebh\_reported, 0) mt\_ebh\_evdo\_reported,

NVL (tot.EBH\_DISCOVERED, 0) mt\_ebh\_evdo\_discovered,

NVL (ebh\_comp, 0) mt\_ebh\_evdo\_comp

FROM moto\_evdo\_audit\_reg\_summ tot)

SELECT CASE

WHEN r.sub\_market IS NOT NULL THEN r.sub\_market --= 'Grand Total'

WHEN r.sub\_market IS NULL AND r.market IS NOT NULL THEN r.market

WHEN r.sub\_market IS NULL AND r.market IS NULL THEN r.territory

END

territory\_market\_sub,

r.territory,

r.market,

r.sub\_market,

NVL (lu\_ebh\_live, 0)

+ NVL (nt\_ebh\_cdma\_live, 0)

+ NVL (nt\_ebh\_evdo\_live, 0)

+ NVL (mt\_ebh\_cdma\_live, 0)

+ NVL (mt\_ebh\_evdo\_live, 0)

LIVE,

NVL (lu\_ebh\_reported, 0)

+ NVL (nt\_ebh\_cdma\_reported, 0)

+ NVL (nt\_ebh\_evdo\_reported, 0)

+ NVL (mt\_ebh\_cdma\_reported, 0)

+ NVL (mt\_ebh\_evdo\_reported, 0)

RPTD,

NVL (lu\_ebh\_discovered, 0)

+ NVL (nt\_ebh\_cdma\_discovered, 0)

+ NVL (nt\_ebh\_evdo\_discovered, 0)

+ NVL (mt\_ebh\_cdma\_discovered, 0)

+ NVL (mt\_ebh\_evdo\_discovered, 0)

DISCOVERED,

NVL (lu\_ebh\_comp, 0) lu\_ebh\_comp,

NVL (nt\_ebh\_cdma\_comp, 0) nt\_ebh\_cdma\_comp,

NVL (nt\_ebh\_evdo\_comp, 0) nt\_ebh\_evdo\_comp,

NVL (mt\_ebh\_cdma\_comp, 0) mt\_ebh\_cdma\_comp,

NVL (mt\_ebh\_evdo\_comp, 0) mt\_ebh\_evdo\_comp,

CASE

WHEN NVL (lu\_ebh\_reported, 0) -- dr is = 0 tot\_comp = null

+ NVL (nt\_ebh\_cdma\_reported, 0)

+ NVL (nt\_ebh\_evdo\_reported, 0)

+ NVL (mt\_ebh\_cdma\_reported, 0)

+ NVL (mt\_ebh\_evdo\_reported, 0) = 0

THEN

0

ELSE

ROUND (

( NVL (lu\_ebh\_live, 0)

+ NVL (nt\_ebh\_cdma\_live, 0)

+ NVL (nt\_ebh\_evdo\_live, 0)

+ NVL (mt\_ebh\_cdma\_live, 0)

+ NVL (mt\_ebh\_evdo\_live, 0))

/ ( NVL (lu\_ebh\_reported, 0)

+ NVL (nt\_ebh\_cdma\_reported, 0)

+ NVL (nt\_ebh\_evdo\_reported, 0)

+ NVL (mt\_ebh\_cdma\_reported, 0)

+ NVL (mt\_ebh\_evdo\_reported, 0))

\* 100,

2)

END

Total\_compliance

FROM regions r

LEFT OUTER JOIN

lucent l

ON ( r.sub\_market = l.sub\_market

AND l.sub\_market != 'NNO'

AND r.market = l.market

AND r.territory = l.territory)

OR ( r.sub\_market IS NULL

AND l.sub\_market IS NULL

AND r.market = l.market

AND l.market != 'NNO'

AND r.territory = l.territory)

OR ( r.sub\_market IS NULL

AND l.sub\_market IS NULL

AND r.market IS NULL

AND l.market IS NULL

AND r.territory = l.territory)

OR (l.territory IS NULL AND r.territory = 'grand Total')

LEFT OUTER JOIN

nt\_cdma ntc

ON ( r.sub\_market = ntc.sub\_market

AND ntc.sub\_market != 'NNO'

AND r.market = ntc.market

AND r.territory = ntc.territory)

OR ( r.sub\_market IS NULL

AND ntc.sub\_market IS NULL

AND r.market = ntc.market

AND ntc.market != 'NNO'

AND r.territory = ntc.territory)

OR ( r.sub\_market IS NULL

AND ntc.sub\_market IS NULL

AND r.market IS NULL

AND ntc.market IS NULL

AND r.territory = ntc.territory)

OR (ntc.territory IS NULL AND r.territory = 'grand Total')

LEFT OUTER JOIN

nt\_evdo nte

ON ( r.sub\_market = nte.sub\_market

AND nte.sub\_market != 'NNO'

AND r.market = nte.market

AND r.territory = nte.territory)

OR ( r.sub\_market IS NULL

AND nte.sub\_market IS NULL

AND r.market = nte.market

AND nte.market != 'NNO'

AND r.territory = nte.territory)

OR ( r.sub\_market IS NULL

AND nte.sub\_market IS NULL

AND r.market IS NULL

AND nte.market IS NULL

AND r.territory = nte.territory)

OR (nte.territory IS NULL AND r.territory = 'grand Total')

LEFT OUTER JOIN

mt\_cdma mtc

ON ( r.sub\_market = mtc.sub\_market

AND mtc.sub\_market != 'NNO'

AND r.market = mtc.market

AND r.territory = mtc.territory)

OR ( r.sub\_market IS NULL

AND mtc.sub\_market IS NULL

AND r.market = mtc.market

AND mtc.market != 'NNO'

AND r.territory = mtc.territory)

OR ( r.sub\_market IS NULL

AND mtc.sub\_market IS NULL

AND r.market IS NULL

AND mtc.market IS NULL

AND r.territory = mtc.territory)

OR (mtc.territory IS NULL AND r.territory = 'grand Total')

LEFT OUTER JOIN

mt\_evdo mte

ON ( r.sub\_market = mte.sub\_market

AND mte.sub\_market != 'NNO'

AND r.market = mte.market

AND r.territory = mte.territory)

OR ( r.sub\_market IS NULL

AND mte.sub\_market IS NULL

AND r.market = mte.market

AND mte.market != 'NNO'

AND r.territory = mte.territory)

OR ( r.sub\_market IS NULL

AND mte.sub\_market IS NULL

AND r.market IS NULL

AND mte.market IS NULL

AND r.territory = mte.territory)

OR (mte.territory IS NULL AND r.territory = 'grand Total')

ORDER BY territory, market, sub\_market

;

--------------------------------------------------------

-- DDL for View EHEALTH\_CSR\_LIST

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."EHEALTH\_CSR\_LIST" ("HOSTNAME", "IP", "EXTRACT\_DATE", "VENDOR") AS

SELECT SC.DEVICE\_NAME hostname,

SC.DEVICE\_IP ip,

SC.SYS\_CREATION\_DATE extract\_date,

---SC.ERROR\_CODE ERROR\_CODE,---------COL NOT THERE ERROR\_CODE

SC.VENDOR vendor

FROM sevone\_csr sc

;

--------------------------------------------------------

-- DDL for View ENODEB\_MKT\_V

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."ENODEB\_MKT\_V" ("MARKET\_ID", "LTE\_MARKET\_NAME", "CLLI", "LEAF\_DOMAIN\_ID", "TERRITORY", "MARKET\_TERRITORY", "SUB\_MARKET", "NODE", "NODE6", "S1MME\_IPADDRESS", "CONNECTION\_STATUS") AS

WITH mkt

AS (SELECT mpen.market\_id,

mpen.lte\_market\_name,

cdm.clli,

leaf\_domain\_id,

territory,

market\_territory,

sub\_market

FROM ng\_reports.market\_prefix\_enb\_naming mpen,

ng\_reports.clli\_domain\_map cdm,

ng\_reports.domains\_leaf\_reporting dlr

WHERE SUBSTR (mpen.clli, 1, 8) = cdm.clli

AND cdm.leaf\_domain\_id = dlr.domain\_id),

eric

as

(SELECT node\_name node,

ip\_address s1mme\_ipaddress,

NULL connection\_status, --satya said use null for status

SUBSTR (node\_name, 1, 6) enb\_number,

SUBSTR (node\_name, 0, 3) mkt\_code, 1 as ord

FROM ng\_reports.ericsson\_enm\_wk

UNION

SELECT node,

ip s1mme\_ipaddress,

connect\_status connection\_status,

SUBSTR (node, 1, 6) enb\_number,

SUBSTR (node, 0, 3) mkt\_code, 2 as ord

FROM ng\_reports.ericsson\_enb\_wk),

eric\_ord

as (select enb\_number, min(ord) as ord

from eric

group by enb\_number),

eric1

as (select e.node, e.s1mme\_ipaddress, e.connection\_status,

e.enb\_number, e.mkt\_code

from eric e, eric\_ord eo

where e.enb\_number = eo.enb\_number

and e.ord = eo.ord),

enb

AS (SELECT node,

s1mme\_ipaddress,

connection\_status,

enb\_number,

mkt\_code

FROM eric1

UNION

SELECT site\_id node,

ipv6\_addr s1mme\_ipaddress,

status connection\_status,

SUBSTR (site\_id, 1, 6) enb\_number,

SUBSTR (site\_id, 0, 3) mkt\_code

FROM ng\_reports.sam\_enb\_wk)

SELECT DISTINCT mkt.market\_id,

mkt.lte\_market\_name,

mkt.clli,

mkt.leaf\_domain\_id,

mkt.territory,

mkt.market\_territory,

mkt.sub\_market,

enb.node,

substr(enb.node, 1,6) node6,

enb.s1mme\_ipaddress,

enb.connection\_status

FROM enb JOIN mkt ON enb.mkt\_code = mkt.market\_id

;

--------------------------------------------------------

-- DDL for View NAURU\_NTLS\_SITE\_MAP

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."NAURU\_NTLS\_SITE\_MAP" ("NT\_SITE\_ID", "NAURU\_ID", "FR\_REF\_KEY\_VALUE") AS

select SA.SITE\_REFERENCE\_ID NTLS\_SITE\_ID, SA.SITE\_VALUE as nauru\_id, SA.FR\_REF\_KEY\_VALUE

from ng\_reports.SITE\_ATTRIBUTES sa --where s.SITE\_REFERENCE\_ID = sa.SITE\_REFERENCE\_ID --and sa.SITE\_GROUP\_NAME='Emergency Contacts'

where sa.SITE\_NAME = 'NWF Site ID (NAURU)'

;

--------------------------------------------------------

-- DDL for View NGMLS\_ALL\_NG\_AUDIT\_V

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."NGMLS\_ALL\_NG\_AUDIT\_V" ("EQP\_REFERENCE\_ID", "EQP\_NAME", "ALTERNATE\_NAME", "CONTAINER", "EQP\_MODEL", "EQP\_TYPE", "SHELF\_TYPE", "FUNCTIONAL\_TYPE", "EQP\_VENDOR", "EQP\_MANUFACTURER", "PART\_NUM", "SERIAL\_NUMBER", "MATERIAL\_ID", "EQUIPMENT\_SPEC\_REF\_ID", "EQUIPMENT\_SPEC\_NAME", "INV\_STATUS", "EQP\_CLEI", "SITE\_REFERENCE\_ID", "LOCATION\_CLLI", "HI\_LOCATION\_TYPE", "HI\_LOCATION\_REFERENCE", "LOW\_LOCATION\_TYPE", "LOW\_LOCATION\_REFERENCE", "LINE\_UP", "FRAME", "PHYSICAL\_SHELF\_POSITION", "LOGICAL\_SHELF", "NO\_OF\_SLOTS", "WIRE\_CENTER", "IS\_MULTI\_TID\_SHELF", "PARENT\_EQP\_TYPE", "PARENT\_EQP\_REFERENCE\_ID", "PARENT\_SHELF\_REFERENCE\_ID", "BAR\_CODE", "BATCH\_NUMBER", "COMMENTS", "CUSTOMER\_REFERENCE\_ID", "EQP\_HEIGHT", "EQP\_WIDTH", "EQP\_DEPTH", "DIST\_FROM\_Z", "DIST\_FROM\_Y", "DIST\_FROM\_X", "DUE\_DATE", "INEFFECT\_DATE", "INSTALLED\_DATE", "DECOMMISION\_DATE", "SCHEDULED\_DATE", "FIELD\_ID", "POINT\_CODE", "TID\_LOGICAL", "TID\_PHYSICAL", "HARDWARE\_REVISION", "SOFTWARE\_REVISION", "EMS", "MGMT\_IP\_ADDRESS", "AID", "IPV4\_ADDRESS", "IPV6\_ADDRESS", "LAST\_MODIFIED\_TIME\_STAMP", "LAST\_MODIFIED\_BY", "ORDER\_DATE", "ORDER\_NUMBER", "ASSET\_LIFE", "EQP\_PURCHASE\_DATE", "EQP\_PURCHASE\_PRICE", "EQP\_SOURCE", "FR\_REF\_KEY\_NAME", "FR\_REF\_KEY\_VALUE", "ASSET\_OWNER", "SUB\_LOCATION", "CUSTOMER\_NAME", "NETWORK\_TYPE", "NETWORK\_DOMAIN", "PROJECT\_REFERENCE\_ID", "INSTANCE\_TYPE", "PHYSICAL\_EQ\_REFERENCE\_ID", "MGMT\_TELNET\_PORT", "MGMT\_SSH\_PORT") AS

SELECT

"EQP\_REFERENCE\_ID","EQP\_NAME","ALTERNATE\_NAME","CONTAINER","EQP\_MODEL","EQP\_TYPE","SHELF\_TYPE","FUNCTIONAL\_TYPE","EQP\_VENDOR","EQP\_MANUFACTURER","PART\_NUM","SERIAL\_NUMBER","MATERIAL\_ID","EQUIPMENT\_SPEC\_REF\_ID","EQUIPMENT\_SPEC\_NAME","INV\_STATUS","EQP\_CLEI","SITE\_REFERENCE\_ID","LOCATION\_CLLI","HI\_LOCATION\_TYPE","HI\_LOCATION\_REFERENCE","LOW\_LOCATION\_TYPE","LOW\_LOCATION\_REFERENCE","LINE\_UP","FRAME","PHYSICAL\_SHELF\_POSITION","LOGICAL\_SHELF","NO\_OF\_SLOTS","WIRE\_CENTER","IS\_MULTI\_TID\_SHELF","PARENT\_EQP\_TYPE","PARENT\_EQP\_REFERENCE\_ID","PARENT\_SHELF\_REFERENCE\_ID","BAR\_CODE","BATCH\_NUMBER","COMMENTS","CUSTOMER\_REFERENCE\_ID","EQP\_HEIGHT","EQP\_WIDTH","EQP\_DEPTH","DIST\_FROM\_Z","DIST\_FROM\_Y","DIST\_FROM\_X","DUE\_DATE","INEFFECT\_DATE","INSTALLED\_DATE","DECOMMISION\_DATE","SCHEDULED\_DATE","FIELD\_ID","POINT\_CODE","TID\_LOGICAL","TID\_PHYSICAL","HARDWARE\_REVISION","SOFTWARE\_REVISION","EMS","MGMT\_IP\_ADDRESS","AID","IPV4\_ADDRESS","IPV6\_ADDRESS","LAST\_MODIFIED\_TIME\_STAMP","LAST\_MODIFIED\_BY","ORDER\_DATE","ORDER\_NUMBER","ASSET\_LIFE","EQP\_PURCHASE\_DATE","EQP\_PURCHASE\_PRICE","EQP\_SOURCE","FR\_REF\_KEY\_NAME","FR\_REF\_KEY\_VALUE","ASSET\_OWNER","SUB\_LOCATION","CUSTOMER\_NAME","NETWORK\_TYPE","NETWORK\_DOMAIN","PROJECT\_REFERENCE\_ID","INSTANCE\_TYPE","PHYSICAL\_EQ\_REFERENCE\_ID","MGMT\_TELNET\_PORT","MGMT\_SSH\_PORT"

FROM

ng\_equipment

;

--------------------------------------------------------

-- DDL for View NG\_ENB\_PARSED

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."NG\_ENB\_PARSED" ("SITE\_NAME", "EQP\_NAME", "CONTAINER", "LEAF\_DOMAIN\_NAME", "SITE\_REFERENCE\_ID", "EQP\_REFERENCE\_ID", "LEAF\_DOMAIN\_INST\_ID", "EQP\_VENDOR", "EQP\_MODEL", "INV\_STATUS", "EQP\_TYPE") AS

SELECT

si.site\_name,

ei.EQP\_NAME,

ei.CONTAINER,

dlr.domain\_name leaf\_domain\_name,

si.SITE\_REFERENCE\_ID,

ei.EQP\_REFERENCE\_ID,

dlr.domain\_id leaf\_domain\_inst\_id,

ei.eqp\_vendor,

ei.eqp\_model,

ei.inv\_status,

ei.eqp\_type

FROM

ng\_reports.equipment ei

JOIN ng\_reports.site si ON si.site\_reference\_id = ei.site\_reference\_id

JOIN NG\_REPORTS.equipment\_domain\_map edm ON edm.eqp\_reference\_id = ei.eqp\_reference\_id

JOIN domains\_leaf\_reporting dlr ON dlr.domain\_id = edm.domain\_id

WHERE

eqp\_type IN (

'BASEBAND',

'ENODEB',

'MICROCELL'

)

AND dlr.domain\_name <> 'NNO\_DOMAIN'

;

--------------------------------------------------------

-- DDL for View NTLS\_CSR\_PARSED\_V

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."NTLS\_CSR\_PARSED\_V" ("EQP\_REFERENCE\_ID", "EQP\_NAME", "CSR\_DEVICE\_NAME", "PARSE\_STATUS", "CSR\_LIVE\_IN\_XNG") AS

select eqp\_reference\_id, eqp\_name, csr\_device\_name,

case

when csr\_device\_name is null then 'Invalid CSR Name'

when csr\_device\_name is not null and not REGEXP\_LIKE(eqp\_name, '^([[:alnum:]]{1,4}|)[[:alnum:]]+-([[:alnum:]]{8}T[[:digit:]]A-P-(AL|CI)-(G[[:digit:]]{3}|[[:digit:]]{4}|[[:digit:]]{6})-(S|E|H|R|W|I|D|[[:digit:]]{1})(I|i|[[:digit:]]{1})(-[[:digit:]]{2})?)[[:space:]]\*(|([[:space:]]\*\[.+[[:space:]]\*\][[:space:]]\*))$')

then 'Xng CSR Name does not match Standard'

else null

end parse\_status,

case when inv\_status = 'LIVE' then 'Y'

else 'N' end csr\_live\_in\_xng

from (select eqp\_reference\_id, eqp\_name, inv\_status,

REGEXP\_SUBSTR(eqp\_name, '([[:alnum:]]{8}T[[:digit:]]A-P-(AL|CI)-(G[[:digit:]]{3}|[[:digit:]]{4}|[[:digit:]]{6})-(S|E|H|R|W|I|D|[[:digit:]]{1})(I|i|[[:digit:]]{1})(-[[:digit:]]{2})?)') csr\_device\_name

from ng\_reports.equipment

where eqp\_type = 'CSR'

and container='SHELF')

;

--------------------------------------------------------

-- DDL for View NTLS\_DL\_ENB\_INFO

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."NTLS\_DL\_ENB\_INFO" ("MARKET", "SUB\_MARKET", "MARKET\_ID", "NODE6", "NODE", "LTE\_MARKET\_NAME", "CONNECTION\_STATUS", "S1MME\_IPADDRESS", "DEVICE\_IP", "LEAF\_DOMAIN\_NAME", "NTLS\_EQP\_ID", "EQUIP\_NAME", "EQUIP\_STATUS", "EQP\_VENDOR", "EQP\_MODEL", "NTLS\_SITE\_ID", "SITE\_NAME", "CLLI", "LATITUDE", "LONGITUDE") AS

select distinct em.market\_territory market, em.sub\_market, em.market\_id, em.node6, em.node, em.lte\_market\_name,

em.connection\_status, em.s1mme\_ipaddress, em.s1mme\_ipaddress device\_ip, nep.leaf\_domain\_name,

nep.eqp\_reference\_id ntls\_eqp\_id, nep.eqp\_name equip\_name, nep.inv\_status equip\_status, nep.eqp\_vendor, nep.eqp\_model,

nep.site\_reference\_id ntls\_site\_id, nep.site\_name, nep.clli, nep.latitude, nep.longitude

from ng\_reports.enodeb\_mkt\_v em, ng\_reports.ntls\_enb\_parsed\_v nep

where nep.enb\_number = em.node6

;

--------------------------------------------------------

-- DDL for View NTLS\_ENB\_PARSED\_V

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."NTLS\_ENB\_PARSED\_V" ("DOMAIN\_ID", "LEAF\_DOMAIN\_NAME", "SITE\_REFERENCE\_ID", "SITE\_NAME", "CLLI", "LATITUDE", "LONGITUDE", "EQP\_REFERENCE\_ID", "EQP\_NAME", "EQP\_VENDOR", "EQP\_MODEL", "INV\_STATUS", "EQP\_TYPE", "ENB\_NUMBER", "PARSE\_STATUS") AS

with enb as (SELECT domain\_id,

domain\_name,

site\_reference\_id,

site\_name,

clli,

latitude,

longitude,

eqp\_reference\_id,

eqp\_name,

eqp\_vendor,

eqp\_model,

inv\_status,

eqp\_type,

enb\_number,

CASE

WHEN enb\_number IS NULL

THEN

'NTLS Equipment Name not Parsable'

WHEN enb\_number <> SUBSTR (eqp\_name, 1, 6)

THEN

'Xng Equipment Name does not match Standard'

ELSE

NULL

END parse\_status,

CASE

WHEN enb\_number IS NULL THEN 1

WHEN enb\_number <> SUBSTR (eqp\_name, 1, 6) THEN 2

ELSE 3

END parse\_stat\_ord

FROM (SELECT dlr.domain\_id,

dlr.domain\_name,

s.site\_reference\_id,

s.site\_name,

s.clli,

s.latitude,

s.longitude,

e.eqp\_reference\_id,

e.eqp\_name,

e.eqp\_vendor,

e.eqp\_model,

e.inv\_status,

eqp\_type,

REGEXP\_SUBSTR (

REGEXP\_SUBSTR (eqp\_name, '(^|\D)\d{6}(\D|$)'),

'\d{6}')

AS enb\_number

FROM ng\_reports.equipment e

JOIN ng\_reports.site s

ON e.site\_reference\_id = s.site\_reference\_id

JOIN ng\_reports.equipment\_domain\_map edm

ON edm.eqp\_reference\_id = e.eqp\_reference\_id

JOIN ng\_reports.domains\_leaf\_reporting dlr

ON dlr.domain\_id = edm.domain\_id

WHERE e.eqp\_type IN ('BASEBAND', 'ENODEB', 'MICROCELL')

AND dlr.domain\_name <> 'NNO\_DOMAIN')),

enb\_max as (select enb\_number, max(parse\_stat\_ord) parse\_stat\_ord

from enb

group by enb\_number)

select domain\_id,

domain\_name leaf\_domain\_name,

site\_reference\_id,

site\_name,

clli,

latitude,

longitude,

eqp\_reference\_id,

eqp\_name,

eqp\_vendor,

eqp\_model,

inv\_status,

eqp\_type,

enb.enb\_number,

parse\_status

from enb, enb\_max

where enb.enb\_number = enb\_max.enb\_number

and enb.parse\_stat\_ord = enb\_max.parse\_stat\_ord

;

--------------------------------------------------------

-- DDL for View SEGMENT\_BTOD\_VW

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."SEGMENT\_BTOD\_VW" ("NTLS\_SEG\_ID", "NTLS\_SEG\_NAME", "NTLS\_SEG\_NAME\_STRIP", "NTLS\_TYPE", "NTLS\_VENDOR\_CK", "NTLS\_VENDOR", "NTLS\_BANDWIDTH", "NTLS\_STATUS", "NTLS\_CURR\_TRAIL\_ID", "NTLS\_NEXT\_TRAIL\_ID", "NTLS\_BAN", "NTLS\_BAN\_CK") AS

SELECT segment\_id AS ntls\_seg\_id,

name AS ntls\_seg\_name,

strip\_segment\_name (name) AS ntls\_seg\_name\_strip,

TYPE AS ntls\_type,

NVL (SUBSTR (vendor, 1, INSTR (vendor, ' ') - 1), vendor)

AS ntls\_vendor\_ck,

vendor AS ntls\_vendor,

bandwidth AS ntls\_bandwidth,

status AS ntls\_status,

current\_trail\_id AS ntls\_curr\_trail\_id,

next\_trail\_id AS ntls\_next\_trail\_id,

billing\_code AS ntls\_ban,

REPLACE (

TRANSLATE (NVL (billing\_code, '0'),

'-/,.()[]\*$;?#&~:=`<>\^@!"' || CHR (160),

' '),

' ',

'')

AS ntls\_ban\_ck

FROM ng\_topology.trail\_segment

;

--------------------------------------------------------

-- DDL for View VZW\_VSM\_NG\_XREF\_V

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."VZW\_VSM\_NG\_XREF\_V" ("AREA", "REGION", "VSM\_DEVICE\_NAME", "VSM\_CLASS", "ALLTEL\_DO\_NOT\_REMOVE", "AS\_OF\_DATE", "DOMAIN\_NAME", "LEAF\_DOMAIN\_INST\_ID", "PARENT\_DOMAIN\_ID") AS

SELECT

dlr.territory area,

dlr.sub\_market region,

vddm.vsm\_device\_name,

vddm.vsm\_class,

vddm.alltel\_do\_not\_remove,

vddm.as\_of\_date,

dlr.domain\_name,

leaf\_domain\_inst\_id,

parent\_domain\_id

FROM

domains\_leaf\_reporting dlr

JOIN vsm\_devices\_domain\_map vddm ON dlr.domain\_id = vddm.leaf\_domain\_inst\_id

;

--------------------------------------------------------

-- DDL for View VZW\_VSM\_NTLS\_XREF

--------------------------------------------------------

CREATE OR REPLACE FORCE EDITIONABLE VIEW "NG\_REPORTS"."VZW\_VSM\_NTLS\_XREF" ("TERRITORY", "MARKET\_TERRITORY", "SUB\_MARKET", "VSM\_DEVICE\_NAME", "VSM\_CLASS", "ALLTEL\_DO\_NOT\_REMOVE", "AS\_OF\_DATE", "LEAF\_DOMAIN\_NAME", "LEAF\_DOMAIN\_INST\_ID", "PARENT\_DOMAIN\_INST\_ID", "CITY", "COUNTY", "DESCR", "DOMAIN\_INST\_ID", "DOMAIN\_NAME", "EQUIP\_INST\_ID", "MARKET\_INST\_ID", "OLD\_SITE\_INST\_ID", "SITE\_INST\_ID", "SITE\_HUM\_ID", "STATE", "STATUS", "VENDOR", "VSM\_CITY", "VSM\_STATE", "VSM\_MOD\_DEVICE\_NAME", "NTLS\_DEVICE\_NAME", "NTLS\_MOD\_DEVICE\_NAME", "SCHEMA") AS

SELECT dlr.territory,

DLR.MARKET\_TERRITORY market,

dlr.sub\_market,

vddm.vsm\_device\_name,

vddm.vsm\_class,

vddm.alltel\_do\_not\_remove,

vddm.as\_of\_date,

dlr.domain\_name,

leaf\_domain\_inst\_id,

parent\_domain\_id,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL,

NULL

FROM domains\_leaf\_reporting dlr

JOIN vsm\_devices\_domain\_map vddm

ON dlr.domain\_id = vddm.leaf\_domain\_inst\_id

;

--------------------------------------------------------

-- DDL for Index SITE\_IDX\_1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SITE\_IDX\_1" ON "NG\_REPORTS"."SITE" ("SITE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NTLS\_CSR\_PARSED\_WK\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NTLS\_CSR\_PARSED\_WK\_PK" ON "NG\_REPORTS"."NTLS\_CSR\_PARSED\_WK" ("EQP\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index FK1\_DIR\_TP\_TYPE

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."FK1\_DIR\_TP\_TYPE" ON "NG\_REPORTS"."PORT" ("TP\_TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NFE\_ADM\_EQP\_REFERENCE\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NFE\_ADM\_EQP\_REFERENCE\_IDX" ON "NG\_REPORTS"."NETSMART\_ADM\_FUJITSU\_EQUIPMENT" ("EQP\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIENA\_6500\_DISC\_METADATA\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."CIENA\_6500\_DISC\_METADATA\_PK" ON "NG\_REPORTS"."CIENA\_6500\_DISC\_METADATA" ("COMMAND\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NFE\_EQP\_REFERENCE\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NFE\_EQP\_REFERENCE\_IDX" ON "NG\_REPORTS"."NETSMART\_FUJITSU\_EQUIPMENT" ("EQP\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NIS\_EXTRACTOR\_GRAPH\_1IX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH\_1IX" ON "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" ("RULE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index LNVXA\_PATHINST\_STATUS\_TYPE\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."LNVXA\_PATHINST\_STATUS\_TYPE\_IDX" ON "NG\_REPORTS"."LUCENT\_NE\_VS\_NG\_AUDIT" ("TRAIL\_ID", "NG\_PATH\_STATUS", "NG\_PATH\_TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIENA\_6500\_NE\_EQUIP\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP\_PK" ON "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP" ("ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NIS\_EXTRACTOR\_PARAM\_1IX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM\_1IX" ON "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" ("PARAM\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NS\_NG\_PORTS\_IDX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NS\_NG\_PORTS\_IDX1" ON "NG\_REPORTS"."NETSMART\_NG\_PORTS" ("TID", "PORT\_NAME", "SLOT\_NAME"||'-'||"PORT\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index EPA\_NEXT\_PATH\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."EPA\_NEXT\_PATH\_IDX" ON "NG\_REPORTS"."PORT" ("NEXT\_PATH\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SAM\_CSR\_VLAN\_NXA\_NE\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SAM\_CSR\_VLAN\_NXA\_NE\_IDX" ON "NG\_REPORTS"."SAM\_CSR\_VLAN\_AUDIT" ("CSR\_VENDOR", "NE\_HOSTNAME", "NE\_VLAN\_NUMBER", "MATCH\_CODE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index VZW\_NT\_EVDO\_CELL\_PATHS\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."VZW\_NT\_EVDO\_CELL\_PATHS\_IDX" ON "NG\_REPORTS"."VZW\_NT\_EVDO\_CELL\_PATHS" ("DOM\_NODE\_IP", "SPAN\_NUMBER", "BANDWIDTH")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index VVDM\_UNQ\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."VVDM\_UNQ\_IDX" ON "NG\_REPORTS"."VSM\_DEVICES\_DOMAIN\_MAP" ("VSM\_DEVICE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NTLS\_CSR\_PARSED\_WK\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NTLS\_CSR\_PARSED\_WK\_IDX" ON "NG\_REPORTS"."NTLS\_CSR\_PARSED\_WK" ("CSR\_DEVICE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_SNCNE\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NETSMART\_SNCNE\_PK" ON "NG\_REPORTS"."NETSMART\_SNCNE" ("INST\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index EPA\_CIRC\_PATH\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."EPA\_CIRC\_PATH\_IDX" ON "NG\_REPORTS"."PORT" ("CKT\_PATH\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index PORT\_NUM\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."PORT\_NUM\_IDX" ON "NG\_REPORTS"."PORT" ("PORT\_NUMBER")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index FK1\_DIR\_PORT\_STATUS

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."FK1\_DIR\_PORT\_STATUS" ON "NG\_REPORTS"."PORT" ("PORT\_STATUS")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INC\_MSPP\_NE\_U01

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."INC\_MSPP\_NE\_U01" ON "NG\_REPORTS"."INC\_MSPP\_NE" ("TID", "GNE\_TID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX1\_P\_EQP\_REFERENCE\_ID

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."IDX1\_P\_EQP\_REFERENCE\_ID" ON "NG\_REPORTS"."PORT" ("EQP\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX\_SITE\_ATTRIBUTE\_ID

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."IDX\_SITE\_ATTRIBUTE\_ID" ON "NG\_REPORTS"."SITE\_ATTRIBUTES" ("SITE\_ATTRIBUTES\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CARD\_PORTNUM\_NAME\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."CARD\_PORTNUM\_NAME\_IDX" ON "NG\_REPORTS"."PORT" ("CARD\_REFERENCE\_ID", "PORT\_NUMBER", "BANDWIDTH\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index PORT\_NEXT\_CHAN\_ID\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."PORT\_NEXT\_CHAN\_ID\_IDX" ON "NG\_REPORTS"."PORT" ("NEXT\_CHANNEL\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NIS\_EXTRACTOR\_GRAPH\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH\_PK" ON "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" ("NODE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index MNVXA\_IDX\_1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."MNVXA\_IDX\_1" ON "NG\_REPORTS"."MOTO\_CDMA\_NE\_VS\_NG\_AUDIT" ("MTX", "NE\_OMCR", "NE\_BTS\_NUMBER", "NE\_CLUSTER\_NUMBER", "NE\_AG\_SPAN\_NUMBER")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index VSM\_OMCR\_UNQ\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."VSM\_OMCR\_UNQ\_IDX" ON "NG\_REPORTS"."MT\_VSM\_SANE\_RAW\_OMCR\_NAME" ("VSM\_DEVICE\_NAME\_OMCR")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INC\_MSPP\_NE\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."INC\_MSPP\_NE\_PK" ON "NG\_REPORTS"."INC\_MSPP\_NE" ("NEID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NORTEL\_NE\_VS\_NG\_AUDIT\_IDX4

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT\_IDX4" ON "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT" ("NE\_DOM\_IP", "NE\_STATUS", "NE\_DCG\_NUMBER", "NE\_SLOT\_NUMBER", "NE\_SPAN\_NUMBER")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIENA\_6500\_NE\_KEY\_VALUE\_1IX

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."CIENA\_6500\_NE\_KEY\_VALUE\_1IX" ON "NG\_REPORTS"."CIENA\_6500\_NE\_KEY\_VALUE" ("PARSED\_INST\_ID", "KEY")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CLLI\_DOMAIN\_MAP\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."CLLI\_DOMAIN\_MAP\_PK" ON "NG\_REPORTS"."CLLI\_DOMAINS\_MAP" ("CLLI", "LEAF\_DOMAIN\_INST\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX1" ON "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" ("AREA", "REGION", "ACTION\_TYPE", "COMMIT\_DATE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index FK\_PHYSICAL\_PORT\_REFERENCE\_ID

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."FK\_PHYSICAL\_PORT\_REFERENCE\_ID" ON "NG\_REPORTS"."PORT" ("PHYSICAL\_PORT\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX4

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX4" ON "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" ("BANDWIDTH", "SEGMENT\_TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index TEMP\_NT\_SITERRA\_ENB\_IX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."TEMP\_NT\_SITERRA\_ENB\_IX" ON "NG\_REPORTS"."TEMP\_NT\_SITERRA\_ENB" ("SITE\_INST\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_SNCNE\_WK\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NETSMART\_SNCNE\_WK\_PK" ON "NG\_REPORTS"."NETSMART\_SNCNE\_WK" ("TID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX1\_Z\_WIRED\_PORT\_REF\_ID

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."IDX1\_Z\_WIRED\_PORT\_REF\_ID" ON "NG\_REPORTS"."PORT" ("Z\_WIRED\_PORT\_REF\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index BSM\_MTX\_MAP\_IDX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."BSM\_MTX\_MAP\_IDX1" ON "NG\_REPORTS"."BSM\_MTX\_MAP" ("VSM\_DEVICE\_NAME\_BSM")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index MOTO\_CDMA\_EI\_OMCR\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."MOTO\_CDMA\_EI\_OMCR\_IDX" ON "NG\_REPORTS"."MOTO\_CDMA\_ETHERNET\_INV" ("VSM\_DEVICE\_NAME\_OMCR")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SEG\_MIS\_EST\_COST\_DOMAIN\_INDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SEG\_MIS\_EST\_COST\_DOMAIN\_INDX" ON "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST" ("DOMAIN")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE( INITIAL 8388608 NEXT 8388608 MINEXTENTS 1

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SITE\_IDX\_2

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SITE\_IDX\_2" ON "NG\_REPORTS"."SITE" ("TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NORTEL\_NE\_VS\_NG\_AUDIT\_IDX3

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT\_IDX3" ON "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT" ("VSM\_DEVICE\_NAME\_BSM", "NE\_DCG\_NUMBER", "NE\_SLOT\_NUMBER", "NE\_SPAN\_NUMBER", "VSM\_DEVICE\_NAME\_MTX")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SITE\_IDX\_4

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SITE\_IDX\_4" ON "NG\_REPORTS"."SITE" ("STATUS")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SAM\_CSR\_VLAN\_CSR\_NE\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SAM\_CSR\_VLAN\_CSR\_NE\_IDX" ON "NG\_REPORTS"."SAM\_CSR\_VLAN\_AUDIT" ("CSR\_DEVICE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_EQPT\_INDEX1

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NETSMART\_EQPT\_INDEX1" ON "NG\_REPORTS"."NETSMART\_EQPT" ("SNCNE\_ID", "COMPONENT\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index PRIMARY

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."PRIMARY" ON "NG\_REPORTS"."PORT" ("PORT\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index DIR\_INV\_DEF\_ATTRIBUTE\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."DIR\_INV\_DEF\_ATTRIBUTE\_PK" ON "NG\_REPORTS"."DIR\_INV\_DEF\_ATTRIBUTE" ("INV\_DEF\_ATTR\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INDX\_LAST\_TEST\_DETAILS

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."INDX\_LAST\_TEST\_DETAILS" ON "NG\_REPORTS"."NOCC\_LAST\_TEST\_DETAILS" ("RUN\_DATE" DESC, "DOMAIN\_ID", "SITE\_INST\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index TEMP\_NT\_SITERRA\_CSR\_IX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."TEMP\_NT\_SITERRA\_CSR\_IX" ON "NG\_REPORTS"."TEMP\_NT\_SITERRA\_CSR" ("SITE\_INST\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index P\_PORT\_ID

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."P\_PORT\_ID" ON "NG\_REPORTS"."PORT" ("FR\_REF\_KEY\_VALUE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index TEST\_HEAD\_XREF\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."TEST\_HEAD\_XREF\_PK" ON "NG\_REPORTS"."TEST\_HEAD\_XREF" ("TEST\_HEAD\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CARD\_REFERENCE\_ID

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."CARD\_REFERENCE\_ID" ON "NG\_REPORTS"."PORT" ("CARD\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SITE\_IDX\_5

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SITE\_IDX\_5" ON "NG\_REPORTS"."SITE" ("LATITUDE", "LONGITUDE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SAM\_ENB\_WK\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SAM\_ENB\_WK\_IDX" ON "NG\_REPORTS"."SAM\_ENB\_WK" ("SITE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index DIR\_PORT\_INDEX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."DIR\_PORT\_INDEX" ON "NG\_REPORTS"."PORT" ("DIRECTION")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NT\_SITERRA\_SEG\_EXTRACT\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT\_PK" ON "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "CIRC\_INST\_ID", "NAURU\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index LUCENT\_NE\_VS\_XNG\_AUDT\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."LUCENT\_NE\_VS\_XNG\_AUDT\_IDX" ON "NG\_REPORTS"."LUCENT\_NE\_VS\_NG\_AUDIT" ("SYS\_ID", "ECP\_SID", "NE\_CELL\_NUM", "NE\_URC\_NUM", "NE\_DS1\_NUM")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NT\_SITERRA\_SITE\_EXTRACT\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT\_PK" ON "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT" ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "NAURU\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIENA\_6500\_NE\_EQUIP\_U01

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP\_U01" ON "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP" ("DEVICE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SANE\_OMCR\_UNQ\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SANE\_OMCR\_UNQ\_IDX" ON "NG\_REPORTS"."MT\_VSM\_SANE\_RAW\_OMCR\_NAME" ("SANE\_NAME\_OMCR")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index OBS\_SANE\_BTS\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."OBS\_SANE\_BTS\_IDX" ON "NG\_REPORTS"."MT\_OMCR\_BTS\_STATUS" ("SANE\_NAME\_OMCR", "BTS\_NUMBER")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_XCONNECT\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NETSMART\_XCONNECT\_PK" ON "NG\_REPORTS"."NETSMART\_XCONNECT" ("INST\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index VZW\_NT\_VOICE\_CELL\_PATHS\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."VZW\_NT\_VOICE\_CELL\_PATHS\_IDX" ON "NG\_REPORTS"."VZW\_NT\_VOICE\_CELL\_PATHS" ("SWITCH\_ID", "DCG\_NUMBER", "SPAN\_NUMBER")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index RELATED\_PORT\_REF\_INDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."RELATED\_PORT\_REF\_INDX" ON "NG\_REPORTS"."PORT" ("RELATED\_PORT\_REF\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index EPA\_EQPID\_BW\_TYP\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."EPA\_EQPID\_BW\_TYP\_IDX" ON "NG\_REPORTS"."PORT" ("EQP\_REFERENCE\_ID", "TP\_TYPE", "BANDWIDTH\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_EQPT\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NETSMART\_EQPT\_PK" ON "NG\_REPORTS"."NETSMART\_EQPT" ("INST\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_SNCNE\_INDEX1

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NETSMART\_SNCNE\_INDEX1" ON "NG\_REPORTS"."NETSMART\_SNCNE" ("TID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index EPA\_SITE\_REF\_ID\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."EPA\_SITE\_REF\_ID\_IDX" ON "NG\_REPORTS"."PORT" ("SITE\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SEG\_MIS\_EST\_COST\_REGION\_INDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SEG\_MIS\_EST\_COST\_REGION\_INDX" ON "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST" ("REGION")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE( INITIAL 8388608 NEXT 8388608 MINEXTENTS 1

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SITE\_IDX\_3

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SITE\_IDX\_3" ON "NG\_REPORTS"."SITE" ("CLLI")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NORTEL\_NE\_VS\_NG\_AUDIT\_IDX2

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT\_IDX2" ON "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT" ("TERMINATION\_TYPE", "NE\_SPAN\_TYPE", "MATCH\_CODE", "MATCH\_STATUS", "NG\_PATH\_STATUS")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NIS\_EXTRACTOR\_PARAM\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM\_PK" ON "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" ("RULE\_ID", "PARAM\_INST\_ID", "PARAM\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index MOTO\_BTS\_LOCATION\_MAP\_INDEX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."MOTO\_BTS\_LOCATION\_MAP\_INDEX1" ON "NG\_REPORTS"."MOTO\_BTS\_LOCATION\_MAP" ("VSM\_DEVICE\_NAME\_OMCR", "BTS\_NUMBER", "BTS\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX1\_A\_WIRED\_PORT\_REF\_ID

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."IDX1\_A\_WIRED\_PORT\_REF\_ID" ON "NG\_REPORTS"."PORT" ("A\_WIRED\_PORT\_REF\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX3

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX3" ON "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" ("COMMIT\_DATE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index JUNIPER\_DEVICES\_WK\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."JUNIPER\_DEVICES\_WK\_PK" ON "NG\_REPORTS"."JUNIPER\_DEVICES\_WK" ("HOST\_NAME", "DEVICE\_IP", "DEVICE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NS\_ADM\_NG\_PORTS\_IDX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NS\_ADM\_NG\_PORTS\_IDX1" ON "NG\_REPORTS"."NETSMART\_ADM\_NG\_PORTS" ("TID", "PORT\_NAME", "SLOT\_NAME"||'-'||"PORT\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index EQP\_TYPE

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."EQP\_TYPE" ON "NG\_REPORTS"."EQUIPMENT" ("EQP\_TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIENA\_6500\_NE\_KEY\_VALUE\_WK\_1IX

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."CIENA\_6500\_NE\_KEY\_VALUE\_WK\_1IX" ON "NG\_REPORTS"."CIENA\_6500\_NE\_KEY\_VALUE\_WK" ("PARSED\_INST\_ID", "KEY")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NIS\_EXTRACTOR\_1IX

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NIS\_EXTRACTOR\_1IX" ON "NG\_REPORTS"."NIS\_EXTRACTOR" ("RULE\_NAME", "RULE\_TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NT\_SITERRA\_ANT\_EXTRACT\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT\_PK" ON "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "EQUIP\_INST\_ID", "NAURU\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SEGMENTS\_MISSING\_EST\_COST\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST\_PK" ON "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST" ("CIRC\_INST\_ID", "DOMAIN")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE( INITIAL 8388608 NEXT 8388608 MINEXTENTS 1

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INDX\_BANDWIDTH

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."INDX\_BANDWIDTH" ON "NG\_REPORTS"."PORT" ("BANDWIDTH\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX\_REF\_ID\_GR\_EQ\_ATTR

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."IDX\_REF\_ID\_GR\_EQ\_ATTR" ON "NG\_REPORTS"."EQUIPMENT\_ATTRIBUTES" ("EQP\_REFERENCE\_ID", "EQP\_GROUP\_NAME", "EQP\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SYS\_C00148741

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."SYS\_C00148741" ON "NG\_REPORTS"."CSR\_ISSUE" ("CSR\_ISSUE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX\_EQ\_ATTRIBUTE\_ID

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."IDX\_EQ\_ATTRIBUTE\_ID" ON "NG\_REPORTS"."EQUIPMENT\_ATTRIBUTES" ("EQP\_ATTRIBUTES\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX\_NHG\_CSR\_DEVICE

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."IDX\_NHG\_CSR\_DEVICE" ON "NG\_REPORTS"."CSR\_DEVICE\_AUDIT\_ISSUES" ("CSR\_DEVICE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NG\_EQUIPMENT\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NG\_EQUIPMENT\_PK" ON "NG\_REPORTS"."NG\_EQUIPMENT" ("EQP\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX2

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL\_IDX2" ON "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" ("SEGMENT\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NIS\_EXTRACTOR\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NIS\_EXTRACTOR\_PK" ON "NG\_REPORTS"."NIS\_EXTRACTOR" ("RULE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_IPADDR\_WK\_IDX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NETSMART\_IPADDR\_WK\_IDX1" ON "NG\_REPORTS"."NETSMART\_IPADDR\_WK" ("TID", "AID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX\_CIENA\_PARSED\_EQ\_ID

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."IDX\_CIENA\_PARSED\_EQ\_ID" ON "NG\_REPORTS"."CIENA\_6500\_NE\_PARSED\_DATA\_WK" ("EQ\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index OBRGC\_SANE\_BTS\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."OBRGC\_SANE\_BTS\_IDX" ON "NG\_REPORTS"."MT\_OMCR\_BTS\_RTR\_GRP\_CONF" ("SANE\_NAME\_OMCR", "BTS\_NUMBER")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SYS\_C00148742

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."SYS\_C00148742" ON "NG\_REPORTS"."CSR\_ISSUE" ("DESCRIPTION")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index EQUIP\_PORT\_PARENT\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."EQUIP\_PORT\_PARENT\_IDX" ON "NG\_REPORTS"."PORT" ("PARENT\_PORT\_REF\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INC\_MSPP\_SLOT\_TID\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."INC\_MSPP\_SLOT\_TID\_IDX" ON "NG\_REPORTS"."INC\_MSPP\_SLOT" ("TID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index MOTO\_CDMA\_T1\_INV\_OMCR\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."MOTO\_CDMA\_T1\_INV\_OMCR\_IDX" ON "NG\_REPORTS"."MOTO\_CDMA\_T1\_INV" ("VSM\_DEVICE\_NAME\_OMCR")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SAM\_CSR\_VLAN\_NXA\_NTLS\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SAM\_CSR\_VLAN\_NXA\_NTLS\_IDX" ON "NG\_REPORTS"."SAM\_CSR\_VLAN\_AUDIT" ("CSR\_VENDOR", "NG\_VLAN\_NUMBER", "VLAN\_STATUS", "MATCH\_CODE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INDX\_LAST\_TEST\_SUMMARY

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."INDX\_LAST\_TEST\_SUMMARY" ON "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY" ("RUN\_DATE" DESC, "DOMAIN\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index IDX\_REF\_ID\_GR\_ATTR

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."IDX\_REF\_ID\_GR\_ATTR" ON "NG\_REPORTS"."SITE\_ATTRIBUTES" ("SITE\_REFERENCE\_ID", "SITE\_GROUP\_NAME", "SITE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_XCONNECT\_INDEX1

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."NETSMART\_XCONNECT\_INDEX1" ON "NG\_REPORTS"."NETSMART\_XCONNECT" ("SNCNE\_ID", "XCONNECT\_ID", "XCONNECT\_ASIDE", "XCONNECT\_ZSIDE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SAM\_ENB\_WK\_IDX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."SAM\_ENB\_WK\_IDX1" ON "NG\_REPORTS"."SAM\_ENB\_WK" ("CHASSIS\_TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index SITE\_PORTAL\_DOMAIN\_MAPPING\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."SITE\_PORTAL\_DOMAIN\_MAPPING\_PK" ON "NG\_REPORTS"."SITE\_PORTAL\_DOMAIN\_MAPPING" ("SITE\_PORTAL\_DOMAIN\_MAPPING\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INC\_MSPP\_XCONN\_AUDIT\_SUMMA\_U01

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_SUMMA\_U01" ON "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_SUMMARY" ("TID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index INC\_MSPPXCONNAUDITDETAILS\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."INC\_MSPPXCONNAUDITDETAILS\_IDX" ON "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_DETAILS" ("DEVICE\_XCONN\_ELEMENTS")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NETSMART\_SNCNE\_INDEX2

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NETSMART\_SNCNE\_INDEX2" ON "NG\_REPORTS"."NETSMART\_SNCNE" ("TID\_TYPE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index LNVXA\_BTS\_URC\_DS1\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."LNVXA\_BTS\_URC\_DS1\_IDX" ON "NG\_REPORTS"."LUCENT\_NE\_VS\_NG\_AUDIT" ("SWITCH\_ID", "NG\_CELL\_NUM", "NG\_URC\_NUM", "NG\_DS1\_NUM")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index PORT\_RESERVATION\_INST\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."PORT\_RESERVATION\_INST\_IDX" ON "NG\_REPORTS"."PORT" ("PORT\_RESVATION\_REF\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NG\_VLAN\_NXA\_XNG\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NG\_VLAN\_NXA\_XNG\_IDX" ON "NG\_REPORTS"."NGMLS\_VLAN\_AUDIT" ("NGMLS\_VENDOR", "EQP\_REFERENCE\_ID", "NG\_VLAN\_NUMBER", "VLAN\_STATUS", "MATCH\_CODE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NORTEL\_NE\_VS\_NG\_AUDIT\_IDX1

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT\_IDX1" ON "NG\_REPORTS"."NORTEL\_NE\_VS\_NG\_AUDIT" ("VSM\_DEVICE\_NAME\_EMS", "VSM\_DEVICE\_NAME\_MTX", "SWITCH\_ID", "VSM\_DEVICE\_NAME\_BSM")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NIS\_EXTRACTOR\_GRAPH\_2IX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH\_2IX" ON "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" ("PARENT\_NODE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index NPI\_SEQNO\_PORT\_TYPE\_SPAN\_IDX

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."NPI\_SEQNO\_PORT\_TYPE\_SPAN\_IDX" ON "NG\_REPORTS"."NORTEL\_PORT\_INVENTORY" ("DOM\_SEQ\_NUM", "PORT\_TYPE", "PORT\_NUM")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index EQP\_REFERENCE\_ID

--------------------------------------------------------

CREATE UNIQUE INDEX "NG\_REPORTS"."EQP\_REFERENCE\_ID" ON "NG\_REPORTS"."EQUIPMENT" ("EQP\_REFERENCE\_ID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index MNVXA\_IDX\_2

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."MNVXA\_IDX\_2" ON "NG\_REPORTS"."MOTO\_CDMA\_NE\_VS\_NG\_AUDIT" ("MTX", "MATCH\_CODE", "MATCH\_STATUS", "TERMINATION\_TYPE", "NG\_PATH\_STATUS")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Index BTOD\_XNG\_PP\_T

--------------------------------------------------------

CREATE INDEX "NG\_REPORTS"."BTOD\_XNG\_PP\_T" ON "NG\_REPORTS"."BTOD\_NTLS\_PP\_T" ("STRIPPED\_EC\_CIRCUIT\_ID", "INVENTORY\_VENDOR\_CODE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ;

--------------------------------------------------------

-- DDL for Trigger JV\_CDO\_CLASS\_CDO\_CLASS\_PK\_TRIG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TRIGGER "NG\_REPORTS"."JV\_CDO\_CLASS\_CDO\_CLASS\_PK\_TRIG" BEFORE INSERT OR UPDATE ON jv\_cdo\_class

FOR EACH ROW

DECLARE

v\_newVal NUMBER(12) := 0;

v\_incval NUMBER(12) := 0;

BEGIN

IF INSERTING AND :new.cdo\_class\_pk IS NULL THEN

SELECT jv\_cdo\_class\_cdo\_class\_pk\_SEQ.NEXTVAL INTO v\_newVal FROM DUAL;

-- If this is the first time this table have been inserted into (sequence == 1)

IF v\_newVal = 1 THEN

--get the max indentity value from the table

SELECT NVL(max(cdo\_class\_pk),0) INTO v\_newVal FROM jv\_cdo\_class;

v\_newVal := v\_newVal + 1;

--set the sequence to that value

LOOP

EXIT WHEN v\_incval>=v\_newVal;

SELECT jv\_cdo\_class\_cdo\_class\_pk\_SEQ.nextval INTO v\_incval FROM dual;

END LOOP;

END IF;

-- assign the value from the sequence to emulate the identity column

:new.cdo\_class\_pk := v\_newVal;

END IF;

END;

/

ALTER TRIGGER "NG\_REPORTS"."JV\_CDO\_CLASS\_CDO\_CLASS\_PK\_TRIG" ENABLE;

--------------------------------------------------------

-- DDL for Trigger JV\_COMMIT\_COMMIT\_PK\_TRIG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TRIGGER "NG\_REPORTS"."JV\_COMMIT\_COMMIT\_PK\_TRIG" BEFORE INSERT OR UPDATE ON jv\_commit

FOR EACH ROW

DECLARE

v\_newVal NUMBER(12) := 0;

v\_incval NUMBER(12) := 0;

BEGIN

IF INSERTING AND :new.commit\_pk IS NULL THEN

SELECT jv\_commit\_commit\_pk\_SEQ.NEXTVAL INTO v\_newVal FROM DUAL;

-- If this is the first time this table have been inserted into (sequence == 1)

IF v\_newVal = 1 THEN

--get the max indentity value from the table

SELECT NVL(max(commit\_pk),0) INTO v\_newVal FROM jv\_commit;

v\_newVal := v\_newVal + 1;

--set the sequence to that value

LOOP

EXIT WHEN v\_incval>=v\_newVal;

SELECT jv\_commit\_commit\_pk\_SEQ.nextval INTO v\_incval FROM dual;

END LOOP;

END IF;

-- assign the value from the sequence to emulate the identity column

:new.commit\_pk := v\_newVal;

END IF;

END;

/

ALTER TRIGGER "NG\_REPORTS"."JV\_COMMIT\_COMMIT\_PK\_TRIG" ENABLE;

--------------------------------------------------------

-- DDL for Trigger JV\_GLOBAL\_ID\_GLOBAL\_ID\_PK\_TRIG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TRIGGER "NG\_REPORTS"."JV\_GLOBAL\_ID\_GLOBAL\_ID\_PK\_TRIG" BEFORE INSERT OR UPDATE ON jv\_global\_id

FOR EACH ROW

DECLARE

v\_newVal NUMBER(12) := 0;

v\_incval NUMBER(12) := 0;

BEGIN

IF INSERTING AND :new.global\_id\_pk IS NULL THEN

SELECT jv\_global\_id\_global\_id\_pk\_SEQ.NEXTVAL INTO v\_newVal FROM DUAL;

-- If this is the first time this table have been inserted into (sequence == 1)

IF v\_newVal = 1 THEN

--get the max indentity value from the table

SELECT NVL(max(global\_id\_pk),0) INTO v\_newVal FROM jv\_global\_id;

v\_newVal := v\_newVal + 1;

--set the sequence to that value

LOOP

EXIT WHEN v\_incval>=v\_newVal;

SELECT jv\_global\_id\_global\_id\_pk\_SEQ.nextval INTO v\_incval FROM dual;

END LOOP;

END IF;

-- assign the value from the sequence to emulate the identity column

:new.global\_id\_pk := v\_newVal;

END IF;

END;

/

ALTER TRIGGER "NG\_REPORTS"."JV\_GLOBAL\_ID\_GLOBAL\_ID\_PK\_TRIG" ENABLE;

--------------------------------------------------------

-- DDL for Trigger PORT\_PORT\_REFERENCE\_ID\_1TRIG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TRIGGER "NG\_REPORTS"."PORT\_PORT\_REFERENCE\_ID\_1TRIG" BEFORE INSERT OR UPDATE ON PORT

FOR EACH ROW

DECLARE

v\_newVal NUMBER(12) := 0;

v\_incval NUMBER(12) := 0;

BEGIN

IF INSERTING AND :new.PORT\_REFERENCE\_ID IS NULL THEN

SELECT PORT\_PORT\_REFERENCE\_ID\_2SEQ.NEXTVAL INTO v\_newVal FROM DUAL;

-- If this is the first time this table have been inserted into (sequence == 1)

IF v\_newVal = 1 THEN

--get the max indentity value from the table

SELECT NVL(max(PORT\_REFERENCE\_ID),0) INTO v\_newVal FROM PORT;

v\_newVal := v\_newVal + 1;

--set the sequence to that value

LOOP

EXIT WHEN v\_incval>=v\_newVal;

SELECT PORT\_PORT\_REFERENCE\_ID\_2SEQ.nextval INTO v\_incval FROM dual;

END LOOP;

END IF;

-- assign the value from the sequence to emulate the identity column

:new.PORT\_REFERENCE\_ID := v\_newVal;

END IF;

END;

/

ALTER TRIGGER "NG\_REPORTS"."PORT\_PORT\_REFERENCE\_ID\_1TRIG" ENABLE;

--------------------------------------------------------

-- DDL for Trigger PORT\_PORT\_REFERENCE\_ID\_TRIG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE TRIGGER "NG\_REPORTS"."PORT\_PORT\_REFERENCE\_ID\_TRIG" BEFORE INSERT OR UPDATE ON "PORT\_OLD"

FOR EACH ROW

DECLARE

v\_newVal NUMBER(12) := 0;

v\_incval NUMBER(12) := 0;

BEGIN

IF INSERTING AND :new.PORT\_REFERENCE\_ID IS NULL THEN

SELECT PORT\_PORT\_REFERENCE\_ID\_SEQ.NEXTVAL INTO v\_newVal FROM DUAL;

-- If this is the first time this table have been inserted into (sequence == 1)

IF v\_newVal = 1 THEN

--get the max indentity value from the table

SELECT NVL(max(PORT\_REFERENCE\_ID),0) INTO v\_newVal FROM PORT;

v\_newVal := v\_newVal + 1;

--set the sequence to that value

LOOP

EXIT WHEN v\_incval>=v\_newVal;

SELECT PORT\_PORT\_REFERENCE\_ID\_SEQ.nextval INTO v\_incval FROM dual;

END LOOP;

END IF;

-- assign the value from the sequence to emulate the identity column

:new.PORT\_REFERENCE\_ID := v\_newVal;

END IF;

END;

/

ALTER TRIGGER "NG\_REPORTS"."PORT\_PORT\_REFERENCE\_ID\_TRIG" ENABLE;

--------------------------------------------------------

-- DDL for Procedure ASSIGN\_AREAS\_TO\_DEVICES

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."ASSIGN\_AREAS\_TO\_DEVICES" AS

methodname VARCHAR2(30) := 'assign\_area\_region\_to\_devices';

message VARCHAR2(300);

updstmt VARCHAR2(32767);

CURSOR area\_region\_dev IS WITH domains AS (

SELECT DISTINCT

substr(clli,1,6) clli\_6,

territory,

market\_territory,

sub\_market,

sub\_market\_leaf,

leaf\_domain\_id

FROM

domains\_leaf\_reporting dlf

LEFT OUTER JOIN clli\_domain\_map clli ON clli.leaf\_domain\_id = dlf.domain\_id

WHERE

sub\_market <> 'NNO'

AND sub\_market <> 'OSS'

) SELECT

di.territory,

di.market\_territory,

di.sub\_market,

di.sub\_market\_leaf,

di.leaf\_domain\_id,

ngmls\_device\_name

FROM

ngmls\_equip\_audit aud

JOIN domains di ON di.clli\_6 = substr(aud.ngmls\_device\_name,1,6);

CURSOR eq\_area\_region\_dev IS WITH domains AS (

SELECT DISTINCT

substr(clli,1,6) clli\_6,

territory,

market\_territory,

sub\_market,

sub\_market\_leaf,

leaf\_domain\_id

FROM

domains\_leaf\_reporting dlf

LEFT OUTER JOIN clli\_domain\_map clli ON clli.leaf\_domain\_id = dlf.domain\_id

WHERE

sub\_market <> 'NNO'

AND sub\_market <> 'OSS'

) SELECT

aud.territory,

aud.market\_territory,

aud.sub\_market,

aud.market,

di.leaf\_domain\_id,

ngmls\_device\_name

FROM

ngmls\_equip\_audit aud

JOIN ng\_reports.equipment\_domain\_map edm ON edm.eqp\_reference\_id = aud.eqp\_reference\_id

JOIN domains di ON di.leaf\_domain\_id = edm.domain\_id -- di.leaf\_domain\_id = edm.DOMAIN\_ID

WHERE

aud.territory IS NULL -- why?

;

BEGIN

updstmt := 'update ngmls\_equip\_audit set territory = :territory

, market\_territory = :market\_territory, sub\_market = :sub\_market, market = :sub\_market\_leaf

, leaf\_domain\_id = :leaf\_domain\_id

where ngMLS\_device\_name = :ngMLS\_device\_name'

;

FOR rec IN area\_region\_dev LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,rec.market\_territory,rec.sub\_market,rec.sub\_market\_leaf,rec.leaf\_domain\_id,rec.ngmls\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market for '

|| rec.ngmls\_device\_name

|| ' in table: ngmls\_equip\_audit';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

--watchdog.logerror (methodName,

-- 8000,

-- SUBSTR (message

-- || TO\_CHAR (SQLCODE)

-- || ': '

-- || SQLERRM,

-- 1,

-- 255

-- ),

-- 'Y'

-- );

ROLLBACK;

RAISE;

END;

END LOOP;

-- area region for put xng\_only based on the area/region of EQP\_REFERENCE\_ID???????????

FOR rec IN eq\_area\_region\_dev LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,rec.market\_territory,rec.sub\_market,rec.market,rec.leaf\_domain\_id,rec.ngmls\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from ng Only for '

|| rec.ngmls\_device\_name

|| ' in table: ngmls\_equip\_audit';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

ROLLBACK;

RAISE;

END;

END LOOP;

-- if there still are some devices which cant be mapped to any domain then mark those as unknown

BEGIN

updstmt := ' update ngmls\_equip\_audit

set territory = ''unknown''

, market\_territory = ''unknown''

, sub\_market = ''unknown''

, market = ''unknown''

, leaf\_domain\_id = 0

where territory is null '

;

EXECUTE IMMEDIATE updstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from to unknown in table: ngmls\_equip\_audit';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

ROLLBACK;

RAISE;

END;

END ASSIGN\_AREAS\_TO\_DEVICES;

/

--------------------------------------------------------

-- DDL for Procedure ASSIGN\_AREA\_REGION\_TO\_CSR

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."ASSIGN\_AREA\_REGION\_TO\_CSR"

IS

methodname VARCHAR2 (30) := 'assign\_area\_region\_to\_CSR';

MESSAGE VARCHAR2 (300);

updstmt VARCHAR2 (32767);

--In Granite there are 48,947 distinct csr\_device\_name, nautilus has 62,319. Nautilus has alot of dups.

--Granite has no records in the sam\_gi\_al\_csr\_audit\_wk table only in the non-wk table

--sam\_gi\_al\_csr\_audit: 49,147 / sam\_NT\_al\_csr\_audit\_wk: 113,440

CURSOR area\_region\_dev

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS')--Granite 609 / Nautilus 608

SELECT di.territory, di.market\_territory, di.sub\_market, di.sub\_market\_leaf, di.leaf\_domain\_id,

csr\_device\_name, clli\_6

FROM sam\_NT\_al\_csr\_audit\_wk aud JOIN domains di

ON di.clli\_6 = SUBSTR (aud.csr\_device\_name, 1, 6)

;--Granite 49,068 non-wk / Nautilus 113,385

CURSOR eq\_area\_region\_dev

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS')--Granite 609 / Nautilus 608

SELECT di.territory, di.market\_territory, di.sub\_market, di.sub\_market\_leaf, di.leaf\_domain\_id,

csr\_device\_name, clli\_6

FROM sam\_NT\_al\_csr\_audit\_wk aud JOIN ng\_reports.EQUIPMENT\_DOMAIN\_MAP edm--- ref vzwnet.equip\_domain\_map

ON edm.EQP\_REFERENCE\_ID = aud.EQP\_REFERENCE\_ID---ref EQUIP\_INST\_ID

JOIN domains di ON di.leaf\_domain\_id = edm.domain\_id

WHERE aud.territory IS NULL

;--Granite 0 non-wk / Nautilus 2,031,134

CURSOR unknown\_domain

IS

SELECT dlr.territory AS territory,dlr.market\_territory AS market\_territory, dlr.sub\_market AS sub\_market, dlr.sub\_market\_leaf AS sub\_market\_leaf,

dlr.domain\_id AS domain\_id,

sgalwk.csr\_device\_name AS csr\_device\_name

FROM ng\_reports.EQUIPMENT equip,

sam\_NT\_al\_csr\_audit\_wk sgalwk,

ng\_reports.EQUIPMENT\_DOMAIN\_MAP equipmap,---ref vzwnet.equip\_domain\_map

ng\_reports.domains\_leaf\_reporting dlr

WHERE sgalwk.leaf\_domain\_id = 0

AND equip.eqp\_name = sgalwk.sam\_csr\_device\_name

AND equipmap.EQP\_REFERENCE\_ID = equip.EQP\_REFERENCE\_ID---EQP\_inst\_ID

AND dlr.domain\_id = equipmap.domain\_id

;--Granite 0 / Nautilus 0

BEGIN

updstmt :=

'update SAM\_NT\_AL\_CSR\_AUDIT\_WK set territory = :territory, market\_territory = :market\_territory

, sub\_market = :sub\_market, market = :sub\_market\_leaf

, leaf\_domain\_id = :leaf\_domain\_id---ref LEAF\_DOMAIN\_INST\_ID

, clli = :clli\_6

where csr\_device\_name = :csr\_device\_name';

FOR rec IN area\_region\_dev

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.leaf\_domain\_id,---ref LEAF\_DOMAIN\_INST\_ID

rec.clli\_6,

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market for '

|| rec.csr\_device\_name

|| ' in table: SAM\_NT\_AL\_CSR\_AUDIT\_WK';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

-- area region for put xng\_only based on the area/region of ne\_inst\_id???????????

FOR rec IN eq\_area\_region\_dev

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.leaf\_domain\_id,

rec.clli\_6,

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from Ng Only for '

|| rec.csr\_device\_name

|| ' in table: SAM\_NT\_AL\_CSR\_AUDIT\_WK';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

BEGIN

-- if there still are some devices which cant be mapped to any domain then mark those as unknown

updstmt :=

' update SAM\_NT\_AL\_CSR\_AUDIT\_WK

set territory = ''unknown''

, market\_territory = ''unknown''

, sub\_market = ''unknown''

, market = ''unknown''

, leaf\_domain\_id = 0---LEAF\_DOMAIN\_ID

, clli = ''unknown''

where territory is null ';

EXECUTE IMMEDIATE updstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from to unknown in table: SAM\_NT\_AL\_CSR\_AUDIT\_WK';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

updstmt :=

' update SAM\_NT\_AL\_CSR\_AUDIT\_WK

set territory = :territory, market\_territory = :market\_territory

, sub\_market = :sub\_market, market = :market

, leaf\_domain\_id = :DOMAIN\_ID

where csr\_device\_name = :csr\_device\_name ';

FOR rec IN unknown\_domain

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.domain\_id,---LEAF\_DOMAIN\_ID

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from Ng Only for '

|| rec.csr\_device\_name

|| ' in table: SAM\_NT\_AL\_CSR\_AUDIT\_WK';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

BEGIN

-- if some ne\_hostname matched > 1 granite ne\_inst\_id then mark it dup

-- does stmt query work even when there are "Dup in NE"???

UPDATE sam\_NT\_al\_csr\_audit\_wk aud

SET clli = SUBSTR (aud.csr\_device\_name, 1, 6)

WHERE clli = 'unknown';

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname

|| '(): Can''t update clli for unknown';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

END assign\_area\_region\_to\_csr;

/

--------------------------------------------------------

-- DDL for Procedure ASSIGN\_AREA\_REGION\_TO\_CSR\_VLAN

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."ASSIGN\_AREA\_REGION\_TO\_CSR\_VLAN" as

methodname VARCHAR2 (30) := 'assign\_area\_region\_to\_csr\_vlan';

MESSAGE VARCHAR2 (300);

updstmt VARCHAR2 (32767);

processname VARCHAR2 (30) := 'SAM\_GI\_CSR\_AUDIT';

CURSOR area\_region\_dev IS WITH domains AS

(SELECT DISTINCT SUBSTR (CLLI, 1, 6) clli\_6, TERRITORY, SUB\_MARKET, SUB\_MARKET\_LEAF, LEAF\_DOMAIN\_ID

FROM NG\_Reports.clli\_domain\_map\_v

WHERE SUB\_MARKET <> 'NNO' AND SUB\_MARKET <> 'OSS')

SELECT di.TERRITORY, di.SUB\_MARKET, di.SUB\_MARKET\_LEAF, di.LEAF\_DOMAIN\_ID, aud.csr\_device\_name, clli\_6

FROM SAM\_CSR\_VLAN\_AUDIT\_WK aud

JOIN domains di ON di.clli\_6 = SUBSTR (aud.csr\_device\_name, 1, 6);

CURSOR area\_region\_SAM IS

SELECT di.TERRITORY, di.SUB\_MARKET, di.market, di.LEAF\_DOMAIN\_ID, di.csr\_device\_name, di.CLLI

FROM SAM\_CSR\_VLAN\_AUDIT\_WK aud

JOIN SAM\_NT\_AL\_CSR\_AUDIT di ON di.csr\_device\_name = aud.csr\_device\_name -- Has to be Created

where aud.TERRITORY is null;

BEGIN

updstmt :=

'update SAM\_CSR\_VLAN\_AUDIT\_WK set TERRITORY = :TERRITORY, SUB\_MARKET = :SUB\_MARKET, market = :market, LEAF\_DOMAIN\_ID = :LEAF\_DOMAIN\_ID, CLLI = :clli\_6

where csr\_device\_name = :csr\_device\_name';

FOR rec IN area\_region\_dev

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.TERRITORY,rec.SUB\_MARKET,rec.SUB\_MARKET\_LEAF,rec.LEAF\_DOMAIN\_ID,rec.clli\_6,rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE := 'Error: ' || methodname || '(): Cant update clli for ' || rec.csr\_device\_name || ' in table: SAM\_CSR\_VLAN\_AUDIT\_WK';

DBMS\_OUTPUT.put\_line (SUBSTR (MESSAGE || TO\_CHAR (SQLCODE) || ': ' || SQLERRM, 1, 255));

ROLLBACK;

RAISE;

END;

END LOOP;

END ASSIGN\_AREA\_REGION\_TO\_CSR\_VLAN;

/

--------------------------------------------------------

-- DDL for Procedure AUDIT\_NGMLS\_EQUIP

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."AUDIT\_NGMLS\_EQUIP" IS

methodname VARCHAR2(30) := 'audit\_ngMLS\_equip';

message VARCHAR2(500);

sqlstmt VARCHAR2(32767);

BEGIN

Begin

dbms\_output.put\_line('deleting');

DELETE FROM ngmls\_equip\_audit\_wk nxa

WHERE

ngmls\_vendor IN (

'CISCO',

'ALCATEL-LUCENT'

);

--delete from ngMLS\_device\_AUDIT\_issues\_wk iss

-- where ngMLS\_vendor = 'CI';

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t delete records from table: NGMLS\_EQUIP\_AUDIT\_WK';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

end;

Begin

insert into ngmls\_equip\_audit\_wk

columns (ngMLS\_vendor, ngMLS\_device\_name

, ne\_device\_name

, gi\_device\_name, eqp\_reference\_id

, Live\_in\_ng, inv\_status

, match\_code

)

WITH NCM\_ngMLSs as

(

select distinct case when ncm.hostname is not null then ncm.hostname

when nvm.hostname is not null then nvm.hostname end host\_name, case when ncm.hostname is not null and nvm.hostname is not null then 'BOTH'

else 'NE ONLY' end match\_code

from ncm\_ngMLS\_wk ncm full outer join ncm\_ngMLS\_vlan\_wk nvm on

ncm.hostname = nvm.hostname where vlan\_number is not null

)

,

SAM\_ngMLSs as

(

select distinct SITE\_NAME --, act\_mgmt\_ip sam\_ngMLS\_device\_ip

, reachability sam\_reachability

, resync\_status sam\_resync\_status, resync\_state sam\_resync\_state

--, server\_name sam\_server\_name, last\_resyn\_start sam\_last\_resync\_start, last\_resyn\_end sam\_last\_resync\_end

from sam\_ebh\_health where site\_name LIKE '%7750-0\_'

)

,

Xng as

(

--ngmls.pattern1=^([A-Z0-9]{1,4} |)[A-Z0-9]{8}[8-9]\\dA-P-(AL|CI)-(9010|7750)-\\d{2}(\\s\*\\[[^\\]\\[]+\\])?

--ngmls.pattern2=.\*([A-Z0-9]{8}[8-9]\\dA-P-(AL|CI)-(9010|7750)-\\d{2}).\*

select ngMLS.\*, case when REGEXP\_SUBSTR(eqp\_name, '([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}') is not null

then

REGEXP\_SUBSTR(eqp\_name, '([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}')

else eqp\_name end

ngMLS\_device\_name

from equipment ngMLS

where eqp\_vendor IN ('CISCO','ALCATEL-LUCENT') and

ngMLS.eqp\_type = 'NGMLS'

--and regexp\_like(eqp\_name, '([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}')

)

select

case when xng.ngMLS\_device\_name is not null then eqp\_vendor

when nfj.host\_name is not null then 'CISCO'

when sam.SITE\_NAME is not null then 'ALCATEL-LUCENT'

end vendor

--Need to update

,case when nfj.host\_name is not null then nfj.host\_name

when sam.SITE\_NAME is not null then sam.SITE\_NAME

when xng.ngMLS\_device\_name is not null then xng.ngMLS\_device\_name --Need to update

else 'zzzz'||xng.eqp\_name

end ngMLS\_device\_name

,

case when nfj.host\_name is not null then nfj.host\_name

when sam.SITE\_NAME is not null then sam.SITE\_NAME

-- else ne\_device\_name

end ne\_device\_name

, xng.eqp\_name gi\_device\_name, eqp\_reference\_id

, case when inv\_status = 'Live' then 'Y'

when inv\_status is not null then 'N'

end Live\_in\_ng

, inv\_status

, case when nfj.host\_name is not null

and xng.eqp\_reference\_id is not null

and match\_code = 'BOTH'

then 'BOTH'

-- it is in both NCM extracts but not in Xng then NE Only

when nfj.host\_name is not null or sam.SITE\_NAME is not null

and xng.eqp\_reference\_id is null

--and match\_code = 'BOTH'

then 'NE Only'

when sam.SITE\_NAME is not null

and xng.eqp\_reference\_id is not null

then 'BOTH'

when nfj.host\_name is null and sam.SITE\_NAME is null and xng.eqp\_reference\_id is not null then 'Nautilus Only'

else match\_code -- when the device is not found in both NCM extracts

end match\_code

-- , case when xng.ne\_inst\_id is null then nvl2(match\_status, '15, '||match\_status, '15')

-- else nvl2(parse\_status, parse\_status || ', ' ||match\_status, match\_status) --null -- when does this condition appear

-- end match\_status

-- , parse\_status match\_status

from NCM\_ngMLSs nfj

full outer join Xng

on xng.eqp\_name like '%'||nfj.host\_name ||'%'

full outer join SAM\_ngMLSs sam

on xng.eqp\_name like '%'||sam.SITE\_NAME ||'%';

commit;

END;

END;

/

--------------------------------------------------------

-- DDL for Procedure COPY\_MOTO\_CDMA\_WRK

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."COPY\_MOTO\_CDMA\_WRK" IS

sqlStmt varchar2(32000);

v\_ready NG\_REPORTS.all\_processes.is\_ready%type;

BEGIN

select is\_ready into v\_ready from NG\_REPORTS.all\_processes where process\_name = 'MOTO\_CDMA\_AUDIT';

IF v\_ready = 'Y' then

sqlStmt := 'delete from moto\_cdma\_ne\_vs\_ng\_audit';

execute immediate sqlStmt;

sqlStmt := 'insert into moto\_cdma\_ne\_vs\_ng\_audit select \* from MOTO\_CDMA\_NE\_VS\_NG\_AUDIT';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done populating moto\_cdma\_ne\_vs\_ng\_audit');

sqlStmt := 'delete from moto\_cdma\_audit\_reg\_summ';

execute immediate sqlStmt;

sqlStmt := 'insert into moto\_cdma\_audit\_reg\_summ (T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,MARKET\_TERRITORY,SUB\_MARKET) select T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,MARKET\_TERRITORY,SUB\_MARKET from MOTO\_CDMA\_AUDIT\_REG\_SUMM';

execute immediate sqlStmt;

commit;

sqlStmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = MARKET\_TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is null' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.MARKET\_TERRITORY = ''Unknown'',wk.SUB\_MARKET=''Unknown'' where territory=''Unknown''' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

dbms\_output.put\_line('Done populating moto\_cdma\_audit\_reg\_summ');

sqlStmt := 'delete from moto\_cdma\_audit\_dev\_summ';

execute immediate sqlStmt;

sqlStmt := 'insert into moto\_cdma\_audit\_dev\_summ select \* from MOTO\_CDMA\_AUDIT\_DEV\_SUMM';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done populating moto\_cdma\_audit\_dev\_summ');

END IF;

EXCEPTION

WHEN OTHERS THEN

WATCHDOG.logerror('MOTO\_CDMA\_AUDIT',4000,SubStr('Error in copy\_moto\_cdma\_wrk; ' ||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255),'Y');

WATCHDOG.updateprocessend('MOTO\_CDMA\_AUDIT','STATUS\_FAILURE','N');

END;

/

--------------------------------------------------------

-- DDL for Procedure INSERTIONS

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."INSERTIONS"

AS

SQLINSERT VARCHAR2(30000);

BEGIN

SQLINSERT := '<?xml version="1.0" encoding="UTF-8"?>

<export>

<properties>

<property name="useAbsoluteFilename" value="false"/>

<property name="timestampFormat" value="yyyyMMdd"/>

<property name="createSubDirWithRunDate" value="true"/>

<property name="verifyRemoteHost" value="false"/>

<!--Optional) For CSV export only !! Specify the custom delimter of the fields. Default: ,-->

<property name="delimiter" value="|"/>

</properties>

<sql id="1" name="TESS" >

<statement>

WITH sites\_with\_domains AS (

SELECT

-- old\_territory territory,

dlr.territory,

dlr.market\_territory,

dlr.sub\_market,

dlr.domain\_name AS market,

si.site\_reference\_id,

si.site\_name,

replace(replace(si.address,CHR(10),'' ''),CHR(14),'''') address,

si.city,

si.state,

si.latitude,

si.longitude,

si.status,

si.zip\_code,

si.county

FROM

ng\_reports.site si

JOIN ng\_reports.site\_domain\_map sdm ON si.site\_reference\_id = sdm.site\_reference\_id

JOIN ng\_reports.domains\_leaf\_reporting dlr ON dlr.domain\_id = sdm.domain\_id

MINUS

SELECT

''NNO'',

''NNO'',

''NNO\_1'',

''NNO'',

si.site\_reference\_id,

si.site\_name,

replace(replace(replace(replace(si.address,CHR(10),CHR(32) ),CHR(13),CHR(32) ),CHR(9),CHR(32) ),CHR(34),CHR(44) ) address,

si.city,

si.state,

si.latitude,

si.longitude,

si.status,

zip\_code,

si.county

FROM

ng\_reports.site si

JOIN ng\_reports.site\_domain\_map sdm ON si.site\_reference\_id = sdm.site\_reference\_id

JOIN ng\_reports.domains\_leaf\_reporting dlr ON dlr.domain\_id = sdm.domain\_id

WHERE

dlr.sub\_market <> ''NNO''

),psloc AS (

SELECT

sds.site\_reference\_id,

substr( (to\_number(ps\_loc\_code) + 10000000000),2,10) AS ps\_loc\_code

FROM

(

SELECT

sas.site\_reference\_id,

nvl(length(TRIM(translate(sas.site\_value,'' +-.0123456789'','' '') ) ),0) AS numck,

sas.site\_value ps\_loc\_code

FROM

ng\_reports.site\_attributes sas

WHERE

sas.site\_attributes\_id = 6963 -- PeopleSoft Location Code #

) sds

WHERE

numck = 0

) SELECT

''"''

|| replace(replace(replace(gig.site\_reference\_id,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.site\_name,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.territory,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.market\_territory,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.sub\_market,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.market,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.nwf\_site\_id,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.ps\_loc\_code,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.address,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.city,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.state,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.post\_code\_1,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.county,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.latitude,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.longitude,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"|"''

|| replace(replace(replace(gig.status,CHR(124),''''),CHR(10),''''),CHR(34),'''')

|| ''"''

FROM

(

SELECT

si.site\_reference\_id,

si.site\_name,

si.territory,

si.market\_territory,

si.sub\_market,

si.market,

--nsm.nauru\_site\_id nwf\_site\_id,

nsm.site\_reference\_id nwf\_site\_id,

ps.ps\_loc\_code,

si.address,

si.city,

si.state,

zip\_code post\_code\_1,

si.county,

si.latitude,

si.longitude,

si.status

FROM

sites\_with\_domains si

LEFT OUTER JOIN psloc ps ON ps.site\_reference\_id = si.site\_reference\_id

LEFT OUTER JOIN (

SELECT

\*

FROM

site\_attributes sa

WHERE

sa.site\_name LIKE ''NWF Site ID (NAURU)''

)--xng\_apps.nauru\_xng\_site\_map

nsm ON nsm.site\_reference\_id = si.site\_reference\_id -- NWF Site ID (NAURU)

WHERE

si.status NOT IN (

''Decommissioned'',

''Cancelled''

)

) gig;

</statement>

</sql>

</export>';

update nis\_extractor set rule\_definition = SQLINSERT

where rule\_id = 604;

END;

/

--------------------------------------------------------

-- DDL for Procedure LOAD\_HIGH\_CAP\_PATH\_CHAN

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."LOAD\_HIGH\_CAP\_PATH\_CHAN"

IS

v\_dstart date;

v\_cnt number;

sql\_stmt varchar2(100);

BEGIN

v\_cnt := 0;

v\_dstart := sysdate;

sql\_stmt:= 'truncate table high\_capacity\_path\_chan\_act';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in LOAD\_HIGH\_CAP\_PATH\_CHAN: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

--get transport action data - done separately for performance

insert into ng\_reports.high\_capacity\_path\_chan\_act

with cp as (select cpi.trail\_id, sum (ci.recur\_costs) act\_tot\_mrc

from (select trail\_id

from ng\_topology.trail

where status not in ('DECOMMISSIONED','CANCELLED')) cpi,

(select current\_trail\_id, recur\_costs

from ng\_topology.trail\_segment ci

where status not in ('DECOMMISSIONED','CANCELLED')) ci

where cpi.trail\_id = ci.current\_trail\_id (+)

group by cpi.trail\_id),

--REMOVED inst\_audit and inst\_update\_log will not be moved to nautilus

-- actdt as

-- (select cp.trail\_id, trunc(max(a.chg\_ts)) as act\_date

-- from cp,

-- (select ia.inst\_id, ia.chg\_ts, ia.version\_chgd

-- from vzwnet.inst\_audit ia, cp

-- where ia.inst\_id = cp.trail\_id

-- and ia.element\_type = 'P'

-- and ia.version\_chgd is not null) a,

-- (select iul.inst\_id, max(iul.version\_chgd) as version\_chgd

-- from vzwnet.inst\_update\_log iul, cp

-- where iul.inst\_id = cp.trail\_id

-- and iul.element\_type = 'P'

-- and iul.version\_chgd is not null

-- and iul.field\_chgd in ('Capacity Reporting / Transport Action',

-- 'Capacity Reporting / Estimated Completion Date',

-- 'Capacity Reporting / Action Justification')

-- group by iul.inst\_id) b

-- where cp.trail\_id = a.inst\_id

-- and a.inst\_id = b.inst\_id

-- and a.version\_chgd = b.version\_chgd

-- group by cp.trail\_id),

ta as (select cp.trail\_id, cpas.attribute\_value trans\_act

from cp,

NG\_TOPOLOGY.TRAIL\_ATTRIBUTES cpas

where cp.trail\_id = cpas.trail\_id

and cpas.group\_name = 'Capacity Reporting'

and cpas.attribute\_name = 'Transport Action'

and cpas.attribute\_value is not null),

ec as (select trail\_id, est\_comp, est\_comp\_days,

case

when est\_comp\_days > 15 and est\_comp\_days <= 30 then 1

when est\_comp\_days > 30 and est\_comp\_days <= 45 then 2

when est\_comp\_days > 45 and est\_comp\_days <= 60 then 3

when est\_comp\_days > 60 and est\_comp\_days <= 90 then 4

when est\_comp\_days > 90 then 5

else 0

end est\_comp\_grp

from (select trail\_id, est\_comp,

nvl(trunc(sysdate - nvl(est\_comp,sysdate)),0) est\_comp\_days

from (select cp.trail\_id,

case

when regexp\_like(cpas.attribute\_value,'[[:digit:]]{2}-[[:alpha:]]{3}-[[:digit:]]{4}','i')

then to\_date(cpas.attribute\_value, 'dd-mon-yyyy') else null

end est\_comp

from cp,

NG\_TOPOLOGY.TRAIL\_ATTRIBUTES cpas

where cp.trail\_id = cpas.trail\_id

and cpas.group\_name = 'Capacity Reporting'

and cpas.attribute\_name = 'Estimated Completion Date'

and cpas.attribute\_value is not null))),

aj as (select cp.trail\_id, cpas.attribute\_value act\_just

from cp,

NG\_TOPOLOGY.TRAIL\_ATTRIBUTES cpas

where cp.trail\_id = cpas.trail\_id

and cpas.group\_name = 'Capacity Reporting'

and cpas.attribute\_name = 'Action Justification'

and cpas.attribute\_value is not null),

ajc as (select cp.trail\_id, cpas.attribute\_value act\_just\_comt

from cp,

NG\_TOPOLOGY.TRAIL\_ATTRIBUTES cpas

where cp.trail\_id = cpas.trail\_id

and cpas.group\_name = 'Capacity Reporting'

and cpas.attribute\_name = 'Action Justification Comments'

and cpas.attribute\_value is not null)

select cp.trail\_id, ta.trans\_act, ec.est\_comp, aj.act\_just, null act\_date, /\*ad.act\_date,\*/ ec.est\_comp\_grp,

cp.act\_tot\_mrc, ec.est\_comp\_days, ajc.act\_just\_comt

from cp, /\*actdt ad,\*/ ta, ec, aj, ajc

where /\*cp.trail\_id = ad.trail\_id

and \*/cp.trail\_id = ta.trail\_id (+)

and cp.trail\_id = ec.trail\_id (+)

and cp.trail\_id = aj.trail\_id (+)

and cp.trail\_id = ajc.trail\_id (+);

commit;

sql\_stmt:= 'truncate table high\_capacity\_path\_chan\_parent';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in LOAD\_HIGH\_CAP\_PATH\_CHAN: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

insert into NG\_REPORTS.high\_capacity\_path\_chan\_parent

with

--get all distinct path\_chan parents with current children and pending if there is no current

pathchan as (select distinct PARENT\_TRAIL\_ID

from ng\_topology.TRAIL\_CHANNEL),

pparent as (select distinct p.a\_site\_name, p.z\_site\_name, p.trail\_name, pc.PARENT\_TRAIL\_ID,

p.path\_type, p.status, p.bandwidth, p.TOTAL\_CHANNELS,

p.TOTAL\_CHANNELS\_USED, p.TOTAL\_CAPACITY\_AVL, p.TOTAL\_CAPACITY\_USED, p.bps\_util

from pathchan pc,

(select distinct TRAIL.a\_site\_name, TRAIL.z\_site\_name,

TRAIL.trail\_name, TRAIL.TRAIL\_ID,

TRAIL.Type as path\_type, TRAIL.STATUS,

TRAIL.BANDWIDTH, TRAIL.TOTAL\_CHANNELS,

nvl(TRAIL.TOTAL\_CHANNELS\_USED, 0) as TOTAL\_CHANNELS\_USED,

nvl(TOTAL\_CAPACITY\_AVL,0) TOTAL\_CAPACITY\_AVL,

nvl(TOTAL\_CAPACITY\_USED, 0) TOTAL\_CAPACITY\_USED, trunc(round((nvl(TOTAL\_CAPACITY\_USED,0) \* 100)/nvl(TOTAL\_CAPACITY\_AVL,0), 2)) bps\_util

from ng\_topology.TRAIL

where status not in ('DECOMMISSIONED','CANCELLED')

and (TRAIL.BANDWIDTH = 'DS3' or TRAIL.BANDWIDTH like 'OC%')) p

where pc.PARENT\_TRAIL\_ID = p.TRAIL\_ID),

--get segment information where that have a path

seginfo as (select nvl(CURRENT\_TRAIL\_ID, NEXT\_TRAIL\_ID) circ\_path\_inst\_id,

segment\_id as segment\_id, nvl(recur\_costs,0) recur\_costs

from ng\_topology.TRAIL\_SEGMENT

where status not in ('DECOMMISSIONED','CANCELLED')

and (CURRENT\_TRAIL\_ID is not null or NEXT\_TRAIL\_ID is not null)),

pparent\_seg as (select pp.a\_site\_name, pp.z\_site\_name, pp.trail\_name, pp.PARENT\_TRAIL\_ID,

pp.path\_type, pp.status, pp.bandwidth, pp.TOTAL\_CHANNELS,

pp.TOTAL\_CHANNELS\_USED, sum(s.recur\_costs) p\_tot\_mrc, pp.TOTAL\_CAPACITY\_AVL,

pp.TOTAL\_CAPACITY\_USED, pp.bps\_util, ta.trans\_act, ta.est\_comp, ta.act\_just, ta.act\_just\_comt,

ta.act\_date, ta.est\_comp\_grp, ta.act\_tot\_mrc,

case

when ta.trans\_act is null then 'Y'

when ta.trans\_act is not null and ta.est\_comp is not null and ta.est\_comp <= sysdate then 'Y'

when ta.trans\_act is not null and ta.est\_comp is null and ta.act\_date <= sysdate - 90 then 'Y'

else 'N'

end top\_disp,

case

when ta.est\_comp is not null and ta.est\_comp\_days > 0 then

ta.est\_comp\_days

when ta.est\_comp is not null and ta.est\_comp\_days <= 0 then

0

else

nvl(trunc(sysdate - nvl(ta.act\_date,sysdate)),0)

end est\_comp\_days

from pparent pp, seginfo s, (ng\_reports.high\_capacity\_path\_chan\_act) ta -- Required a table else need to add high\_capacity\_path\_chan\_act query

where pp.PARENT\_TRAIL\_ID = s.circ\_path\_inst\_id (+)

and pp.PARENT\_TRAIL\_ID = ta.TRAIL\_ID (+)

group by pp.a\_site\_name, pp.z\_site\_name, pp.trail\_name, pp.PARENT\_TRAIL\_ID,

pp.path\_type, pp.status, pp.bandwidth, pp.TOTAL\_CHANNELS, pp.TOTAL\_CHANNELS\_USED,

pp.TOTAL\_CAPACITY\_AVL, pp.TOTAL\_CAPACITY\_USED, pp.bps\_util, ta.trans\_act, ta.est\_comp,

ta.act\_just, ta.act\_just\_comt, ta.act\_date, ta.est\_comp\_grp, ta.act\_tot\_mrc, ta.est\_comp\_days),

la as (select domain\_id, domain\_name, territory, market\_territory, sub\_market, nno\_no || substr((alphaseq + 100),2,2) as leaf\_ord

from (select domain\_id, domain\_name, territory, market\_territory, sub\_market, rownum as alphaseq,

case when sub\_market = 'NNO' then 1 else 2 end nno\_no

from (select domain\_id, domain\_name, territory, market\_territory, sub\_market

from ng\_reports.domains\_regional\_reporting

where territory <> 'OSS'

order by territory, market\_territory, sub\_market, domain\_name))),

--get first the domain order seq for each parent path

mla as (select pp.PARENT\_TRAIL\_ID, min(leaf\_ord) as minl

from ng\_topology.TRAIL\_DOMAIN\_MAP pdm, pparent pp, la

where pp.PARENT\_TRAIL\_ID = pdm.TRAIL\_ID

and pdm.DOMAIN\_ID = la.domain\_id

group by pp.PARENT\_TRAIL\_ID),

--get domain information for the first domain for each parent path

cl as (select la.domain\_id, la.domain\_name, la.territory,

la.market\_territory, la.sub\_market,mla.PARENT\_TRAIL\_ID

from la, mla

where la.leaf\_ord = mla.minl),

--get parent's first domain and the first domain of the user who added the parent path

hcpc as (select cl.territory, cl.market\_territory, cl.sub\_market, cl.domain\_name, cl.domain\_id, pcci.a\_site\_name,

pcci.z\_site\_name, pcci.trail\_name pathparent\_name, pcci.PARENT\_TRAIL\_ID pathparent\_id,

pcci.path\_type pathparent\_type, pcci.status pathparent\_status,

pcci.bandwidth pathparent\_bandwidth, pcci.TOTAL\_CHANNELS pathparent\_nchl,

pcci.TOTAL\_CHANNELS\_USED pathparent\_nchla, pcci.p\_tot\_mrc, pcci.TOTAL\_CAPACITY\_AVL,

pcci.TOTAL\_CAPACITY\_USED, pcci.bps\_util, pcci.trans\_act, pcci.est\_comp, pcci.act\_just, pcci.act\_just\_comt,

pcci.act\_date, pcci.est\_comp\_grp, pcci.act\_tot\_mrc, pcci.top\_disp, pcci.est\_comp\_days

from pparent\_seg pcci, cl

where pcci.PARENT\_TRAIL\_ID = cl.parent\_trail\_id),

--get the percent of parent paths assigned and their grouping for use in the report summary

ppct as (select territory, market\_territory, sub\_market, pathparent\_id, pct\_asgn,

case

when pct\_asgn <= 20 then 1

when pct\_asgn >= 21 and pct\_asgn <= 40 then 2

when pct\_asgn >= 41 and pct\_asgn <= 60 then 3

when pct\_asgn >= 61 and pct\_asgn <= 80 then 4

when pct\_asgn >= 81 and pct\_asgn <= 100 then 5

else 6

end pct\_grp

from (select territory, market\_territory, sub\_market, pathparent\_id,

case when tot\_chan = 0 then 0

else bps\_util

end pct\_asgn

from (select distinct territory, market\_territory, sub\_market, pathparent\_id, pathparent\_nchl tot\_chan,

pathparent\_nchla tot\_asgn, total\_capacity\_avl, total\_capacity\_used, bps\_util

from hcpc)))

select domain\_id, domain\_name, territory, market\_territory as market, sub\_market, a\_site\_name, z\_site\_name, pathparent\_name,

pathparent\_id, pathparent\_type, pathparent\_status, pathparent\_bandwidth, pathparent\_nchl,

pathparent\_nchla, p\_tot\_mrc, p\_pct\_asgn, p\_pct\_grp, total\_capacity\_avl, total\_capacity\_used,

bps\_util, trans\_act, est\_comp, act\_just, act\_date, est\_comp\_grp, act\_tot\_mrc,

p\_top\_ar, p\_top\_arbw, est\_comp\_days, act\_just\_comt

from ((select domain\_id, domain\_name, territory, market\_territory, sub\_market, a\_site\_name, z\_site\_name, pathparent\_name,

pathparent\_id, pathparent\_type, pathparent\_status, pathparent\_bandwidth,

pathparent\_nchl, pathparent\_nchla, p\_tot\_mrc, p\_pct\_asgn, p\_pct\_grp, total\_capacity\_avl,

total\_capacity\_used, bps\_util, trans\_act, est\_comp, act\_just, act\_date, est\_comp\_grp, act\_tot\_mrc,

row\_number() over (partition by territory, market\_territory, sub\_market, p\_pct\_grp order by p\_tot\_mrc desc) as p\_top\_ar,

row\_number() over (partition by territory, market\_territory, sub\_market, p\_pct\_grp, pathparent\_bandwidth order by p\_tot\_mrc desc) as p\_top\_arbw,

est\_comp\_days, act\_just\_comt

from (select distinct hcpc.domain\_id, hcpc.domain\_name, hcpc.territory, hcpc.market\_territory, hcpc.sub\_market,

hcpc.a\_site\_name, hcpc.z\_site\_name, hcpc.pathparent\_name,

hcpc.pathparent\_id, hcpc.pathparent\_type, hcpc.pathparent\_status,

hcpc.pathparent\_bandwidth, hcpc.pathparent\_nchl, hcpc.pathparent\_nchla,

nvl(hcpc.p\_tot\_mrc,0) p\_tot\_mrc, hcpc.total\_capacity\_avl, hcpc.total\_capacity\_used,

hcpc.bps\_util, hcpc.trans\_act, hcpc.est\_comp, hcpc.act\_just, hcpc.act\_date,

hcpc.est\_comp\_grp, hcpc.act\_tot\_mrc, ppct.pct\_asgn p\_pct\_asgn,

ppct.pct\_grp p\_pct\_grp, hcpc.top\_disp, hcpc.est\_comp\_days, hcpc.act\_just\_comt

from hcpc, ppct

where hcpc.pathparent\_id = ppct.pathparent\_id

and hcpc.top\_disp = 'Y'))

union

(select domain\_id, domain\_name, territory, market\_territory, sub\_market, a\_site\_name, z\_site\_name, pathparent\_name,

pathparent\_id, pathparent\_type, pathparent\_status, pathparent\_bandwidth,

pathparent\_nchl, pathparent\_nchla, p\_tot\_mrc, p\_pct\_asgn, p\_pct\_grp, total\_capacity\_avl,

total\_capacity\_used, bps\_util, trans\_act, est\_comp, act\_just, act\_date, est\_comp\_grp,

act\_tot\_mrc, 99999 as p\_top\_ar, 99999 as p\_top\_arbw, est\_comp\_days, act\_just\_comt

from (select distinct hcpc.domain\_id, hcpc.domain\_name, hcpc.territory, hcpc.market\_territory, hcpc.sub\_market,

hcpc.a\_site\_name, hcpc.z\_site\_name, hcpc.pathparent\_name, hcpc.pathparent\_id,

hcpc.pathparent\_type, hcpc.pathparent\_status, hcpc.pathparent\_bandwidth,

hcpc.pathparent\_nchl, hcpc.pathparent\_nchla, nvl(hcpc.p\_tot\_mrc,0) p\_tot\_mrc,

hcpc.total\_capacity\_avl, hcpc.total\_capacity\_used, hcpc.bps\_util, hcpc.trans\_act,

hcpc.est\_comp, hcpc.act\_just, hcpc.act\_date, hcpc.est\_comp\_grp,

hcpc.act\_tot\_mrc, ppct.pct\_asgn p\_pct\_asgn, ppct.pct\_grp p\_pct\_grp,

hcpc.top\_disp, hcpc.est\_comp\_days, hcpc.act\_just\_comt

from hcpc, ppct

where hcpc.pathparent\_id = ppct.pathparent\_id

and hcpc.top\_disp = 'N')));

COMMIT WORK;

sql\_stmt:= 'truncate table high\_capacity\_path\_chan\_child';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in LOAD\_HIGH\_CAP\_PATH\_CHAN: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

insert into NG\_REPORTS.high\_capacity\_path\_chan\_child

--get all distinct path\_chan parents with current children and pending if there is no current

with

circuit\_path as

(select p.TRAIL\_ID, p.TRAIL\_NAME,p.bandwidth, p.status from

ng\_topology.TRAIL p,

ng\_reports.SITE s1,

ng\_reports.SITE s2,

ng\_topology.TRAIL p1,

ng\_topology.TRAIL p2,

ng\_topology.DIR\_TRAIL\_TOPOLOGY t4

where

p.a\_site\_name = s1.site\_reference\_id (+) and

p.z\_site\_name = s2.site\_reference\_id (+) and

p.PREVIOUS\_REVISION\_TRAIL\_ID = p1.TRAIL\_ID (+) and

p.NEXT\_REVISION\_TRAIL\_ID = p2.TRAIL\_ID (+) and

p.topology = t4.CODE(+)),

pathchan as (

select path\_chan\_inst\_id, parent\_path\_inst\_id, channel\_name

from (select distinct nvl(pc.member\_trail\_id,pc.next\_trail\_id) as path\_chan\_inst\_id,

pc.parent\_trail\_id parent\_path\_inst\_id, pc.channel\_name

from ng\_topology.trail\_channel pc,

(select distinct trail\_id

from circuit\_path

where status not in ('DECOMMISSIONED','CANCELLED')

and (circuit\_path.bandwidth = 'DS3' or circuit\_path.bandwidth like 'OC%')) cp

where pc.parent\_trail\_id = cp.trail\_id)

where path\_chan\_inst\_id is not null)

, pchild as (

select distinct pc.path\_chan\_inst\_id circ\_path\_inst\_id, c.circ\_path\_hum\_id, c.path\_type,

c.status, c.bandwidth, c.nbr\_chan\_assigned, c.nbr\_channels,

pc.parent\_path\_inst\_id, pc.channel\_name

from pathchan pc,

(select TRAIL\_ID circ\_path\_inst\_id, TRAIL\_NAME circ\_path\_hum\_id, type as path\_type, status, bandwidth,

nvl(TOTAL\_CHANNELS\_USED,0) as nbr\_chan\_assigned, TOTAL\_CHANNELS nbr\_channels

from NG\_TOPOLOGY.TRAIL

where status <> 'CANCELLED'

and bandwidth <> 'AGGREGATE') c

where pc.path\_chan\_inst\_id = c.circ\_path\_inst\_id (+)),

seginfo as (select nvl(current\_trail\_id, next\_trail\_id) circ\_path\_inst\_id,

segment\_id, nvl(recur\_costs,0) recur\_costs

from NG\_TOPOLOGY.TRAIL\_SEGMENT

where status not in ('DECOMMISSIONED','CANCELLED')

and (current\_trail\_id is not null or next\_trail\_id is not null))

, pchild\_seg as (

select circ\_path\_inst\_id, circ\_path\_hum\_id, path\_type, status, bandwidth,

nbr\_chan\_assigned, nbr\_channels, parent\_path\_inst\_id, channel\_name,

trans\_act, est\_comp, act\_just, act\_date, est\_comp\_grp, act\_tot\_mrc,

case

when status = 'DECOMMISSIONED' then 0

else tot\_mrc

end tot\_mrc, est\_comp\_days, act\_just\_comt

from (select c.circ\_path\_inst\_id, c.circ\_path\_hum\_id, c.path\_type, c.status,

c.bandwidth, c.nbr\_chan\_assigned, c.nbr\_channels,

sum(s.recur\_costs) tot\_mrc, c.parent\_path\_inst\_id, c.channel\_name,

ta.trans\_act, ta.est\_comp, ta.act\_just, ta.act\_date,

ta.est\_comp\_grp, ta.act\_tot\_mrc,

case

when ta.est\_comp is not null and ta.est\_comp\_days > 0 then

ta.est\_comp\_days

when ta.est\_comp is not null and ta.est\_comp\_days <= 0 then

0

else

nvl(trunc(sysdate - nvl(ta.act\_date,sysdate)),0)

end est\_comp\_days, ta.act\_just\_comt

from pchild c, seginfo s, ng\_reports.high\_capacity\_path\_chan\_act ta

where c.circ\_path\_inst\_id = s.circ\_path\_inst\_id (+)

and c.circ\_path\_inst\_id = ta.trail\_id (+)

and c.bandwidth <> 'AGGREGATE'

group by c.circ\_path\_inst\_id, c.circ\_path\_hum\_id, c.path\_type, c.status,

c.bandwidth, c.nbr\_chan\_assigned, c.nbr\_channels,

c.parent\_path\_inst\_id, c.channel\_name, ta.trans\_act,

ta.est\_comp, ta.act\_just, ta.act\_date, ta.est\_comp\_grp,

ta.act\_tot\_mrc, ta.est\_comp\_days, ta.act\_just\_comt))

select circ\_path\_inst\_id, path\_name, path\_type, status, bandwidth, nbr\_channels,

nbr\_chan\_assigned, tot\_mrc, path\_type\_ck, parent\_path\_inst\_id, channel\_name, trans\_act,

est\_comp, act\_just, act\_date, est\_comp\_grp, act\_tot\_mrc, est\_comp\_days, act\_just\_comt

from (select circ\_path\_inst\_id, circ\_path\_hum\_id path\_name, path\_type, status, bandwidth, nbr\_channels,

nbr\_chan\_assigned, nvl(tot\_mrc,0) tot\_mrc, nvl(path\_type,'NULL') path\_type\_ck,

parent\_path\_inst\_id, channel\_name, trans\_act, est\_comp, act\_just, act\_date, est\_comp\_grp,

act\_tot\_mrc, est\_comp\_days, act\_just\_comt

from pchild\_seg);

commit;

COMMIT WORK;

--take a back up of the util table

sql\_stmt:= 'truncate table high\_capacity\_path\_chan\_util\_b';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in LOAD\_HIGH\_CAP\_PATH\_CHAN: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

insert into NG\_REPORTS.high\_capacity\_path\_chan\_util\_b

(select \* from NG\_REPORTS.high\_capacity\_path\_chan\_util); commit;

--reload util table only changing bps data when it has changed so historical data retained and only

--current parent paths included

sql\_stmt:= 'truncate table high\_capacity\_path\_chan\_util';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in LOAD\_HIGH\_CAP\_PATH\_CHAN: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

insert into ng\_reports.high\_capacity\_path\_chan\_util

(select n.TRAIL\_ID pathparent\_id,

case

when n.TOTAL\_CAPACITY\_AVL <> p.TOTAL\_CAPACITY\_AVL or n.TOTAL\_CAPACITY\_USED <> p.TOTAL\_CAPACITY\_USED

then n.TOTAL\_CAPACITY\_AVL

else p.TOTAL\_CAPACITY\_AVL

end TOTAL\_CAPACITY\_AVL,

case

when n.TOTAL\_CAPACITY\_AVL <> p.TOTAL\_CAPACITY\_AVL or n.TOTAL\_CAPACITY\_USED <> p.TOTAL\_CAPACITY\_USED

then n.TOTAL\_CAPACITY\_USED

else p.TOTAL\_CAPACITY\_USED

end TOTAL\_CAPACITY\_USED,

case

when n.TOTAL\_CAPACITY\_AVL <> p.TOTAL\_CAPACITY\_AVL or n.TOTAL\_CAPACITY\_USED <> p.TOTAL\_CAPACITY\_USED

then n.bps\_util

else p.bps\_util

end bps\_util,

case

when n.TOTAL\_CAPACITY\_AVL <> p.TOTAL\_CAPACITY\_AVL or n.TOTAL\_CAPACITY\_USED <> p.TOTAL\_CAPACITY\_USED

then n.updated\_by

else p.last\_mod\_by

end last\_mod\_by,

case

when n.TOTAL\_CAPACITY\_AVL <> p.TOTAL\_CAPACITY\_AVL or n.TOTAL\_CAPACITY\_USED <> p.TOTAL\_CAPACITY\_USED

then n.last\_mod\_ts

else p.last\_mod\_ts

end last\_mod\_ts

from (select a.TRAIL\_ID, nvl(a.TOTAL\_CAPACITY\_AVL,0) TOTAL\_CAPACITY\_AVL, nvl(a.TOTAL\_CAPACITY\_USED, 0) TOTAL\_CAPACITY\_USED,

trunc(round((nvl(a.TOTAL\_CAPACITY\_USED,0) \* 100)/nvl(a.TOTAL\_CAPACITY\_AVL,0), 2)) bps\_util,

a.updated\_by, trunc(a.updated\_timestamp) last\_mod\_ts

from ng\_topology.TRAIL a, ng\_reports.high\_capacity\_path\_chan\_parent b

where a.TRAIL\_ID = b.pathparent\_id) n,

(select a.TRAIL\_ID, nvl(b.TOTAL\_CAPACITY\_AVL,0) TOTAL\_CAPACITY\_AVL, nvl(b.TOTAL\_CAPACITY\_USED,0) TOTAL\_CAPACITY\_USED,

nvl(b.bps\_util,0) bps\_util, b.last\_mod\_by, b.last\_mod\_ts

from ng\_topology.TRAIL a, ng\_reports.high\_capacity\_path\_chan\_util\_b b

where a.TRAIL\_ID = b.pathparent\_id (+)) p

where n.TRAIL\_ID = p.TRAIL\_ID (+));

commit;

sql\_stmt:= 'truncate table high\_capacity\_path\_chan';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in LOAD\_HIGH\_CAP\_PATH\_CHAN: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

insert into NG\_REPORTS.high\_capacity\_path\_chan

--combine parent and child information even if there is no child associated to parent

select distinct domain\_id, domain\_name, territory, market, sub\_market, a\_site\_name,

z\_site\_name, pathparent\_name, pathparent\_id, pathparent\_type, pathparent\_status,

pathparent\_bandwidth, pathparent\_nchl, pathparent\_nchla, p\_tot\_mrc,

path\_name, trail\_id, path\_type, status, bandwidth, nbr\_channels,

nbr\_chan\_assigned, tot\_mrc, p\_pct\_asgn, p\_pct\_grp, sysdate, path\_type\_ck,

channel\_used, total\_capacity\_avl, total\_capacity\_used, bps\_util,

bps\_util\_prev, bps\_util\_last\_mod\_ts, p\_trans\_act, p\_est\_comp,

p\_act\_just, p\_act\_date, p\_est\_comp\_grp, trans\_act, est\_comp,

act\_just, act\_date, est\_comp\_grp, p\_act\_tot\_mrc, act\_tot\_mrc,

p\_top\_ar, p\_top\_arbw, est\_comp\_days,

p\_est\_comp\_days, p\_act\_just\_comt, act\_just\_comt

from (select ps.domain\_id, ps.domain\_name, ps.territory, ps.market, ps.sub\_market, ps.a\_site\_name,

ps.z\_site\_name, ps.pathparent\_name, ps.pathparent\_id, ps.pathparent\_type, ps.pathparent\_status,

ps.pathparent\_bandwidth, ps.pathparent\_nchl, ps.pathparent\_nchla, nvl(ps.p\_tot\_mrc,0) p\_tot\_mrc,

cs.path\_name, cs.trail\_id, cs.path\_type, cs.status, cs.bandwidth, cs.nbr\_channels,

cs.nbr\_chan\_assigned, nvl(cs.tot\_mrc,0) tot\_mrc, ps.p\_pct\_asgn, ps.p\_pct\_grp, cs.path\_type\_ck,

cs.channel\_used, ps.total\_capacity\_avl, ps.total\_capacity\_used, ps.bps\_util,

nvl(b.bps\_util,0) bps\_util\_prev, u.last\_mod\_ts bps\_util\_last\_mod\_ts, ps.p\_trans\_act, ps.p\_est\_comp,

ps.p\_act\_just, ps.p\_act\_date, ps.p\_est\_comp\_grp, cs.trans\_act, cs.est\_comp,

cs.act\_just, cs.act\_date, cs.est\_comp\_grp, ps.p\_act\_tot\_mrc, cs.act\_tot\_mrc,

ps.p\_top\_ar, ps.p\_top\_arbw, nvl(cs.est\_comp\_days,0) est\_comp\_days,

nvl(ps.p\_est\_comp\_days,0) p\_est\_comp\_days, ps.p\_act\_just\_comt, cs.act\_just\_comt

from NG\_REPORTS.high\_capacity\_path\_chan\_parent ps,

NG\_REPORTS.high\_capacity\_path\_chan\_child cs,

NG\_REPORTS.high\_capacity\_path\_chan\_util u,

NG\_REPORTS.high\_capacity\_path\_chan\_util\_b b

where ps.pathparent\_id = cs.PARENT\_PATH\_INST\_ID (+)

and ps.pathparent\_id = u.pathparent\_id (+)

and ps.pathparent\_id = b.pathparent\_id (+));

COMMIT;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put(SQLCODE);

dbms\_output.put(SQLERRM);

--raise;

END; -- Procedure

/

--------------------------------------------------------

-- DDL for Procedure LOAD\_NGMLS\_REGIONAL\_SUMMARY

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."LOAD\_NGMLS\_REGIONAL\_SUMMARY" AS

-- get the all ngMLSs found in NCM ngMLS Extract

cursor ngMLSs\_NCM is

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6,territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id

from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> 'NNO' AND sub\_market <> 'OSS'

)

,NCM as(

select distinct nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market,hostname

from ncm\_ngMLS\_wk ncm--, sam\_ebh\_health sam

left outer join DOMAINS di

on substr(hostname, 1, 6) = di.clli\_6

-- group by rollup(territory, market\_territory, sub\_market)

)

,SAM as (

select distinct nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, SITE\_NAME

from sam\_ebh\_health sam

left outer join DOMAINS di

on substr(sam.SITE\_NAME, 1, 6) = di.clli\_6

where site\_name LIKE '%7750-0\_'

-- group by rollup(territory, market\_territory, sub\_market)

)

select territory, market\_territory, sub\_market

, count (hostname) ncm\_ngMLSs

from NCM

group by rollup(territory, market\_territory, sub\_market)

union

select territory, market\_territory, sub\_market

, count(SITE\_NAME) ncm\_ngMLSs

from SAM

group by rollup(territory, market\_territory, sub\_market)

;

-- get the all ngMLSs found in NCM ngMLS-VLAN Extract

cursor ngMLSs\_VLAN\_NCM is

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6,territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> 'NNO' AND sub\_market <> 'OSS'

)

,

NCM as

(

select distinct nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, hostname

from ncm\_ngMLS\_vlan\_wk ncm

left outer join DOMAINS di

on substr(hostname, 1, 6) = di.clli\_6

)

select territory, market\_territory, sub\_market, count (1) ncm\_ngMLSs

from NCM

group by rollup(territory, market\_territory, sub\_market)

;

-- make sure duplicate matches in GI are not considered as good match-------??????

-- DUP means > 1 Live ones in ng

-- if 1 is live and others are not, then that is NOT a DIP. Shirley's logic

-- this should be identifiable from the audit

-- get mismatches between ncm-ngMLS-vlan extract and gi

CURSOR ng\_mismatch\_ngMLSs is

WITH

--NCM GI mismatch details

SAM\_NCM\_NG\_MISMATCH as

(

-- any issue with the device is a mismatch

select distinct cdai.ngMLS\_device\_name -- distinct

from ngMLS\_device\_AUDIT\_ISSUEs\_wk cdai

where ngMLS\_vendor IN ('CISCO', 'ALCATEL-LUCENT')

)

,

GUD\_MATCHES as

(

-- LR 11/07

select ngMLS\_device\_name

from ngMLS\_equip\_audit\_wk aud

where match\_code <> 'Nautilus Only'

minus

select cdai.ngMLS\_device\_name

from ngMLS\_device\_AUDIT\_ISSUEs\_wk cdai, ngmls\_issue n

where ngMLS\_vendor IN ('CISCO', 'ALCATEL-LUCENT')

and N.IS\_CRITICAL='Y'

and N.NGMLS\_ISSUE\_ID = CDAI.NGMLS\_ISSUE\_ID

)

,

SAM\_NCM\_NG\_MATCH as

(

select aud.ngMLS\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end SAM\_NCM\_NG\_L\_MATCH

, case when live\_in\_ng <> 'Y' then 1 else 0 end SAM\_NCM\_NG\_NL\_MATCH

from ngMLS\_equip\_audit\_wk aud

join GUD\_MATCHES gud

on gud.ngMLS\_device\_name = aud.ngMLS\_device\_name

)

,

MATCH\_COUNTS AS

(

select

case when ma.ngMLS\_device\_name is not null then substr(ma.ngMLS\_device\_name, 1, 6)

when ms.ngMLS\_device\_name is not null then substr(ms.ngMLS\_device\_name, 1, 6)

end clli\_6

, count(ma.ngMLS\_device\_name) SAM\_NCM\_NG\_MATCH

, sum(ma.SAM\_NCM\_NG\_L\_MATCH) SAM\_NCM\_NG\_L\_MATCH

, sum(ma.SAM\_NCM\_NG\_NL\_MATCH) SAM\_NCM\_NG\_NL\_MATCH

, count(ms.ngMLS\_device\_name) SAM\_NCM\_NG\_MISMATCH

from SAM\_NCM\_NG\_MATCH ma

full outer join SAM\_NCM\_NG\_MISMATCH ms

on ma.ngMLS\_device\_name = ms.ngMLS\_device\_name

group by

case when ma.ngMLS\_device\_name is not null then substr(ma.ngMLS\_device\_name, 1, 6)

when ms.ngMLS\_device\_name is not null then substr(ms.ngMLS\_device\_name, 1, 6)

end

)

,

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6,territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> 'NNO' AND sub\_market <> 'OSS'

)

select territory, market\_territory, sub\_market

, sum(SAM\_NCM\_NG\_MATCH) SAM\_NCM\_NG\_MATCH

, sum(SAM\_NCM\_NG\_L\_MATCH) SAM\_NCM\_NG\_L\_MATCH

, sum(SAM\_NCM\_NG\_NL\_MATCH) SAM\_NCM\_NG\_NL\_MATCH

, sum(SAM\_NCM\_NG\_MISMATCH) SAM\_NCM\_NG\_MISMATCH

from MATCH\_COUNTS cnt

join DOMAINS di

on cnt.clli\_6 = di.clli\_6

group by rollup(territory, market\_territory, sub\_market)

;

methodName varchar2(30) := 'load\_ngMLS\_regional\_summary ';

message varchar2(300);

updStmt varchar2(32767);

updAreaStmt varchar2(32767);

updVzwStmt varchar2(32767);

insStmt varchar2(32767);

insAreaStmt varchar2(32767);

insVzwStmt varchar2(32767);

sqlStmt varchar2(32767);

l\_cnt number;

BEGIN

dbms\_output.put\_line('FYI: In '||methodName);

sqlStmt := ' delete from ngMLS\_region\_summ ';

execute immediate sqlStmt;

commit;

BEGIN

-- from ng insert all CLLI and ng# in the table

sqlStmt := 'insert into ngMLS\_region\_summ columns (territory, market\_territory, sub\_market, NG\_NGMLSS, ng\_l\_ngMLSs, ng\_nl\_ngMLSs)

WITH

DOMAINS AS

(

select distinct territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id, substr(clli, 1, 6) clli\_6

from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> ''NNO'' AND sub\_market <> ''OSS''

union

select ''unknown'' territory, ''unknown'' market\_territory, ''unknown'' sub\_market, ''unknown'' sub\_market\_leaf, null leaf\_domain\_inst\_id, ''unknown'' clli\_6

from dual

)

,

ne\_name as(

select case when REGEXP\_SUBSTR(eqp\_name, ''([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}'') is not null then

REGEXP\_SUBSTR(eqp\_name, ''([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}'')

else eqp\_name end

ngMLS\_device\_name

, ngMLS.eqp\_reference\_id

, nvl(case when inv\_status = ''LIVE'' then 1 else 0 end, 0) l

, nvl(case when inv\_status <> ''LIVE'' then 1 else 0 end, 0) nl

, inv\_status

from equipment ngMLS

where eqp\_vendor IN (''CISCO'',''ALCATEL-LUCENT'') and

ngMLS.eqp\_type = ''NGMLS''

)

,

ng as( select distinct ngMLS\_device\_name

, substr(ngMLS\_device\_name, 1, 6) clli\_6 , ngMLS.eqp\_reference\_id

, ngMLS.l

, ngMLS.nl

from ne\_name ngMLS

)

,

GI\_CNTS as

(

select nvl(territory, ''unknown'') territory, nvl(market\_territory, ''unknown'') market\_territory, nvl(sub\_market, ''unknown'') sub\_market, case when di.clli\_6 is null then ''unknown'' else ng.clli\_6 end clli\_6

, sum(l) l

, sum(nl) nl

from DOMAINS di

right outer join ng

on ng.clli\_6 = di.clli\_6

group by nvl(territory, ''unknown''), nvl(market\_territory, ''unknown''), nvl(sub\_market, ''unknown'')

, case when di.clli\_6 is null then ''unknown'' else ng.clli\_6 end

)

select di.territory, di.market\_territory, di.sub\_market

, sum(nvl(l, 0)) + sum (nvl(nl, 0)) ng\_ngMLS

, sum(l) ng\_l\_ngMLSs

, sum(nl) ng\_nl\_ngMLSs

from DOMAINS di

left outer join GI\_CNTS ng

on ng.clli\_6 = di.clli\_6

group by rollup(di.territory, di.market\_territory, di.sub\_market)

order by di.territory, di.market\_territory, di.sub\_market

';

execute immediate sqlStmt;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '

||' GI metrics in table: ngMLS\_region\_summ';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

-- insert new or update metric in the same table for total\_ngMLSs found in NCM in the ngMLS-extract

updStmt := 'update ngMLS\_region\_summ

set ncm\_ngMLSs = :ncm\_ngMLSs

where territory = :territory and market\_territory = :market\_territory

and sub\_market = :sub\_market

';

updAreaStmt := 'update ngMLS\_region\_summ

set ncm\_ngMLSs = :ncm\_ngMLSs

where territory = :territory and market\_territory = :market\_territory

and sub\_market is null

';

updVzwStmt := 'update ngMLS\_region\_summ

set ncm\_ngMLSs = :ncm\_ngMLSs

where territory is null and market\_territory is null

and sub\_market is null

';

for rec in ngMLSs\_NCM

loop

BEGIN

-- dbms\_output.put\_line(rec.territory ||'/'|| rec.market\_territory ||'/'|| rec.sub\_market||'/'||rec.ncm\_ngMLSs);

if rec.territory is null then

-- dbms\_output.put\_line('In ngMLSs\_NCM VZW');

execute immediate updVzwStmt using rec.ncm\_ngMLSs;

else

if rec.sub\_market is null then

-- dbms\_output.put\_line('In ngMLSs\_NCM territory');

execute immediate updAreaStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory;

else

-- dbms\_output.put\_line('In ngMLSs\_NCM sub\_market');

execute immediate updStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory, rec.sub\_market;

end if;

end if;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '

|| rec.territory ||'/'|| rec.market\_territory||'/'|| rec.sub\_market||'/'||rec.ncm\_ngMLSs

||' for NCM-ngMLS-extract in table: ngMLS\_region\_summ';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

commit;

-- insert new or update metric in the same table for ncm total\_ngMLSs

updStmt := 'update ngMLS\_region\_summ

set SAM\_NCM\_NGMLSS\_W\_VLAN = :SAM\_NCM\_NGMLSS\_W\_VLAN

where territory = :territory and market\_territory = :market\_territory

and sub\_market = :sub\_market

';

updAreaStmt := 'update ngMLS\_region\_summ

set SAM\_NCM\_NGMLSS\_W\_VLAN = :SAM\_NCM\_NGMLSS\_W\_VLAN

where territory = :territory and market\_territory = :market\_territory

and sub\_market is null

';

updVzwStmt := 'update ngMLS\_region\_summ

set SAM\_NCM\_NGMLSS\_W\_VLAN = :SAM\_NCM\_NGMLSS\_W\_VLAN

where territory is null and market\_territory is null

and sub\_market is null

';

for rec in ngMLSs\_VLAN\_NCM

loop

BEGIN

if rec.territory is null then execute immediate updVzwStmt using rec.ncm\_ngMLSs;

else

if rec.sub\_market is null then execute immediate updAreaStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory;

else execute immediate updStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory, rec.sub\_market;

end if;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '

|| rec.territory ||'/'|| rec.market\_territory||'/'|| rec.sub\_market||'/'|| rec.ncm\_ngMLSs

||' for NCM-ngMLS-VLAN-extract in table: ngMLS\_region\_summ';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

-- calc matches / mismatches metric between ncm and gi

updStmt := 'update ngMLS\_region\_summ

set SAM\_NCM\_NG\_MISMATCH = :SAM\_NCM\_NG\_MISMATCH

, SAM\_NCM\_NG\_MATCH = :SAM\_NCM\_NG\_MATCH

, SAM\_NCM\_NG\_L\_MATCH = :SAM\_NCM\_NG\_L\_MATCH

, SAM\_NCM\_NG\_NL\_MATCH = :SAM\_NCM\_NG\_NL\_MATCH

where territory = :territory and market\_territory = :market\_territory

and sub\_market = :sub\_market

';

updAreaStmt := 'update ngMLS\_region\_summ

set SAM\_NCM\_NG\_MISMATCH = :SAM\_NCM\_NG\_MISMATCH

, SAM\_NCM\_NG\_MATCH = :SAM\_NCM\_NG\_MATCH

, SAM\_NCM\_NG\_L\_MATCH = :SAM\_NCM\_NG\_L\_MATCH

, SAM\_NCM\_NG\_NL\_MATCH = :SAM\_NCM\_NG\_NL\_MATCH

where territory = :territory and market\_territory = :market\_territory

and sub\_market is null

';

updVzwStmt := 'update ngMLS\_region\_summ

set SAM\_NCM\_NG\_MISMATCH = :SAM\_NCM\_NG\_MISMATCH

, SAM\_NCM\_NG\_MATCH = :SAM\_NCM\_NG\_MATCH

, SAM\_NCM\_NG\_L\_MATCH = :SAM\_NCM\_NG\_L\_MATCH

, SAM\_NCM\_NG\_NL\_MATCH = :SAM\_NCM\_NG\_NL\_MATCH

where territory is null and market\_territory is null

and sub\_market is null

';

for rec in ng\_mismatch\_ngMLSs

loop

BEGIN

if rec.territory is null then

execute immediate updVzwStmt using rec.SAM\_NCM\_NG\_MISMATCH, rec.SAM\_NCM\_NG\_MATCH, rec.SAM\_NCM\_NG\_L\_MATCH, rec.SAM\_NCM\_NG\_NL\_MATCH;

else

if rec.sub\_market is null then

execute immediate updAreaStmt using rec.SAM\_NCM\_NG\_MISMATCH, rec.SAM\_NCM\_NG\_MATCH, rec.SAM\_NCM\_NG\_L\_MATCH, rec.SAM\_NCM\_NG\_NL\_MATCH, rec.territory, rec.market\_territory;

else

execute immediate updStmt using rec.SAM\_NCM\_NG\_MISMATCH, rec.SAM\_NCM\_NG\_MATCH, rec.SAM\_NCM\_NG\_L\_MATCH, rec.SAM\_NCM\_NG\_NL\_MATCH, rec.territory, rec.market\_territory, rec.sub\_market;

end if;

end if;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update NCM-NG mismatch in table: ngMLS\_region\_summ';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

END LOOP;

commit;

-- calculate the percentages

BEGIN

update ngMLS\_region\_summ

set SAM\_NCM\_NGMLSS\_WO\_VLAN = nvl(ncm\_ngMLSs,0) - nvl(SAM\_NCM\_NGMLSS\_W\_VLAN,0)

,

SAM\_NCM\_NGMLSS\_W\_VLAN\_PER = round(nvl(SAM\_NCM\_NGMLSS\_W\_VLAN, 0) / decode (nvl(ncm\_ngMLSs,0), 0, 1, nvl(ncm\_ngMLSs,0)) \* 100, 2)

, ncm\_gi\_per = round(nvl(SAM\_NCM\_NG\_MATCH, 0) \* 2 / decode ((nvl(ncm\_ngMLSs,0) + nvl(NG\_NGMLSS,0)), 0, 1, nvl(ncm\_ngMLSs,0) + nvl(ng\_ngMLSs,0)) \* 100, 2)

;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update percentages in table: ngMLS\_region\_summ';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

END LOAD\_NGMLS\_REGIONAL\_SUMMARY;

/

--------------------------------------------------------

-- DDL for Procedure LOAD\_NTLS\_NGMLS\_VLAN\_PATHS

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."LOAD\_NTLS\_NGMLS\_VLAN\_PATHS" AS

sqlStmt varchar2(32000);

message varchar2(250);

methodName varchar2(30) := 'load\_ntls\_ngmls\_vlan\_paths';

BEGIN

message := 'Error: '|| methodName ||'(): Can''t truncate table: NG\_REPORTS.ng\_ngmls\_vlan\_paths\_wk';

sqlStmt := 'truncate table NG\_REPORTS.ng\_ngmls\_vlan\_paths\_wk' ;

execute immediate sqlStmt;

message := '';

-- this query looks at only the 1st leg of the VLAN

-- and 1st leg of the 1Gig and 10Gig path

sqlStmt:='

INSERT INTO ng\_ngmls\_vlan\_paths\_wk columns (

eqp\_reference\_id, --ne\_inst\_id, removing EQUIP\_INST\_ID

card\_reference\_id,

ckt\_path\_reference\_id,

port\_bandwidth,

path\_type,

vlan\_reference\_id,

vlan\_name,

vlan\_number,

vlan\_status,

extract\_date

)

WITH ngmls\_gige\_paths AS (

SELECT DISTINCT

xne.eqp\_reference\_id, --xne

--xne.equip\_inst\_id,

p.card\_reference\_id, --p.CARD\_INST\_ID --, p.port\_inst\_id, pli.leg\_inst\_id,(pli.rel\_order), PLI.LEG\_NAME

cpi.trail\_id gige\_inst\_id, --cpi.circ\_path\_inst\_id gige\_inst\_id

cpi.bandwidth,

cpi.type

FROM

ng\_reports.equipment xne, --xng\_reports.xng\_ngMLS\_equip\_wk xne

ng\_reports.PORT p, --vzwnet.epa p,

ng\_topology.trail cpi, --vzwnet.circ\_path\_inst cpi,

ng\_topology.trail\_component pli, --vzwnet.path\_leg\_inst pli

ng\_topology.trail\_component\_element plm, --vzwnet.path\_leg\_member plm

ng\_topology.trail\_element cpe --vzwnet.circ\_path\_element cpe

WHERE

p.eqp\_reference\_id = xne.eqp\_reference\_id --p.equip\_inst\_id = xne.equip\_inst\_id

AND xne.eqp\_type = ''NGMLS''

AND cpi.trail\_id IN (

p.ckt\_path\_reference\_id, --p.circ\_path\_inst\_id

p.next\_path\_reference\_Id --p.next\_path\_inst\_id

)

AND cpi.bandwidth LIKE ''%Gbps''

AND pli.TRAIL\_ID = cpi.trail\_id --pli.circ\_path\_inst\_id = cip.circ\_path\_inst\_id

AND pli.component\_id = plm.component\_id --pli.leg\_inst\_id = plm.leg\_inst\_id

-- and plm.sequence = cpe.sequence

AND plm.element\_id = cpe.element\_id --plm.element\_inst\_id = cpe.element\_inst\_id \*\*confirm with KUMAR MANI

AND cpe.element\_ref\_id = p.port\_reference\_id AND cpe.element\_type=''E'' --CPE.PORT\_INST\_ID = p.port\_inst\_id

AND pli.sequence = (

SELECT

MIN(sequence) sequence

FROM

ng\_topology.trail\_component li --vzwnet.path\_leg\_inst li

WHERE

li.trail\_id = cpi.trail\_id --LI.CIRC\_PATH\_INST\_ID=cpi.circ\_path\_inst\_id

)

) --select count(\*) from ngmls\_gige\_paths

, gige\_vlan\_paths AS (

SELECT DISTINCT

gige.trail\_id gige\_inst\_id, --gige.circ\_path\_inst\_id gige\_inst\_id,

vlan.trail\_id vlan\_inst\_id, --vlan.circ\_path\_inst\_id vlan\_inst\_id,

vlan.trail\_name vlan\_hum\_id, --vlan.circ\_path\_hum\_id vlan\_hum\_id,

regexp\_substr(regexp\_substr(vlan.trail\_name,''VLAN(-|[[:space:]]|)\*[[:digit:]]+''),''[[:digit:]]+'') vlan\_number, --vlan.circ\_path\_hum\_id

vlan.status

FROM

ng\_topology.trail gige, --vzwnet.circ\_path\_inst gige,

ng\_topology.trail vlan, --vzwnet.circ\_path\_inst vlan,

ng\_topology.trail\_component pli, --vzwnet.path\_leg\_inst pli,

ng\_topology.trail\_component\_element plm, --vzwnet.path\_leg\_member plm,

ng\_topology.trail\_element cpe --vzwnet.circ\_path\_element cpe

WHERE

gige.bandwidth LIKE ''%Gbps''

AND vlan.bandwidth = ''VLAN''

AND vlan.type = ''EBH''

AND vlan.trail\_id = cpe.trail\_id --vlan.circ\_path\_inst\_id = cpe.circ\_path\_inst\_id

AND gige.trail\_id = cpe.parent\_trail\_id AND cpe.element\_type=''P'' --gige.circ\_path\_inst\_id = cpe.path\_inst\_id

AND vlan.trail\_id = pli.trail\_id --vlan.circ\_path\_inst\_id = pli.circ\_path\_inst\_id

AND pli.component\_id = plm.component\_id --pli.leg\_inst\_id = plm.leg\_inst\_id

--and plm.sequence = cpe.sequence

AND plm.element\_id = cpe.element\_id --plm.element\_inst\_id = cpe.element\_inst\_id

AND pli.sequence = ( --pli.rel\_order = (

SELECT

MIN(sequence)

FROM

ng\_topology.trail\_component pl --vzwnet.path\_leg\_inst pl

WHERE

pl.trail\_id = vlan.trail\_id --pl.circ\_path\_inst\_id = vlan.circ\_path\_inst\_id

)

) --select count(\*) from gige\_vlan\_paths

SELECT

ngp.eqp\_reference\_id,

--ngp.equip\_inst\_id,

ngp.card\_reference\_id,

ngp.gige\_inst\_id trail\_id, --circ\_path\_inst\_id

ngp.bandwidth,

ngp.type,

gvp.vlan\_inst\_id,

gvp.vlan\_hum\_id,

gvp.vlan\_number,

gvp.status vlan\_status,

SYSDATE

--count(ngp.card\_reference\_id)

FROM

ngmls\_gige\_paths ngp,

gige\_vlan\_paths gvp

WHERE

ngp.gige\_inst\_id = gvp.gige\_inst\_id

';

execute immediate sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr(message||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END LOAD\_NTLS\_NGMLS\_VLAN\_PATHS;

/

--------------------------------------------------------

-- DDL for Procedure LOAD\_SEGMENTS\_STRIPPED

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."LOAD\_SEGMENTS\_STRIPPED"

IS

--

-- Purpose: Update LOAD\_SEGMENTS\_STRIPPED

-- NOTE: This procedure is runs as part of

-- NG\_REPORTS.BTOD\_AUDIT and NG\_REPORTS.BTOD\_UPDATE and NG\_REPORTS.BTOD\_PP\_V

--

-- MODIFICATION HISTORY

-- Person Date Comments

-- Sayre 10/11/17

-- --------- ------ -------------------------------------------

--!!!!! This is also set up in NIS\_EXTRACTOR rule\_id 241 and runBtodPPDiscovery.sh in dev

--left it here because it is used by several things

v\_cnt number;

BEGIN

execute immediate 'truncate table ng\_reports.stripped\_segment\_data reuse storage'; commit;

insert into ng\_reports.stripped\_segment\_data

(select distinct a.segment\_id, a.seg\_name, a.n\_strip\_seg\_name, n\_vendor, a.n\_vendor\_ck,

n\_status, n\_ban, a.n\_ban\_ck, n\_amt, n\_amt\_ck, 'Y','Y', 'Y',

to\_date(attribute\_value, 'dd-MON-yyyy') n\_bill\_date,

attribute\_value n\_bill\_date\_chr,

a.n\_type, a.n\_bandwidth, a.current\_trail\_id, a.next\_trail\_id, sysdate

from

(select segment\_id, name seg\_name, vendor as n\_vendor, status as n\_status,

ng\_reports.strip\_segment\_name(name) AS n\_strip\_seg\_name, billing\_code as n\_ban,

replace(translate(nvl(billing\_code,'0'), '-/,.()[]\*$;?#&~:=`<>\^@!"'||chr(160), ' '), ' ', '') as n\_ban\_ck,

recur\_costs as n\_amt, nvl(recur\_costs,0) as n\_amt\_ck,

NVL (SUBSTR (vendor, 1, INSTR (vendor, ' ') - 1), vendor) AS n\_vendor\_ck,

bandwidth AS n\_bandwidth, type AS n\_type, current\_trail\_id, next\_trail\_id

from ng\_topology.trail\_segment

) a,

(select segment\_id, attribute\_value

from ng\_topology.segment\_attributes

where attribute\_name = 'Bill Date'

and attribute\_value like '%-%-%') b

where a.segment\_id = b.segment\_id (+));

commit;

update ng\_reports.stripped\_segment\_data cd

set ntls\_seg\_vendor\_ban\_unq = 'N'

where exists (select 1

from (select ntls\_seg\_name\_strip, ntls\_vendor\_ck, ntls\_ban\_ck

from ng\_reports.segment\_btod\_vw

where ntls\_ban\_ck is not null

and ntls\_vendor\_ck is not null

and ntls\_seg\_name\_strip is not null

group by ntls\_seg\_name\_strip, ntls\_vendor\_ck, ntls\_ban\_ck

having count(\*) > 1) cv

where cd.ntls\_seg\_name\_strip = cv.ntls\_seg\_name\_strip

and cd.ntls\_vendor\_strip = cv.ntls\_vendor\_ck

and cd.ntls\_ban\_strip = cv.ntls\_ban\_ck);

commit;

update ng\_reports.stripped\_segment\_data cd

set ntls\_seg\_vendor\_unq = 'N'

where exists (select 1

from (select ntls\_seg\_name\_strip, ntls\_vendor\_ck, ntls\_ban\_ck

from ng\_reports.segment\_btod\_vw

where ntls\_ban\_ck is not null

and ntls\_vendor\_ck is not null

and ntls\_seg\_name\_strip is not null

group by ntls\_seg\_name\_strip, ntls\_vendor\_ck, ntls\_ban\_ck

having count(\*) > 1) cv

where cd.ntls\_seg\_name\_strip = cv.ntls\_seg\_name\_strip

and cd.ntls\_vendor\_strip = cv.ntls\_vendor\_ck);

commit;

update ng\_reports.stripped\_segment\_data cd

set ntls\_seg\_unq = 'N'

where exists (select 1

from (select ntls\_seg\_name\_strip, ntls\_vendor\_ck, ntls\_ban\_ck

from ng\_reports.segment\_btod\_vw

where ntls\_ban\_ck is not null

and ntls\_vendor\_ck is not null

and ntls\_seg\_name\_strip is not null

group by ntls\_seg\_name\_strip, ntls\_vendor\_ck, ntls\_ban\_ck

having count(\*) > 1) cv

where cd.ntls\_seg\_name\_strip = cv.ntls\_seg\_name\_strip);

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put(SQLCODE);

dbms\_output.put(SQLERRM);

--raise;

END;

/

--------------------------------------------------------

-- DDL for Procedure MOTO\_CDMA\_AUDIT

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."MOTO\_CDMA\_AUDIT"

IS

-- One Live and N non Live match

CURSOR t1\_nl\_match IS

SELECT NE\_OMCR,NE\_BTS\_NUMBER,NE\_CLUSTER\_NUMBER,NE\_SPAN\_NUMBER

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

WHERE MATCH\_CODE='BOTH' AND NG\_PATH\_STATUS='Live'

AND TERMINATION\_TYPE = 'T1';

-- One Live and N non Live match

CURSOR ebh\_nl\_match IS

SELECT NE\_OMCR,NE\_BTS\_NUMBER,NE\_CLUSTER\_NUMBER

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

WHERE MATCH\_CODE='BOTH' AND NG\_PATH\_STATUS='Live'

AND TERMINATION\_TYPE = 'Ethernet';

-- T1 spans matching multiple live paths in xng or matching multiple non live paths in xng in case a live match does not exist are marked NE Only

CURSOR t1\_cdma\_multiples\_xng IS

WITH

dup AS(

SELECT DISTINCT NE\_OMCR,NE\_BTS\_NUMBER,NE\_CLUSTER\_NUMBER,NE\_SPAN\_NUMBER,

count(CASE WHEN NG\_PATH\_STATUS = 'Live' THEN 1 ELSE NULL END) over (Partition by NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER, NE\_SPAN\_NUMBER) as live\_count,

COUNT(CASE WHEN NG\_PATH\_STATUS != 'Live' THEN 1 ELSE NULL END) OVER (PARTITION BY NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER, NE\_SPAN\_NUMBER) AS non\_live\_count

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT src

WHERE src.MATCH\_CODE='BOTH'

AND src.TERMINATION\_TYPE='T1'

AND src.NE\_BTS\_STATUS IN ('OOS','INS','INS\_ACTIVE'))

SELECT TGT.ROWID

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT tgt

JOIN dup ON dup.NE\_OMCR = tgt.NE\_OMCR AND dup.NE\_BTS\_NUMBER = tgt.NE\_BTS\_NUMBER AND dup.NE\_CLUSTER\_NUMBER = tgt.NE\_CLUSTER\_NUMBER AND dup.NE\_SPAN\_NUMBER = tgt.NE\_SPAN\_NUMBER

AND (dup.live\_count > 1 OR (dup.live\_count = 0 AND dup.non\_live\_count > 1));

-- Ethernet spans matching multiple live paths in xng or matching multiple non live paths in xng incase a live match does not exist are marked NE Only

CURSOR ethernet\_cdma\_multiples\_xng IS

WITH dup AS(

SELECT DISTINCT NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER,

count(CASE WHEN NG\_PATH\_STATUS = 'Live' THEN 1 ELSE null end) over (Partition by NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER) as live\_count,

COUNT(CASE WHEN NG\_PATH\_STATUS != 'Live' THEN 1 ELSE NULL END) OVER (PARTITION BY NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER) AS non\_live\_count

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT src

WHERE src.MATCH\_CODE='BOTH'

AND src.TERMINATION\_TYPE='Ethernet'

and src.NE\_BTS\_STATUS IN ('OOS','INS','INS\_ACTIVE'))

SELECT TGT.ROWID

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT tgt

JOIN dup ON dup.NE\_OMCR = tgt.NE\_OMCR AND dup.NE\_BTS\_NUMBER = tgt.NE\_BTS\_NUMBER AND dup.NE\_CLUSTER\_NUMBER = tgt.NE\_CLUSTER\_NUMBER

AND (dup.live\_count > 1 OR (dup.live\_count = 0 AND dup.non\_live\_count > 1));

--JC

-- Update those paths that dont have a CSR port

cursor update\_mcdma\_wrk\_no\_csr\_port is

WITH MXA\_PATHS as(

select mxa.NG\_PATH\_INST\_ID from MOTO\_CDMA\_NE\_VS\_NG\_AUDIT mxa

where mxa.TERMINATION\_TYPE = 'Ethernet' AND mxa.NG\_BANDWIDTH LIKE '%bps%' and mxa.MATCH\_CODE = 'BOTH' and mxa.NG\_PATH\_STATUS = 'Live'),

PATHS\_W\_CSR as( select cpi.NG\_PATH\_INST\_ID

from MXA\_PATHS cpi

join NG\_TOPOLOGY.TRAIL\_ELEMENT cpe on cpi.NG\_PATH\_INST\_ID = cpe.TRAIL\_ID

join NG\_REPORTS.PORT p on cpe.PORT\_AID = P.PORT\_REFERENCE\_ID

join NG\_REPORTS.EQUIPMENT ei on p.EQP\_REFERENCE\_ID = ei.EQP\_REFERENCE\_ID and ei.EQP\_TYPE = 'CSR')

select MXAP.NG\_PATH\_INST\_ID from MXA\_PATHS mxap

MINUS

select pc.NG\_PATH\_INST\_ID

FROM PATHS\_W\_CSR PC;

-- mis-matched BTS Type

CURSOR BTS\_TYPE\_MATCH IS

SELECT rowid rid

--NE\_OMCR,NE\_BTS\_NUMBER,NE\_CLUSTER\_NUMBER,NE\_SPAN\_NUMBER

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

WHERE MATCH\_CODE='BOTH' and SUBSTR(NE\_BTS\_TYPE, 1,2) = 'UB';

-- find the duplicated entries and remove the NE Only entry

cursor ne\_ethernet\_multiples is

select NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT A

WHERE A.MATCH\_CODE = 'BOTH' AND A.TERMINATION\_TYPE LIKE '%Eth%';

cursor ne\_t1\_multiples is

select NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER, NE\_SPAN\_NUMBER

FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT A

where a.MATCH\_CODE = 'BOTH' and a.TERMINATION\_TYPE = 'T1';

sql\_stmt varchar2(32000);

BEGIN

sql\_stmt := 'truncate TABLE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in moto\_cdma\_audit();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

Insert into MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

(NE\_OMCR,NE\_BTS\_NUMBER,NE\_CLUSTER\_NUMBER,ne\_BTS\_ROUTER\_GROUP,ne\_bandwidth,NE\_SPAN\_NUMBER, NE\_SPAN\_TYPE,ne\_ACCESS\_NODE\_NUMBER,NE\_AG\_NODE\_NUMBER,NE\_AG\_SPAN\_NUMBER,ne\_t1\_line\_number,

NE\_BTS\_TYPE, NE\_BTS\_STATUS,TERMINATION\_TYPE, switch\_id, MTX, NG\_PATH\_NAME, NG\_PATH\_INST\_ID,NG\_IPBSCDO\_NUMBER,NG\_BTS\_NUMBER,NG\_CLUSTER\_NUMBER,NG\_SPAN\_NUMBER,

NG\_BANDWIDTH,NG\_PATH\_STATUS, NG\_PATH\_TYPE, MATCH\_CODE, MATCH\_STATUS, EXTRACT\_DATE, AUDIT\_DATE)

WITH BTS\_STATUS as ( select distinct \* from mt\_omcr\_bts\_status stat where stat.BTS\_MLPPP\_NUMBER is null AND STAT.PBTS\_NUMBER IS NULL ),

moto\_ether\_terminations AS (SELECT inv.VSM\_DEVICE\_NAME\_OMCR,inv.BTS\_NUMBER,inv.CLUSTER\_NUMBER, inv.ROUTER\_GRP,inv.BANDWIDTH,inv.EXTRACT\_DATE,

SUBSTR (bmm.VSM\_DEVICE\_NAME\_MTX, 1, 6) || SUBSTR (bmm.VSM\_DEVICE\_NAME\_MTX, 11, 2) switch\_id, 'Ethernet' TERMINATION\_TYPE, bmm.VSM\_DEVICE\_NAME\_MTX, MP.BTS\_TYPE, stat.TEL\_STATE NE\_BTS\_STATUS

FROM moto\_cdma\_ethernet\_inv inv

JOIN NG\_REPORTS.bsm\_mtx\_map bmm ON inv.VSM\_DEVICE\_NAME\_OMCR = bmm.VSM\_DEVICE\_NAME\_BSM and upper(bmm.MTX\_STATUS )='LIVE' and upper(bmm.BSM\_STATUS)='LIVE'

JOIN MOTO\_BTS\_LOCATION\_MAP MP on inv.VSM\_DEVICE\_NAME\_OMCR = MP.VSM\_DEVICE\_NAME\_OMCR AND INV.BTS\_NUMBER = MP.BTS\_NUMBER

join BTS\_STATUS stat ON inv.VSM\_DEVICE\_NAME\_OMCR = stat.VSM\_DEVICE\_NAME\_OMCR and inv.BTS\_NUMBER = stat.BTS\_NUMBER),

moto\_t1\_terminations AS (SELECT mcs.BTS\_NUMBER, mcs.CLUSTER\_NUMBER, mcs.BTS\_ROUTER\_GROUP, mcs.SPAN\_NUMBER, mcs.ACCESS\_NODE\_NUMBER, mcs.AG\_NODE\_NUMBER, mcs.AG\_SPAN\_NUMBER, mcs.T1\_LINE\_NUMBER,

mcs.EXTRACT\_DATE, mcs.VSM\_DEVICE\_NAME\_OMCR, SUBSTR (bmm.VSM\_DEVICE\_NAME\_MTX, 1, 6) || SUBSTR (bmm.VSM\_DEVICE\_NAME\_MTX, 11, 2) switch\_id, 'T1' TERMINATION\_TYPE, bmm.VSM\_DEVICE\_NAME\_MTX, MP.BTS\_TYPE, stat.TEL\_STATE NE\_BTS\_STATUS

FROM moto\_cdma\_t1\_inv mcs

JOIN NG\_REPORTS.bsm\_mtx\_map bmm ON mcs.VSM\_DEVICE\_NAME\_OMCR = bmm.VSM\_DEVICE\_NAME\_BSM and upper(bmm.MTX\_STATUS )='LIVE' and upper(bmm.BSM\_STATUS)='LIVE'

JOIN MOTO\_BTS\_LOCATION\_MAP MP on mcs.VSM\_DEVICE\_NAME\_OMCR = MP.VSM\_DEVICE\_NAME\_OMCR AND MCS.BTS\_NUMBER = MP.BTS\_NUMBER

join BTS\_STATUS stat ON mcs.VSM\_DEVICE\_NAME\_OMCR = stat.VSM\_DEVICE\_NAME\_OMCR and mcs.BTS\_NUMBER = stat.BTS\_NUMBER),

moto\_ether\_audit AS (SELECT met.VSM\_DEVICE\_NAME\_OMCR NE\_OMCR, met.BTS\_NUMBER NE\_BTS\_NUMBER, met.CLUSTER\_NUMBER NE\_CLUSTER\_NUMBER, met.router\_grp ne\_BTS\_ROUTER\_GROUP, met.bandwidth ne\_bandwidth,

NULL NE\_SPAN\_NUMBER, 'VOICE' NE\_SPAN\_TYPE, NULL ne\_ACCESS\_NODE\_NUMBER, NULL NE\_AG\_NODE\_NUMBER, NULL NE\_AG\_SPAN\_NUMBER, NULL ne\_t1\_line\_number, met.BTS\_TYPE ne\_BTS\_TYPE, met.NE\_BTS\_STATUS,

met.TERMINATION\_TYPE, met.SWITCH\_ID, met.VSM\_DEVICE\_NAME\_MTX mtx, cpi.PATH\_NAME NG\_PATH\_NAME, cpi.TRAIL\_ID NG\_PATH\_INST\_ID, null NG\_IPBSCDO\_NUMBER, cpi.BTS\_NUMBER NG\_BTS\_NUMBER,

cpi.CLUSTER\_NUMBER NG\_CLUSTER\_NUMBER, cpi.SPAN\_NUMBER NG\_SPAN\_NUMBER,cpi.BANDWIDTH NG\_BANDWIDTH, NG\_PATH\_STATUS, NG\_PATH\_TYPE,

CASE WHEN cpi.SWITCH\_ID IS NULL THEN 'NE Only'

ELSE 'BOTH'

END MATCH\_CODE,

CASE WHEN cpi.SWITCH\_ID IS NULL THEN 'Not found in Granite'

ELSE GET\_UBS\_MATCH\_STATUS (CPI.SPAN\_NUMBER, cpi.BANDWIDTH, met.BTS\_TYPE, EBH\_TYPE, NON\_UBS\_TYPE, CPI.PARSE\_STATUS)

END match\_status, met.EXTRACT\_DATE,

cpi.ROWID cpi\_rowid

FROM moto\_ether\_terminations met

LEFT OUTER JOIN (SELECT \* FROM vzw\_moto\_voice\_cell\_paths cp

WHERE cp.BTS\_NUMBER IS NOT NULL AND cp.CLUSTER\_NUMBER IS NOT NULL)cpi ON met.SWITCH\_ID = cpi.SWITCH\_ID AND met.BTS\_NUMBER = cpi.BTS\_NUMBER AND met.CLUSTER\_NUMBER = cpi.CLUSTER\_NUMBER

),

moto\_t1\_audit AS (SELECT mtt.VSM\_DEVICE\_NAME\_OMCR NE\_OMCR, mtt.BTS\_NUMBER NE\_BTS\_NUMBER, mtt.CLUSTER\_NUMBER NE\_CLUSTER\_NUMBER,mtt.BTS\_ROUTER\_GROUP ne\_BTS\_ROUTER\_GROUP, NULL ne\_bandwidth,

mtt.SPAN\_NUMBER NE\_SPAN\_NUMBER, 'VOICE' NE\_SPAN\_TYPE, mtt.ACCESS\_NODE\_NUMBER ne\_ACCESS\_NODE\_NUMBER, mtt.AG\_NODE\_NUMBER NE\_AG\_NODE\_NUMBER, mtt.AG\_SPAN\_NUMBER NE\_AG\_SPAN\_NUMBER,

mtt.T1\_LINE\_NUMBER ne\_t1\_line\_number, mtt.BTS\_TYPE ne\_BTS\_TYPE, mtt.NE\_BTS\_STATUS, mtt.TERMINATION\_TYPE, mtt.SWITCH\_ID, mtt.VSM\_DEVICE\_NAME\_MTX mtx,

cpi.PATH\_NAME NG\_PATH\_NAME, cpi.TRAIL\_ID NG\_PATH\_INST\_ID, null NG\_IPBSCDO\_NUMBER, cpi.BTS\_NUMBER NG\_BTS\_NUMBER, cpi.CLUSTER\_NUMBER NG\_CLUSTER\_NUMBER,

cpi.SPAN\_NUMBER NG\_SPAN\_NUMBER,cpi.BANDWIDTH NG\_BANDWIDTH, NG\_PATH\_STATUS, NG\_PATH\_TYPE,

CASE WHEN cpi.SWITCH\_ID IS NULL THEN 'NE Only'

ELSE 'BOTH'

END MATCH\_CODE,

CASE WHEN cpi.SWITCH\_ID IS NULL THEN 'Not found in Granite'

ELSE

GET\_UBS\_MATCH\_STATUS (CPI.SPAN\_NUMBER, cpi.BANDWIDTH, mtt.BTS\_TYPE, T1\_TYPE, NON\_UBS\_TYPE, CPI.PARSE\_STATUS)

END match\_status, mtt.EXTRACT\_DATE,

cpi.ROWID cpi\_rowid

FROM moto\_t1\_terminations mtt

LEFT OUTER JOIN (SELECT \* FROM vzw\_moto\_voice\_cell\_paths cp

WHERE cp.BTS\_NUMBER IS NOT NULL AND cp.CLUSTER\_NUMBER IS NOT NULL and cp.SPAN\_NUMBER IS NOT NULL) cpi ON mtt.SWITCH\_ID = cpi.SWITCH\_ID

AND mtt.BTS\_NUMBER = cpi.BTS\_NUMBER AND mtt.CLUSTER\_NUMBER = cpi.CLUSTER\_NUMBER AND mtt.SPAN\_NUMBER = cpi.SPAN\_NUMBER ),

moto\_ne\_only\_and\_ng AS (SELECT \* FROM moto\_ether\_audit ea UNION SELECT \* FROM moto\_t1\_audit t1a ),

xng\_only\_path\_id AS (SELECT ROWID cpi\_rowid FROM vzw\_moto\_voice\_cell\_paths cpi MINUS SELECT cpi\_rowid FROM moto\_ne\_only\_and\_ng),

moto\_xng\_only AS (SELECT NULL NE\_OMCR, NULL NE\_BTS\_NUMBER, NULL NE\_CLUSTER\_NUMBER, NULL ne\_BTS\_ROUTER\_GROUP, NULL ne\_bandwidth,

NULL NE\_SPAN\_NUMBER, NULL NE\_SPAN\_TYPE, NULL ne\_ACCESS\_NODE\_NUMBER, NULL NE\_AG\_NODE\_NUMBER, NULL NE\_AG\_SPAN\_NUMBER, NULL ne\_t1\_line\_number, NULL ne\_bts\_type, null NE\_BTS\_STATUS,

CASE WHEN cpi.BANDWIDTH = 'DS1' THEN 'T1'

ELSE 'Ethernet'

END TERMINATION\_TYPE, cpi.SWITCH\_ID, bmm.VSM\_DEVICE\_NAME\_MTX mtx, cpi.PATH\_NAME NG\_PATH\_NAME, cpi.TRAIL\_ID NG\_PATH\_INST\_ID,

null NG\_IPBSCDO\_NUMBER, cpi.BTS\_NUMBER NG\_BTS\_NUMBER, cpi.CLUSTER\_NUMBER NG\_CLUSTER\_NUMBER, cpi.SPAN\_NUMBER NG\_SPAN\_NUMBER,cpi.BANDWIDTH NG\_BANDWIDTH, NG\_PATH\_STATUS, NG\_PATH\_TYPE,

'Xng Only' MATCH\_CODE, parse\_status match\_status, NULL EXTRACT\_DATE, cpi.ROWID cpi\_rowid

FROM vzw\_moto\_voice\_cell\_paths cpi, NG\_REPORTS.bsm\_mtx\_map bmm, xng\_only\_path\_id xopi

WHERE cpi.SWITCH\_ID = ( SUBSTR (bmm.VSM\_DEVICE\_NAME\_MTX, 1, 6) || SUBSTR (bmm.VSM\_DEVICE\_NAME\_MTX, 11, 2))

AND upper(bmm.MTX\_STATUS )='LIVE' AND cpi.ROWID = xopi.cpi\_rowid), full\_audit AS (SELECT \* FROM moto\_ne\_only\_and\_ng UNION SELECT \* FROM moto\_xng\_only)

SELECT NE\_OMCR, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER, ne\_BTS\_ROUTER\_GROUP, ne\_bandwidth, NE\_SPAN\_NUMBER, NE\_SPAN\_TYPE, ne\_ACCESS\_NODE\_NUMBER, NE\_AG\_NODE\_NUMBER,

NE\_AG\_SPAN\_NUMBER, ne\_t1\_line\_number, NE\_BTS\_TYPE, NE\_BTS\_STATUS, TERMINATION\_TYPE, switch\_id, mtx, NG\_PATH\_NAME, NG\_PATH\_INST\_ID, NG\_IPBSCDO\_NUMBER,

NG\_BTS\_NUMBER, NG\_CLUSTER\_NUMBER, NG\_SPAN\_NUMBER, NG\_BANDWIDTH, NG\_PATH\_STATUS, NG\_PATH\_TYPE, MATCH\_CODE, MATCH\_STATUS, EXTRACT\_DATE, TRUNC (SYSDATE) AS audit\_date

FROM full\_audit fa;

COMMIT;

dbms\_output.put\_line('Motorola CDMA audit table created');

-- perform UBS Type audit

CREATE\_UBS\_AUDIT();

-- remove the NE Only entry if BOTH exists in UBS path

-- due to multiple comparison for UBS

FOR CURSORREC IN NE\_ETHERNET\_MULTIPLES LOOP

DELETE FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT U

WHERE U.MATCH\_CODE LIKE 'NE%' AND u.TERMINATION\_TYPE LIKE 'Eth%' and u.NE\_OMCR = CURSORREC.NE\_OMCR and u.NE\_BTS\_NUMBER = CURSORREC.NE\_BTS\_NUMBER and u.NE\_CLUSTER\_NUMBER = CURSORREC.NE\_CLUSTER\_NUMBER;

END LOOP;

COMMIT;

FOR CURSORREC IN ne\_t1\_multiples LOOP

DELETE FROM MOTO\_CDMA\_NE\_VS\_NG\_AUDIT U

WHERE U.MATCH\_CODE LIKE 'NE%' AND u.TERMINATION\_TYPE = 'T1' and u.NE\_OMCR = CURSORREC.NE\_OMCR AND U.NE\_BTS\_NUMBER = CURSORREC.NE\_BTS\_NUMBER and u.NE\_SPAN\_NUMBER = CURSORREC.NE\_SPAN\_NUMBER AND U.NE\_CLUSTER\_NUMBER = CURSORREC.NE\_CLUSTER\_NUMBER;

END LOOP;

COMMIT;

-- check duplication

FOR CURSORREC IN T1\_CDMA\_MULTIPLES\_XNG LOOP

UPDATE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

SET MATCH\_STATUS = NVL2(MATCH\_STATUS,MATCH\_STATUS||',Multiple spans found in Granite for same tokens','Multiple spans found in Granite for same tokens') ,

MATCH\_CODE='NE Only'

WHERE rowid = CURSORREC.ROWID;

END LOOP;

commit;

FOR CURSORREC IN ETHERNET\_CDMA\_MULTIPLES\_XNG LOOP

UPDATE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

SET MATCH\_STATUS = NVL2(MATCH\_STATUS,MATCH\_STATUS||',Multiple spans found in Granite for same tokens','Multiple spans found in Granite for same tokens') ,

MATCH\_CODE='NE Only'

WHERE rowid = CURSORREC.ROWID;

END LOOP;

commit;

UPDATE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT

SET MATCH\_CODE = 'NE Only'

WHERE MATCH\_CODE = 'BOTH' AND match\_status IS NOT NULL;

COMMIT;

FOR CURSORREC IN t1\_nl\_match LOOP

UPDATE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT mnvxa

SET match\_status = nvl2(match\_status,'Live Path Exists, '||match\_status,'Live Path Exists')

WHERE mnvxa.NE\_OMCR=CURSORREC.NE\_OMCR AND mnvxa.NE\_BTS\_NUMBER=CURSORREC.NE\_BTS\_NUMBER AND mnvxa.NE\_CLUSTER\_NUMBER=CURSORREC.NE\_CLUSTER\_NUMBER AND mnvxa.NE\_SPAN\_NUMBER=CURSORREC.NE\_SPAN\_NUMBER

AND mnvxa.TERMINATION\_TYPE='T1' AND mnvxa.NG\_PATH\_STATUS <> 'Live';

END LOOP;

COMMIT;

FOR CURSORREC IN ebh\_nl\_match LOOP

UPDATE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT mnvxa

SET match\_status = nvl2(match\_status,'Live Path Exists, '||match\_status,'Live Path Exists')

WHERE mnvxa.NE\_OMCR=CURSORREC.NE\_OMCR AND mnvxa.NE\_BTS\_NUMBER=CURSORREC.NE\_BTS\_NUMBER AND mnvxa.NE\_CLUSTER\_NUMBER=CURSORREC.NE\_CLUSTER\_NUMBER

AND mnvxa.TERMINATION\_TYPE='Ethernet' AND mnvxa.NG\_PATH\_STATUS <> 'Live';

END LOOP;

COMMIT;

--JC

FOR CURSORREC IN update\_mcdma\_wrk\_no\_csr\_port LOOP

UPDATE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT mxa

SET MATCH\_CODE = 'NE Only', MATCH\_STATUS = CASE when MATCH\_STATUS = 'BOTH' then 'No CSR port on the e-pipe'

else

nvl2(match\_status, 'No CSR port on the e-pipe, '||match\_status, 'No CSR port on the e-pipe')

END

WHERE mxa.NG\_PATH\_INST\_ID = CURSORREC.NG\_PATH\_INST\_ID;

END LOOP;

commit;

--JC

dbms\_output.put\_line('Motorola audit completd successfully');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr('Error processing moto\_cdma\_audit(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

/

--------------------------------------------------------

-- DDL for Procedure MOTO\_CDMA\_DEVICE\_SUMMARY

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."MOTO\_CDMA\_DEVICE\_SUMMARY" IS

CURSOR UPDATECOMPLIANCE IS

select summ.rowid rd,summ.\* from MOTO\_CDMA\_AUDIT\_DEV\_SUMM summ;

sqlStmt VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

BEGIN

sqlStmt := 'truncate table ' || CDMA\_DEVICE\_SUMM;

execute immediate sqlStmt;

insert into MOTO\_CDMA\_AUDIT\_DEV\_SUMM (TERRITORY, SUB\_MARKET,LEAF\_DOMAIN,VSM\_DEVICE\_NAME\_MTX,SWITCH\_ID,T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,

EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE, EBH\_MATCHED\_LIVE\_CSR,EBH\_MATCHED\_NONLIVE\_CSR, UBS\_COUNT, NON\_UBS\_COUNT)

WITH mtx\_region\_map AS(SELECT DISTINCT cdmp.TERRITORY, cdmp.SUB\_MARKET, cdmp.LEAF\_DOMAIN\_NAME,m.VSM\_DEVICE\_NAME\_MTX mtx,

(SUBSTR (VSM\_DEVICE\_NAME\_MTX, 1, 6) || SUBSTR (VSM\_DEVICE\_NAME\_MTX, 11, 2) ) switch\_id

FROM clli\_domain\_map\_v cdmp

RIGHT OUTER JOIN (SELECT \* FROM NG\_REPORTS.bsm\_mtx\_map bmm WHERE bmm.VSM\_DEVICE\_NAME\_BSM LIKE '%OMCR' AND bmm.MTX\_STATUS = 'Live') m ON SUBSTR (m.VSM\_DEVICE\_NAME\_MTX, 1, 6) = SUBSTR (cdmp.CLLI, 1, 6)

WHERE cdmp.TERRITORY not in ('NNO', 'OSS')), AUDIT\_DETAILS AS( SELECT \* FROM NG\_REPORTS.MOTO\_CDMA\_NE\_VS\_NG\_AUDIT where NE\_BTS\_STATUS IN ('OOS','INS', 'INS\_ACTIVE')

and MATCH\_CODE IN ('BOTH', 'NE Only')),

T1\_DETAILS AS( select \* FROM audit\_details mnvxa where mnvxa.TERMINATION\_TYPE = 'T1' ),

t1\_discovered AS(SELECT MTX, COUNT (1) T1\_DISCOVERED FROM (SELECT DISTINCT MTX, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER, NE\_SPAN\_NUMBER from T1\_DETAILS ) GROUP BY mtx),

EBH\_DETAILS AS ( select \* FROM audit\_details mnvxa where mnvxa.TERMINATION\_TYPE = 'Ethernet' ),

ethernet\_discovered AS(SELECT MTX, COUNT (1) ETHERNET\_DISCOVERED FROM (SELECT DISTINCT MTX, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER from ebh\_details)GROUP BY mtx),

t1\_matched\_live AS (SELECT mtx, COUNT (1) t1\_matched\_live FROM (SELECT DISTINCT MTX, NE\_BTS\_NUMBER, NE\_CLUSTER\_NUMBER, NE\_SPAN\_NUMBER from T1\_DETAILS WHERE NG\_PATH\_STATUS = 'Live' AND MATCH\_CODE = 'BOTH') GROUP BY mtx),

t1\_matched\_nonlive AS (SELECT mtx, COUNT (1) t1\_matched\_nonlive FROM (SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER, MNVXA.NE\_SPAN\_NUMBER FROM t1\_details mnvxa WHERE mnvxa.NG\_PATH\_STATUS <> 'Live' AND mnvxa.MATCH\_CODE = 'BOTH'

MINUS

SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER,

MNVXA.NE\_CLUSTER\_NUMBER, MNVXA.NE\_SPAN\_NUMBER

FROM t1\_details mnvxa

WHERE mnvxa.NG\_PATH\_STATUS = 'Live'

AND mnvxa.MATCH\_CODE = 'BOTH'

)

GROUP BY mtx),

-- ethernet\_matched\_live AS

-- (SELECT mtx, COUNT (1) ethernet\_matched\_live

-- FROM (SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

-- from ebh\_details mnvxa

-- WHERE MNVXA.NG\_PATH\_STATUS = 'Live'

-- AND mnvxa.MATCH\_CODE = 'BOTH')

-- GROUP BY mtx),

ethernet\_matched\_live\_csr AS

(SELECT mtx, COUNT (1) ethernet\_matched\_live\_csr

FROM (SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

FROM ebh\_details mnvxa

join VZWNET.CIRC\_PATH\_INST cpi on mnvxa.NG\_PATH\_INST\_ID = cpi.CIRC\_PATH\_INST\_ID

join VZWNET.EPA p on cpi.CIRC\_PATH\_INST\_ID = P.CIRC\_PATH\_INST\_ID

join VZWNET.EQUIP\_INST ei on EI.EQUIP\_INST\_ID = P.EQUIP\_INST\_ID

WHERE mnvxa.NG\_PATH\_STATUS = 'Live' AND mnvxa.MATCH\_CODE = 'BOTH' and mnvxa.NG\_BANDWIDTH like '%bps' and EI.TYPE = 'CSR') GROUP BY mtx),

-- ethernet\_matched\_nonlive AS

-- (SELECT MTX, COUNT (1) ETHERNET\_MATCHED\_NONLIVE

-- FROM (SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

-- FROM ebh\_details mnvxa

-- WHERE mnvxa.NG\_PATH\_STATUS <> 'Live'

-- AND mnvxa.MATCH\_CODE = 'BOTH'

-- MINUS

-- SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

-- FROM ebh\_details mnvxa

-- WHERE mnvxa.NG\_PATH\_STATUS = 'Live'

-- AND mnvxa.MATCH\_CODE = 'BOTH')

-- GROUP BY mtx),

ethernet\_matched\_nonlive\_csr AS

(SELECT mtx, COUNT (1) ethernet\_matched\_nonlive\_csr

FROM (SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

FROM ebh\_details mnvxa

join VZWNET.CIRC\_PATH\_INST cpi on mnvxa.NG\_PATH\_INST\_ID = cpi.TRAIL\_ID

join VZWNET.EPA p on cpi.TRAIL\_ID = P.CIRC\_PATH\_INST\_ID

join VZWNET.EQUIP\_INST ei on EI.EQUIP\_INST\_ID = P.EQUIP\_INST\_ID

WHERE mnvxa.NG\_PATH\_STATUS <> 'Live' AND mnvxa.MATCH\_CODE = 'BOTH' and mnvxa.NG\_BANDWIDTH like '%bps' and EI.TYPE = 'CSR'

MINUS

SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

FROM ebh\_details mnvxa

join VZWNET.CIRC\_PATH\_INST cpi on mnvxa.NG\_PATH\_INST\_ID = cpi.TRAIL\_ID

join VZWNET.EPA p on cpi.TRAIL\_ID = P.CIRC\_PATH\_INST\_ID

join VZWNET.EQUIP\_INST ei on EI.EQUIP\_INST\_ID = P.EQUIP\_INST\_ID

WHERE mnvxa.NG\_PATH\_STATUS = 'Live' AND mnvxa.MATCH\_CODE = 'BOTH' and mnvxa.NG\_BANDWIDTH like '%bps' and EI.TYPE = 'CSR') GROUP BY MTX),

CONF\_BOTH\_COUNT AS -- for CONCONF BOTH

(SELECT MTX, COUNT (1) CONF\_BOTH\_COUNT

FROM (SELECT distinct MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_SPAN\_NUMBER

FROM t1\_details mnvxa

WHERE MNVXA.NE\_BTS\_TYPE LIKE 'UB%' AND MNVXA.NE\_CLUSTER\_NUMBER IS NULL and MATCH\_CODE='BOTH') GROUP BY MTX),

CONF\_NE\_COUNT AS -- for CONCONF NE only

(SELECT mtx, COUNT (1) CONF\_NE\_COUNT

FROM (SELECT distinct mnvxa.mtx, mnvxa.NE\_BTS\_NUMBER, mnvxa.NE\_SPAN\_NUMBER

FROM t1\_details mnvxa

WHERE mnvxa.ne\_bts\_type like 'UB%' AND MNVXA.NE\_CLUSTER\_NUMBER IS NULL AND MATCH\_CODE like 'NE%')GROUP BY MTX) ,

ubs\_count AS(SELECT MTX, COUNT (1) UBS\_COUNT

FROM (SELECT distinct MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER, MNVXA.NE\_SPAN\_NUMBER

FROM audit\_details mnvxa

WHERE mnvxa.ne\_bts\_type like 'UB%' ) GROUP BY mtx),

non\_ubs\_count AS (SELECT mtx, COUNT (1) non\_ubs\_count

FROM (SELECT distinct mnvxa.mtx, mnvxa.NE\_BTS\_NUMBER, mnvxa.NE\_CLUSTER\_NUMBER, mnvxa.NE\_SPAN\_NUMBER

FROM audit\_details mnvxa

WHERE mnvxa.ne\_bts\_type not like 'UB%') GROUP BY mtx)

SELECT mrm.TERRITORY, mrm.SUB\_MARKET, mrm.LEAF\_DOMAIN\_NAME, mrm.MTX, mrm.SWITCH\_ID, NVL (t1\_discovered, 0) AS t1\_discovered, NVL (t1\_matched\_live, 0) AS t1\_matched\_live,

NVL (t1\_matched\_nonlive, 0) AS t1\_matched\_nonlive, NVL (ETHERNET\_DISCOVERED, 0) AS Ebh\_DISCOVERED, NVL (CONF\_BOTH\_COUNT, 0) AS EBH\_MATCHED\_LIVE,

NVL (CONF\_NE\_COUNT, 0) AS ebh\_matched\_nonlive,NVL (ETHERNET\_MATCHED\_LIVE\_CSR, 0) AS Ebh\_MATCHED\_LIVE\_CSR, NVL (ETHERNET\_MATCHED\_NONLIVE\_CSR, 0) AS ebh\_MATCHED\_NONLIVE\_CSR,

NVL (ubs\_count, 0) AS ubs\_count, NVL (non\_ubs\_count, 0) AS non\_ubs\_count

FROM mtx\_region\_map mrm LEFT OUTER JOIN t1\_discovered td ON mrm.mtx = td.mtx

LEFT OUTER JOIN t1\_matched\_live tm ON mrm.mtx = tm.mtx

LEFT OUTER JOIN t1\_matched\_nonlive tum ON mrm.mtx = tum.mtx

LEFT OUTER JOIN ETHERNET\_DISCOVERED ED ON MRM.MTX = ED.MTX

LEFT OUTER JOIN CONF\_BOTH\_COUNT EM ON MRM.MTX = EM.MTX

LEFT OUTER JOIN CONF\_NE\_COUNT eum ON mrm.mtx = eum.mtx

LEFT OUTER JOIN ethernet\_matched\_live\_csr emc ON mrm.mtx = emc.mtx

LEFT OUTER JOIN ethernet\_matched\_nonlive\_csr eumc ON mrm.mtx = eumc.mtx

LEFT OUTER JOIN ubs\_count ubs ON mrm.mtx = ubs.mtx

LEFT OUTER JOIN NON\_UBS\_COUNT NUBS ON MRM.MTX = NUBS.MTX

ORDER BY mrm.mtx;

commit;

SQLSTMT := 'update '|| CDMA\_DEVICE\_SUMM ||' set t1\_comp = t, ebh\_comp = e, overall\_comp = o where rowid = a';

for cur in updateCompliance

loop

T1COMP := NVL(CUR.T1\_DISCOVERED,0) - NVL(CUR.T1\_MATCHED\_NONLIVE,0);

ebhComp := nvl(cur.ebh\_discovered,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

OVERALLCOMP := NVL(CUR.T1\_DISCOVERED,0) + NVL(CUR.EBH\_DISCOVERED,0) ;

overallComp := overallComp - nvl(cur.t1\_matched\_nonlive,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

if(t1Comp >0) then t1Comp := (nvl(cur.t1\_matched\_live,0)/t1Comp) \*100;

end if;

IF(EBHCOMP>0) THEN

ebhComp := (nvl(cur.ebh\_matched\_live\_csr,0)/ebhComp) \*100;

end if;

IF(OVERALLCOMP>0) THEN

overallComp := ((nvl(cur.t1\_matched\_live,0)+nvl(cur.ebh\_matched\_live\_csr,0))/overallComp) \*100;

--dbms\_output.put\_line('good EBH '||overallComp);

end if;

EXECUTE IMMEDIATE SQLSTMT USING T1COMP, EBHCOMP, OVERALLCOMP, CUR.RD;

end loop;

commit;

MOTO\_CDMA\_REGIONAL\_SUMMARY();

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in moto\_cdma\_device\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END;

/

--------------------------------------------------------

-- DDL for Procedure MOTO\_CDMA\_REGIONAL\_SUMMARY

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."MOTO\_CDMA\_REGIONAL\_SUMMARY" IS

CURSOR UPDATECOMPLIANCE IS

SELECT summ.rowid rd,summ.\* FROM MOTO\_CDMA\_AUDIT\_REG\_SUMM summ;

sqlStmt VARCHAR2(32767);

sqlStmtT1Comp VARCHAR2(32767);

sqlStmtEbhComp VARCHAR2(32767);

sqlStmtOvlComp VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

t1Reported NUMBER;

ebhReported NUMBER;

BEGIN

sqlStmt := 'truncate table ' || CDMA\_REGION\_SUMM;

execute IMMEDIATE sqlStmt;

sqlStmt := 'INSERT INTO '|| CDMA\_REGION\_SUMM || '

(TERRITORY, SUB\_MARKET, T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE, EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,ebh\_matched\_live\_csr,

ebh\_matched\_nonlive\_csr, t1\_reported, ebh\_reported, t1\_comp, ebh\_comp, overall\_comp, ubs\_count, non\_ubs\_count)

WITH mtx\_region\_map AS

(SELECT DISTINCT cdmp.TERRITORY, cdmp.SUB\_MARKET, m.VSM\_DEVICE\_NAME\_MTX mtx, ( SUBSTR (VSM\_DEVICE\_NAME\_MTX, 1, 6) || SUBSTR (VSM\_DEVICE\_NAME\_MTX, 11, 2)) switch\_id

FROM NG\_REPORTS.clli\_domain\_map\_v cdmp

LEFT OUTER JOIN

(SELECT \* FROM NG\_REPORTS.bsm\_mtx\_map bmm WHERE bmm.VSM\_DEVICE\_NAME\_BSM LIKE ''%OMCR%'' AND bmm.MTX\_STATUS = ''Live'') m ON SUBSTR (m.VSM\_DEVICE\_NAME\_MTX, 1, 6) = SUBSTR (cdmp.CLLI, 1, 6)

WHERE cdmp.TERRITORY not in (''NNO'', ''OSS''))

SELECT mrm.TERRITORY, mrm.SUB\_MARKET, SUM (NVL (cds.t1\_discovered, 0)) AS t1\_discovered, SUM (NVL (cds.t1\_matched\_live, 0)) AS t1\_matched\_live,

SUM (NVL (cds.t1\_matched\_nonlive, 0)) AS t1\_matched\_nonlive, SUM (NVL (cds.ebh\_discovered, 0)) AS ebh\_discovered, SUM (NVL (cds.ebh\_matched\_live, 0)) AS ebh\_matched\_live,

SUM (NVL (cds.ebh\_matched\_nonlive, 0)) AS ebh\_matched\_nonlive, SUM (NVL (cds.ebh\_matched\_live\_csr, 0)) AS ebh\_matched\_live\_csr, SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)) AS ebh\_matched\_nonlive\_csr,

SUM (NVL (cds.t1\_discovered, 0)) - SUM (NVL (cds.t1\_matched\_nonlive, 0)) t1\_reported, SUM (NVL (cds.ebh\_discovered, 0)) - SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)) ebh\_reported,

case when SUM (NVL (cds.t1\_discovered, 0)) - SUM (NVL (cds.t1\_matched\_nonlive, 0)) > 0

then SUM (NVL (cds.t1\_matched\_live, 0)) / (SUM (NVL (cds.t1\_discovered, 0)) - SUM (NVL (cds.t1\_matched\_nonlive, 0)))\*100

else 0

end t1\_comp,

case when SUM (NVL (cds.ebh\_discovered, 0)) - SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)) > 0

then SUM (NVL (cds.ebh\_matched\_live\_csr, 0)) / (SUM (NVL (cds.ebh\_discovered, 0)) - SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)))\*100

else 0

end ebh\_comp,

case when SUM(nvl(cds.t1\_discovered,0) + nvl(cds.ebh\_discovered,0))- SUM(NVL (cds.ebh\_matched\_nonlive\_csr, 0)+NVL (cds.t1\_matched\_nonlive, 0)) > 0

then ((sum (nvl(cds.t1\_matched\_live,0)+nvl(cds.ebh\_matched\_live\_csr,0)))/(SUM(nvl(cds.t1\_discovered,0) + nvl(cds.ebh\_discovered,0))- SUM(NVL (cds.ebh\_matched\_nonlive\_csr, 0)+NVL (cds.t1\_matched\_nonlive, 0))))\*100

else 0

end overall\_comp,

SUM (NVL (cds.ubs\_count, 0)) AS ubs\_count,

SUM (NVL (cds.non\_ubs\_count, 0)) AS non\_ubs\_count

FROM mtx\_region\_map mrm

LEFT OUTER JOIN '|| CDMA\_DEVICE\_SUMM ||' cds ON mrm.mtx = cds.VSM\_DEVICE\_NAME\_MTX

GROUP BY ROLLUP (mrm.TERRITORY, mrm.SUB\_MARKET)

ORDER BY mrm.TERRITORY, mrm.SUB\_MARKET';

execute IMMEDIATE sqlStmt;

COMMIT;

sqlStmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is null and sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.SUB\_MARKET=''Unknown'' where territory=''Unknown''' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in populate\_cdma\_regional\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END;

/

--------------------------------------------------------

-- DDL for Procedure NTLS\_CSR\_VLAN\_PATHS\_TABLE

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS\_TABLE" AS

BEGIN

insert into /\*parallel (s,6) \*/ NTLS\_CSR\_VLAN\_PATHS columns ( EQP\_REFERENCE\_ID, CARD\_REFERENCE\_ID, CKT\_PATH\_REFERENCE\_ID, port\_BANDWIDTH, path\_TYPE, vlan\_inst\_id, vlan\_hum\_id, vlan\_number, vlan\_status, extract\_date)

with

csr\_gige\_paths as (select distinct XNE.EQP\_REFERENCE\_ID, p.CARD\_REFERENCE\_ID --, p.port\_inst\_id, pli.COMPONENT\_ID,(pli.rel\_order), PLI.LEG\_NAME

,cpi.TRAIL\_ID gige\_inst\_id, cpi.BANDWIDTH, cpi.type

from NG\_REPORTS.NTLS\_CSR\_PARSED\_WK xne,ng\_reports.PORT p,ng\_topology.TRAIL cpi,ng\_topology.TRAIL\_COMPONENT pli,ng\_topology.TRAIL\_COMPONENT\_ELEMENT plm,ng\_topology.TRAIL\_ELEMENT cpe

where p.EQP\_REFERENCE\_ID = xne.EQP\_REFERENCE\_ID and cpi.TRAIL\_ID in (p.CKT\_PATH\_REFERENCE\_ID, p.NEXT\_PATH\_REFERENCE\_ID) and cpi.BANDWIDTH like '%Gbps' and pli.TRAIL\_ID = cpi.TRAIL\_ID and pli.COMPONENT\_ID = plm.COMPONENT\_ID

-- and plm.sequence = cpe.sequence

and plm.ELEMENT\_ID = cpe.ELEMENT\_ID and CPE.ELEMENT\_REF\_ID = p.PORT\_REFERENCE\_ID and CPE.ELEMENT\_TYPE = 'E' and pli.SEQUENCE = (select min(SEQUENCE) SEQUENCE from ng\_topology.TRAIL\_COMPONENT li where li.TRAIL\_ID = cpi.TRAIL\_ID)

),

gige\_vlan\_paths as (

select distinct gige.TRAIL\_ID gige\_inst\_id , VLAN.TRAIL\_ID vlan\_inst\_id

,vlan.TRAIL\_NAME vlan\_hum\_id, regexp\_substr(regexp\_substr(vlan.TRAIL\_NAME, 'VLAN(-|[[:space:]]|)\*[[:digit:]]+'), '[[:digit:]]+') vlan\_number, vlan.STATUS

from ng\_topology.TRAIL gige, ng\_topology.TRAIL vlan, ng\_topology.TRAIL\_COMPONENT pli, ng\_topology.TRAIL\_COMPONENT\_ELEMENT plm, ng\_topology.TRAIL\_ELEMENT cpe

where gige.BANDWIDTH like '%Gbps' and vlan.BANDWIDTH = 'VLAN' and vlan.TYPE = 'EBH' and VLAN.TRAIL\_ID = CPE.TRAIL\_ID and gige.TRAIL\_ID = CPE.ELEMENT\_REF\_ID and CPE.ELEMENT\_TYPE = 'P'

and VLAN.TRAIL\_ID = pli.TRAIL\_ID and pli.COMPONENT\_ID = plm.COMPONENT\_ID

-- and plm.sequence = cpe.sequence

and plm.ELEMENT\_ID = cpe.ELEMENT\_ID and pli.SEQUENCE=(select min(SEQUENCE) from ng\_topology.TRAIL\_COMPONENT pl where pl.TRAIL\_ID = VLAN.TRAIL\_ID)

)select ngp.EQP\_REFERENCE\_ID, ngp.CARD\_REFERENCE\_ID ,ngp.gige\_inst\_id CKT\_PATH\_REFERENCE\_ID, ngp.BANDWIDTH, ngp.type , gvp.vlan\_inst\_id, gvp.vlan\_hum\_id, gvp.vlan\_number, gvp.STATUS vlan\_status, sysdate

from csr\_gige\_paths ngp, gige\_vlan\_paths gvp where ngp.gige\_inst\_id = gvp.gige\_inst\_id;

NULL;

END NTLS\_CSR\_VLAN\_PATHS\_table;

/

--------------------------------------------------------

-- DDL for Procedure RESTORE\_MOTO\_CDMA\_WRK

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."RESTORE\_MOTO\_CDMA\_WRK" IS

sqlStmt varchar2(32000);

BEGIN

sqlStmt := 'delete from MOTO\_CDMA\_NE\_VS\_NG\_AUDIT';

execute immediate sqlStmt;

sqlStmt := 'insert into MOTO\_CDMA\_NE\_VS\_NG\_AUDIT select \* from moto\_cdma\_ne\_vs\_ng\_audit';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done populating MOTO\_CDMA\_NE\_VS\_NG\_AUDIT');

sqlStmt := 'delete from MOTO\_CDMA\_AUDIT\_REG\_SUMM';

execute immediate sqlStmt;

sqlStmt := 'insert into MOTO\_CDMA\_AUDIT\_REG\_SUMM (T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,MARKET\_TERRITORY,SUB\_MARKET, TERRITORY\_MARKET\_SUB) select T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,MARKET\_TERRITORY,SUB\_MARKET, TERRITORY\_MARKET\_SUB from moto\_cdma\_audit\_reg\_summ';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done populating MOTO\_CDMA\_AUDIT\_REG\_SUMM');

sqlStmt := 'delete from MOTO\_CDMA\_AUDIT\_DEV\_SUMM';

execute immediate sqlStmt;

sqlStmt := 'insert into MOTO\_CDMA\_AUDIT\_DEV\_SUMM select \* from moto\_cdma\_audit\_dev\_summ';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done populating MOTO\_CDMA\_AUDIT\_DEV\_SUMM');

EXCEPTION

WHEN OTHERS THEN

WATCHDOG.logerror('MOTO\_CDMA\_AUDIT',4000,SubStr('Error in restore\_moto\_cdma\_wrk; ' ||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255),'Y');

WATCHDOG.updateprocessend('MOTO\_CDMA\_AUDIT','STATUS\_FAILURE','N');

END;

/

--------------------------------------------------------

-- DDL for Procedure RUN\_DAILY\_DETAIL\_RPT

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."RUN\_DAILY\_DETAIL\_RPT"

IS

v\_id NUMBER;

v\_cutoff\_time\_last DATE;

v\_cutoff\_time\_new DATE;

v\_group\_name VARCHAR2 (30);

BEGIN

SELECT NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN\_SEQ.NEXTVAL

INTO v\_id

FROM DUAL;

BEGIN

SELECT MAX (PROC\_START\_TS)

INTO v\_cutoff\_time\_last

FROM CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN;

EXCEPTION

WHEN NO\_DATA\_FOUND

THEN

SELECT TO\_DATE ('01-01-1970', 'MM-DD-YYYY')

INTO v\_cutoff\_time\_last

FROM DUAL;

END;

SELECT SYSDATE INTO v\_cutoff\_time\_new FROM DUAL;

INSERT INTO CIRC\_SEG\_NEW\_DECOM\_DETAIL (RPT\_ID,

DOMAIN\_ID, --domain\_id / NG\_TOPOLOGY.SEGMENT\_DOMAIN\_MAP

DOMAIN\_NAME, --domain\_name / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

TERRITORY, --territory / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

MARKET\_TERRITORY, --market\_territory / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

SUB\_MARKET, --sub\_market / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

SEGMENT\_ID, --segment\_id / NG\_TOPOLOGY.TRAIL\_SEGMENT

SEGMENT\_NAME, --circ\_hum\_id / XNG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DET\_TESS\_VW = segment\_id / NG\_TOPOLOGY.TRAIL\_SEGMENT

SEGMENT\_TYPE, --type / NG\_TOPOLOGY.TRAIL\_SEGMENT

Z\_SITE\_ID, --z\_site\_id / NG\_TOPOLOGY.TRAIL\_SEGMENT

SITE\_NAME, --site\_hum\_id = site\_name / NG\_REPORTS.SITE - ng\_inventory.SITE

VENDOR, --vendor / NG\_TOPOLOGY.TRAIL\_SEGMENT

BILLING\_CODE, --billing\_code / NG\_TOPOLOGY.TRAIL\_SEGMENT

STATUS, --status / NG\_TOPOLOGY.TRAIL\_SEGMENT

ACTION\_TYPE, --part of the insert fn

COMMIT\_DATE, --commit\_date / NG\_TOPOLOGY.JV\_COMMIT

AUTHOR, --author / NG\_TOPOLOGY.JV\_COMMIT

BANDWIDTH, --bandwidth / NG\_TOPOLOGY.TRAIL\_SEGMENT

MRC --recur\_costs / VZWNET.CIRC\_INST = recur\_costs / NG\_TOPOLOGY.TRAIL\_SEGMENT

)

WITH INST\_AUDIT\_UPDATED

AS (SELECT g.local\_id,

cc.qualified\_name,

c.commit\_date,

c.author,

s.type,

s.version

FROM ng\_topology.jv\_global\_id g,

ng\_topology.jv\_cdo\_class cc,

ng\_topology.jv\_commit c,

ng\_topology.jv\_snapshot s

WHERE cc.cdo\_class\_pk = g.cdo\_class\_fk

AND g.global\_id\_pk = s.global\_id\_fk

AND s.commit\_fk = c.commit\_pk

AND c.commit\_date >= '05-JAN-18' --v\_cutoff\_time\_last

AND c.commit\_date < '06-JAN-18' --v\_cutoff\_time\_new

AND cc.qualified\_name = 'Trail'

AND s.type = 'UPDATE'),

INST\_UPDATE\_LOG\_DECOM

AS (SELECT g.local\_id,

cc.qualified\_name,

s.version,

s.state

FROM ng\_topology.jv\_global\_id g,

ng\_topology.jv\_cdo\_class cc,

ng\_topology.jv\_snapshot s

WHERE cc.cdo\_class\_pk = g.cdo\_class\_fk

AND g.global\_id\_pk = s.global\_id\_fk

AND REGEXP\_LIKE (state, '"FIELD\_CHANGED": "Status".\*"NEW\_VALUE": "Decommissioned"', 'i')),

AUDIT\_DECOM\_LOG

AS (SELECT a.local\_id, a.commit\_date, a.author

FROM INST\_AUDIT\_UPDATED a, INST\_UPDATE\_LOG\_DECOM u

WHERE u.local\_id = a.local\_id

AND u.qualified\_name = a.qualified\_name

AND u.version = a.version),

AUDIT\_DECOM

AS (SELECT a.local\_id, a.commit\_date, a.author

FROM AUDIT\_DECOM\_LOG a

WHERE a.commit\_date = (SELECT MAX (a.commit\_date)

FROM AUDIT\_DECOM\_LOG b

WHERE a.local\_id = b.local\_id)),

AUDIT\_INSERT

AS (SELECT g.local\_id,

c.commit\_date,

c.author

FROM ng\_topology.jv\_global\_id g,

ng\_topology.jv\_commit c,

ng\_topology.jv\_cdo\_class cc,

ng\_topology.jv\_snapshot s

WHERE c.commit\_date >= '05-JAN-18' --v\_cutoff\_time\_last

AND c.commit\_date < '06-JAN-18' --v\_cutoff\_time\_new

AND cc.qualified\_name = 'Trail'

AND s.type = 'INITIAL'),

AUDIT\_INSERT\_MAX

AS (SELECT a.local\_id, a.commit\_date, a.author

FROM AUDIT\_INSERT a

WHERE a.commit\_date = (SELECT MAX (a.commit\_date)

FROM AUDIT\_INSERT b

WHERE a.local\_id = b.local\_id)),

AUDIT\_INSERT\_DECOM

AS (SELECT local\_id,

commit\_date,

author,

'NEW' AS action

FROM AUDIT\_INSERT\_MAX

UNION

SELECT local\_id,

commit\_date,

author,

'DECOM' AS action

FROM AUDIT\_DECOM),

CIRC\_INST\_ALL

AS (SELECT segment\_id,

type AS segment\_type,

name,

z\_site\_id,

vendor,

billing\_code,

status,

bandwidth,

recur\_costs

FROM ng\_topology.AR\_TRAIL\_SEGMENT

UNION

SELECT segment\_id,

type AS segment\_type,

name,

z\_site\_id,

vendor,

billing\_code,

status,

bandwidth,

recur\_costs

FROM ng\_topology.TRAIL\_SEGMENT)

SELECT v\_id,

rpt.domain\_id,

RPT.DOMAIN\_NAME,

RPT.TERRITORY,

RPT.MARKET\_TERRITORY,

RPT.SUB\_MARKET,

circ.segment\_id,

circ.name,

circ.segment\_type,

circ.z\_site\_id,

z.site\_name,

circ.vendor,

circ.billing\_code,

circ.status,

aud.action,

aud.commit\_date,

aud.author,

circ.bandwidth,

circ.recur\_costs

FROM AUDIT\_INSERT\_DECOM aud,

CIRC\_INST\_ALL circ,

NG\_TOPOLOGY.SEGMENT\_DOMAIN\_MAP dm,

NG\_REPORTS.DOMAINS\_LEAF\_REPORTING rpt,

NG\_REPORTS.SITE z -- site is for detail only. ng\_inventory.SITE

WHERE aud.local\_id = circ.segment\_id

AND circ.segment\_id = dm.segment\_id

AND dm.domain\_id = rpt.domain\_id

AND circ.z\_site\_id = z.site\_reference\_id;

--

-- -- Update 4 fields related to Disconnect

-- v\_group\_name := 'Disconnect Information';

--

-- DELETE FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL a

-- WHERE a.commit\_date <>

-- (SELECT MAX (commit\_date)

-- FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL b

-- WHERE a.domain\_id = b.domain\_id

-- AND a.segment\_id = b.segment\_id

-- AND a.action\_type = b.action\_type);

--

-- UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

-- SET DISCONNECT\_PON =

-- (SELECT ATTRIBUTE\_VALUE

-- FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa

-- WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID

-- AND sa.GROUP\_NAME = v\_group\_name

-- AND sa.ATTRIBUTE\_NAME = 'Disconnect PON')

-- WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

--

-- UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

-- SET DISCONNECT\_ORDER\_TS =

-- (SELECT ATTRIBUTE\_VALUE

-- FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa

-- WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID

-- AND sa.GROUP\_NAME = v\_group\_name

-- AND sa.ATTRIBUTE\_NAME = 'Disconnect Order Date')

-- WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

--

-- UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

-- SET DISCONNECT\_ORDER\_NUM =

-- (SELECT ATTRIBUTE\_VALUE

-- FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa

-- WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID

-- AND sa.GROUP\_NAME = v\_group\_name

-- AND sa.ATTRIBUTE\_NAME = 'Disconnect Order Number')

-- WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

--

-- UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

-- SET DISCONNECT\_TS =

-- (SELECT ATTRIBUTE\_VALUE

-- FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa

-- WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID

-- AND sa.GROUP\_NAME = v\_group\_name

-- AND sa.ATTRIBUTE\_NAME = 'Disconnect Date')

-- WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

--

-- UPDATE CIRC\_SEG\_NEW\_DECOM\_DETAIL D

-- SET D.SEGMENT\_NAME =

-- (SELECT TS.NAME

-- FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS

-- WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID)

-- WHERE (D.SEGMENT\_NAME IS NULL)

-- OR D.SEGMENT\_NAME != (SELECT TS.NAME

-- FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS

-- WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID);

--

-- UPDATE CIRC\_SEG\_NEW\_DECOM\_DETAIL D

-- SET D.MRC =

-- (SELECT TS.RECUR\_COSTS

-- FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS

-- WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID)

-- WHERE (D.MRC IS NULL)

-- OR D.MRC != (SELECT TS.RECUR\_COSTS

-- FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS

-- WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID);

--

-- RUN\_ALL\_MONTHLY\_WEEKLY\_RPT (); --as we deleted some duplicated, so need recalculate all count in summary

--

-- --RUN\_ONE\_MONTHLY\_RPT (ADD\_MONTHS (v\_cutoff\_time\_new, 1)); --current month=next mon's previous mon

--

-- --RUN\_ONE\_WEEKLY\_RPT (v\_cutoff\_time\_new + 7); --current week=next week's previous week

INSERT

INTO CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN (RPT\_ID, PROC\_START\_TS, PROC\_END\_TS)

VALUES (v\_id, v\_cutoff\_time\_new, SYSDATE);

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

ROLLBACK;

END;

/

--------------------------------------------------------

-- DDL for Procedure TRAIL\_SEGMENT

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."TRAIL\_SEGMENT" AS

DSC VARCHAR (512);

BEGIN

SELECT

segment\_id,

name,

status,

type,

vendor,

bandwidth,

order\_number,

ordered\_date,

due,

installed,

in\_service,

due,

decommision,

parent,

CASE

WHEN a\_termination\_type = 'PORT' THEN a\_termination\_inst\_id

else null

END

a\_port,

CASE

WHEN z\_termination\_type = 'PORT' THEN z\_termination\_inst\_id

else null

END

p\_port,

CASE

WHEN a\_termination\_type = 'SEGMENT' THEN a\_termination\_inst\_id

else null

END

a\_linked\_seg\_id,

CASE

WHEN z\_termination\_type = 'SEGMENT' THEN z\_termination\_inst\_id

else null

END

z\_linked\_seg\_id,

CASE

WHEN a\_termination\_type = 'EQUIPMENT' THEN a\_termination\_inst\_id

else null

END

a\_equip\_reference\_id,

CASE

WHEN z\_termination\_type = 'EQUIPMENT' THEN z\_termination\_inst\_id

else null

END

z\_equip\_reference\_id,

A\_TERMINATION\_TYPE,Z\_TERMINATION\_TYPE,

a\_site\_id,

z\_site\_id,

renewal\_date,

billing\_code,

current\_trail\_id,

path\_chg\_date,

next\_trail\_id,

last\_mod\_ts,

last\_mod\_by,

last\_renewal\_date,

renewal\_term,

recur\_costs,

non\_recur\_costs

INTO DSC

FROM

ng\_topology.trail\_segment;

END TRAIL\_SEGMENT;

/

--------------------------------------------------------

-- DDL for Procedure TRUNCATE\_MOTO\_CDMA\_WRK

--------------------------------------------------------

set define off;

CREATE OR REPLACE EDITIONABLE PROCEDURE "NG\_REPORTS"."TRUNCATE\_MOTO\_CDMA\_WRK" IS

sqlStmt varchar2(32000);

BEGIN

sqlStmt := 'delete from MOTO\_CDMA\_NE\_VS\_NG\_AUDIT';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting MOTO\_CDMA\_NE\_VS\_NG\_AUDIT');

sqlStmt := 'delete from MOTO\_CDMA\_AUDIT\_REG\_SUMM';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting MOTO\_CDMA\_AUDIT\_REG\_SUMM');

sqlStmt := 'delete from MOTO\_CDMA\_AUDIT\_DEV\_SUMM';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting MOTO\_CDMA\_AUDIT\_DEV\_SUMM');

END;

/

--------------------------------------------------------

-- DDL for Package AUDIT\_AL\_CSR

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."AUDIT\_AL\_CSR" AS

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

PROCEDURE audit\_al\_csr\_vlans

IS

methodname VARCHAR2 (30) = 'audit\_al\_csr\_vlans';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

BEGIN

EXECUTE IMMEDIATE (delete SAM\_CSR\_VLAN\_AUDIT\_WK where CSR\_VENDOR = 'AL');

COMMIT;

sqlstmt :=

'insert into SAM\_CSR\_VLAN\_AUDIT\_WK columns (CSR\_DEVICE\_NAME, CSR\_VENDOR, ne\_hostName, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ntls\_vlan\_number -- ne\_hostName???

, vlan\_inst\_id, vlan\_status, PORT\_NAME)

WITH

al\_vlans AS(SELECT xnvp.\*, xne.CSR\_DEVICE\_NAME

FROM NTLS\_CSR\_vlan\_paths xnvp , NG\_Reports.NTLS\_CSR\_PARSED\_WK xne

where xne.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID and xne.CSR\_DEVICE\_NAME like ''%-AL-%'' AND xnvp.vlan\_number IS NOT NULL)

,matches AS(

SELECT aud.CSR\_DEVICE\_NAME, sam.OUT\_ENCAP\_VALUE ne\_vlan\_number,sam.DISPLAYED\_NAME, sam.NODE\_ID, ''BOTH'' match\_code,NULL match\_status

,xnvp.vlan\_number ntls\_vlan\_number, xnvp.vlan\_inst\_id,

xnvp.vlan\_status, aud.CSR\_VENDOR, SUBSTR(sam.PORT\_NAME, 0, INSTR(sam.PORT\_NAME, '':'')-1) AS PORT\_NAME

FROM NG\_REPORTS.SAM\_EBH\_RTR sam

JOIN SAM\_NT\_AL\_CSR\_AUDIT\_WK aud ON upper(aud.CSR\_DEVICE\_NAME) = upper(sam.NODE\_NAME) -- Audit table

JOIN al\_vlans xnvp ON aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID AND sam.OUT\_ENCAP\_VALUE = xnvp.vlan\_number

WHERE sam.NODE\_ID IS NOT NULL AND sam.NODE\_NAME not like ''%7750-\_\_''),

ne\_only AS(SELECT aud.CSR\_DEVICE\_NAME, sam.DISPLAYED\_NAME, sam.NODE\_ID,sam.OUT\_ENCAP\_VALUE ne\_vlan\_number, ''NE Only'' match\_code,''Missing in Nautilus'' match\_status, null,

null ntls\_vlan\_number, NULL vlan\_inst\_id, NULL vlan\_status,aud.CSR\_VENDOR, SUBSTR(sam.PORT\_NAME, 0, INSTR(sam.PORT\_NAME, '':'')-1) AS PORT\_NAME

FROM NG\_REPORTS.SAM\_EBH\_RTR sam

JOIN SAM\_NT\_AL\_CSR\_AUDIT\_WK aud ON upper(aud.CSR\_DEVICE\_NAME) = upper(sam.NODE\_NAME)

WHERE sam.OUT\_ENCAP\_VALUE IS NOT NULL AND sam.NODE\_ID IS NOT NULL AND sam.NODE\_NAME not like ''%7750-\_\_''

AND NOT EXISTS (SELECT 1

FROM al\_vlans xnvp

WHERE aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID AND upper(xnvp.CSR\_DEVICE\_NAME) = upper(aud.CSR\_DEVICE\_NAME) AND sam.OUT\_ENCAP\_VALUE = xnvp.vlan\_number)), xng\_only AS

(SELECT aud.CSR\_DEVICE\_NAME, NULL DISPLAYED\_NAME, NULL NODE\_ID, null ne\_vlan\_number, ''Nautilus Only'' match\_code, ''Missing in NE'' match\_status, xnvp.vlan\_number ntls\_vlan\_number, xnvp.vlan\_inst\_id,

xnvp.vlan\_status, aud.CSR\_VENDOR, NULL PORT\_NAME

FROM al\_vlans xnvp

JOIN SAM\_NT\_AL\_CSR\_AUDIT\_WK aud ON aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID AND upper(AUD.CSR\_DEVICE\_NAME) = upper(xnvp.CSR\_DEVICE\_NAME) and not exists(select 1 from sam\_ebh\_rtr sam

where upper(aud.CSR\_DEVICE\_NAME) = upper(sam.NODE\_NAME) AND sam.OUT\_ENCAP\_VALUE = xnvp.vlan\_number and sam.OUT\_ENCAP\_VALUE IS NOT NULL AND sam.NODE\_NAME not like ''%7750-\_\_''))

SELECT CSR\_DEVICE\_NAME, CSR\_VENDOR, CSR\_DEVICE\_NAME, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ntls\_vlan\_number, vlan\_inst\_id, vlan\_status, port\_name

FROM matches

UNION

SELECT CSR\_DEVICE\_NAME, CSR\_VENDOR, CSR\_DEVICE\_NAME, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ntls\_vlan\_number, vlan\_inst\_id, vlan\_status, port\_name

FROM ne\_only

UNION

SELECT CSR\_DEVICE\_NAME, CSR\_VENDOR, NULL, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ntls\_vlan\_number, vlan\_inst\_id, vlan\_status, port\_name

FROM xng\_only'

;

EXECUTE IMMEDIATE (sqlstmt);

COMMIT;

insert\_al\_CSR\_vlan\_issues ();

EXCEPTION

WHEN OTHERS

THEN

MESSAGE = 'Error: ' || methodname || '(): Can't audit AL VLANs';

RAISE;

END;

END audit\_al\_csr\_vlans;

END AUDIT\_AL\_CSR;

/

--------------------------------------------------------

-- DDL for Package BTOD\_AUDIT\_PKG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."BTOD\_AUDIT\_PKG" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: TEST

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 10/23/2017 Catherine Sayre Created

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Procedure runAudit;

Procedure load\_btod\_ntls\_audit\_detail;

Procedure load\_btod\_ntls\_audit\_summary;

END BTOD\_AUDIT\_PKG;

/

--------------------------------------------------------

-- DDL for Package CIENA\_6500\_DISCOVERY

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."CIENA\_6500\_DISCOVERY" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: CIENA\_6500\_DISCOVERY

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 10/3/2017 ramamla 1. Created this package.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CUT\_OFF NUMBER := 7;

PROCEDURE CLEANUP\_PARSED\_DATA;

PROCEDURE DELETE\_WK\_TABLES;

PROCEDURE DELETE\_STALE\_DATA\_PRD\_TABLES;

PROCEDURE COPY\_WK\_TO\_PRD;

PROCEDURE DELETE\_DEVICES\_NOT\_IN\_TIDLIST;

END CIENA\_6500\_DISCOVERY;

/

--------------------------------------------------------

-- DDL for Package CIRC\_SEG\_NEW\_DECOM\_RPT\_PKG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_RPT\_PKG" AS

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

PROCEDURE RUN\_DAILY\_DETAIL\_RPT;

PROCEDURE RUN\_ONE\_MONTHLY\_RPT (p\_run\_date DATE);

PROCEDURE RUN\_ONE\_WEEKLY\_RPT (p\_run\_date DATE);

PROCEDURE RUN\_ALL\_MONTHLY\_WEEKLY\_RPT;

PROCEDURE RUN\_ONE\_YEARLY\_RPT (p\_run\_date DATE);

END CIRC\_SEG\_NEW\_DECOM\_RPT\_PKG;

/

--------------------------------------------------------

-- DDL for Package ENB\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."ENB\_AUDIT" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: eNBAudit

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 10/25/2010 1. Created this package.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*FUNCTION MyFunction(Param1 IN NUMBER) RETURN NUMBER;\*/

AUDIT\_TABLE\_NAME varchar2(30) := 'NE\_VS\_XNG\_ENB\_AUDIT\_WRK';

SAM\_ENB varchar2(30) := 'SAM\_ENB\_WK';

ERICSSON\_ENB\_WK varchar2(30) := 'ERICSSON\_ENB\_WK';

NTLS\_ENB\_PARSED\_V varchar2(30) := 'NTLS\_PARSED\_ENB\_V';

PROCEDURE eNbNeVsXngAudit;

/\* generate reports and save them for rendering adhoc reports\*/

procedure generateRegionalSummary;

procedure generateMarketIdSummary;

-- procedure copy\_enb\_wrk;

procedure truncate\_enb\_wrk;

-- procedure restore\_enb\_wrk;

END eNB\_Audit;

/

--------------------------------------------------------

-- DDL for Package LOAD\_NTLS\_CSR\_GIGE\_TABLES

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."LOAD\_NTLS\_CSR\_GIGE\_TABLES"

IS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: LOAD\_GIGE\_TABLES

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 11/13/2017 1.Created this package.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE runLoad;

PROCEDURE get\_csr\_gige\_trails;

PROCEDURE get\_csr\_gige\_parents;

PROCEDURE get\_gige\_parents\_in\_1st\_leg (start\_trail in number, current\_trail in number, p\_sys\_conn\_by\_trail in varchar2, deep in number);

PROCEDURE get\_csr\_gige\_delivery;

RECURSIVE\_DEPTH number := 10;

END; -- Package spec

/

--------------------------------------------------------

-- DDL for Package LUCENT\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."LUCENT\_AUDIT" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: lucent\_audit

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 2/14/2011 1. Created this package.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

AUDIT\_TBL varchar2(30) := 'LUCENT\_NE\_VS\_NG\_AUDIT';

-- this is wrapper that calls the bottom 2 procs. done this way for backward compatibility

Procedure auditNgVsProspect;

PROCEDURE auditLucentBTSTerminations;

PROCEDURE generateLucentRegSummary;

PROCEDURE generateLucentSwitchSummary;

PROCEDURE truncate\_lucent\_wrk;

END lucent\_audit;

/

--------------------------------------------------------

-- DDL for Package MOTOROLA\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."MOTOROLA\_AUDIT" AS

-- this is wrapper that calls the bottom 2 procs. done this way for backward compatibility

PROCEDURE moto\_cdma\_audit;

PROCEDURE moto\_cdma\_regional\_summary;

PROCEDURE MOTO\_CDMA\_DEVICE\_SUMMARY;

PROCEDURE CREATE\_UBS\_AUDIT;

PROCEDURE moto\_evdo\_audit;

PROCEDURE moto\_evdo\_regional\_summary;

PROCEDURE MOTO\_EVDO\_DEVICE\_SUMMARY;

FUNCTION SET\_MESSAGE (fullMsg IN VARCHAR2, message IN VARCHAR2) RETURN VARCHAR2;

FUNCTION GET\_UBS\_MATCH\_STATUS (DS1\_NUMBER IN NUMBER, BANDWIDTH IN VARCHAR, BTS\_TYPE IN VARCHAR, IST1 IN INTEGER,

ISUBS IN INTEGER, PARSESTATS IN VARCHAR) RETURN VARCHAR2;

T1\_TYPE INTEGER := 1;

EBH\_TYPE INTEGER := 0;

UBS\_TYPE INTEGER := 1;

NON\_UBS\_TYPE INTEGER := 0;

CDMA\_DEVICE\_SUMM VARCHAR2(30) := 'MOTO\_CDMA\_AUDIT\_DEV\_SUMM';

CDMA\_REGION\_SUMM VARCHAR2(30) := 'MOTO\_CDMA\_AUDIT\_REG\_SUMM';

PROCEDURE copy\_moto\_cdma\_wrk;

PROCEDURE copy\_moto\_evdo\_wrk;

PROCEDURE truncate\_moto\_cdma\_wrk;

PROCEDURE truncate\_moto\_evdo\_wrk;

PROCEDURE restore\_moto\_cdma\_wrk;

PROCEDURE restore\_moto\_evdo\_wrk;

END MOTOROLA\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package MSPP\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."MSPP\_AUDIT" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: MSPP\_AUDIT

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 7/7/2011 Satya Modugula 1. Created this package.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

INC\_MSPP\_XCONN\_AUDIT\_DETAILS varchar2(50) := 'INC\_MSPP\_XCONN\_AUDIT\_DETAILS';

INC\_MSPP\_XCONN\_AUDIT\_SUMMARY varchar2(50) := 'INC\_MSPP\_XCONN\_AUDIT\_SUMMARY';

MSPP\_CIRC\_PATH\_ELEMENTS varchar2(50) := 'MSPP\_CIRC\_PATH\_ELEMENTS';

MSPP\_SUMMARY\_REGION\_MAP varchar2(50) := 'MSPP\_SUMMARY\_REGION\_MAP';

MSPP\_VSM\_MATCH varchar2(50) := 'MSPP\_VSM\_MATCH';

INC\_MSPP\_ADM\_COMP\_SUMMARY varchar2(50) := 'INC\_MSPP\_ADM\_COMP\_SUMMARY';

INC\_MSPP\_COMPLIANCE\_SUMMARY varchar2(50) := 'INC\_MSPP\_COMPLIANCE\_SUMMARY';

INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK varchar2(50) := 'INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK';

INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK varchar2(50) := 'INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK';

MSPP\_CIRC\_PATH\_ELEMENTS\_WK varchar2(50) := 'MSPP\_CIRC\_PATH\_ELEMENTS\_WK';

MSPP\_SUMMARY\_REGION\_MAP\_WK varchar2(50) := 'MSPP\_SUMMARY\_REGION\_MAP\_WK';

MSPP\_VSM\_MATCH\_WK varchar2(50) := 'MSPP\_VSM\_MATCH\_WK';

INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK varchar2(50) := 'INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK';

INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK varchar2(50) := 'INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK';

PROCEDURE copy\_mspp\_data\_from\_wk\_to\_prod;

PROCEDURE copy\_mspp\_data\_from\_prod\_to\_wk;

PROCEDURE truncate\_mspp\_wk;

END MSPP\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package NCM\_CSR\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."NCM\_CSR\_AUDIT" as

Procedure NCM\_AUDIT;

Procedure insert\_ci\_csr\_equip\_issues;

Procedure NCM\_CSR\_CLLI\_SUM\_LOAD;

Procedure NCM\_CSR\_REGN\_SUM\_LOAD;

END NCM\_CSR\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package NCM\_CSR\_VLAN\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."NCM\_CSR\_VLAN\_AUDIT" AS

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

END NCM\_CSR\_VLAN\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package NCM\_CSR\_VLAN\_AUDIT\_PKG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."NCM\_CSR\_VLAN\_AUDIT\_PKG" AS

procedure load\_csr\_vlan\_paths;

procedure audit\_ci\_csr\_vlans;

procedure insert\_ci\_csr\_vlan\_issues;

procedure assign\_area\_region\_to\_csr\_vlan;

procedure load\_vlan\_csr\_clli\_summary;

procedure load\_csr\_vlan\_regional\_summ;

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

END NCM\_CSR\_VLAN\_AUDIT\_pkg;

/

--------------------------------------------------------

-- DDL for Package NETSMART\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."NETSMART\_AUDIT" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: netsmart\_audit

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 5/2/2012 NSP Created this package.

1.1 5/11/2012 SME Realized detail query and

added report table functions

1.2 3/9/2015 SME Split audit into MSPP and ADM

methods.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--

-- MSPP Methods

--

PROCEDURE GETNAUTILUSEQUIP;

PROCEDURE AUDITFWEQUIP;

PROCEDURE GENSUMMARY;

PROCEDURE AUDIT\_XCONNECTS;

FUNCTION NS\_SUMMARY\_TBL RETURN NETSMART\_SUMMARY\_TBL PIPELINED;

FUNCTION NS\_DETAIL\_TBL(SUB\_MARKET\_IN IN VARCHAR2, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED;

FUNCTION NS\_XCONNECT\_TBL (TID\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED;

FUNCTION NS\_XCONNECT\_SUMMARY (SUB\_MARKET\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED;

/\* This is used to build the realized Summary Report table \*/

FUNCTION NS\_DETAIL\_TBL\_WRK(START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED;

--

-- ADM Methods

--

PROCEDURE GETADMNAUTILUSEQUIP;

PROCEDURE AUDITADMFWEQUIP;

PROCEDURE GENADMSUMMARY;

PROCEDURE AUDIT\_ADM\_XCONNECTS;

FUNCTION NS\_ADM\_SUMMARY\_TBL RETURN NETSMART\_SUMMARY\_TBL PIPELINED;

FUNCTION NS\_ADM\_DETAIL\_TBL(SUB\_MARKET\_IN IN VARCHAR2, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED;

FUNCTION NS\_ADM\_XCONNECT\_TBL (TID\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED;

FUNCTION NS\_ADM\_XCONNECT\_SUMMARY (SUB\_MARKET\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED;

/\* This is used to build the realized Summary Report table \*/

FUNCTION NS\_ADM\_DETAIL\_TBL\_WRK(START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED;

--

-- Both MSPP and ADM

--

PROCEDURE RUNAUDIT;

PROCEDURE TRUNCATE\_WK;

END NETSMART\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package NGMLS\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."NGMLS\_AUDIT" AS

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

PROCEDURE audit\_ngMLS\_equip;

PROCEDURE assign\_area\_region\_to\_devices;

PROCEDURE insert\_ngMLS\_equip\_issues;

PROCEDURE load\_ntls\_ngMLS\_vlan\_paths;

PROCEDURE audit\_CI\_ngMLS\_vlans;

PROCEDURE audit\_al\_ngmls\_vlans;

PROCEDURE insert\_CI\_ngMLS\_vlan\_issues;

PROCEDURE load\_CI\_vlan\_device\_summary;

PROCEDURE load\_al\_vlan\_device\_summary;

PROCEDURE load\_vlan\_regional\_summary;

PROCEDURE load\_ngMLS\_regional\_summary;

END NGMLS\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package NORTEL\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."NORTEL\_AUDIT" IS

-- Purpose: To create nortel audit tables for voice spans

NT\_DOM\_INV varchar2(30) := 'NORTEL\_DOM\_INVENTORY';

NT\_EVDO\_PORT\_INV varchar2(30) := 'NORTEL\_PORT\_INVENTORY';

NG\_NT\_PARSED\_CELL\_PATHS varchar2(30) := 'VZW\_NT\_EVDO\_CELL\_PATHS';

NORTEL\_NE\_VS\_NG\_AUDIT varchar2(30) := 'NORTEL\_NE\_VS\_NG\_AUDIT';

PROCEDURE nt\_cdma\_audit;

PROCEDURE nt\_evdo\_audit;

NORTEL\_CDMA\_AUDIT\_REG\_SUMM varchar2(30) := 'NORTEL\_CDMA\_AUDIT\_REG\_SUMM';

NORTEL\_CDMA\_AUDIT\_DEV\_SUMM varchar2(30) := 'NORTEL\_CDMA\_AUDIT\_DEV\_SUMM';

NT\_EVDO\_BTS\_TERM\_REG\_SUMM varchar2(30) := 'NT\_EVDO\_BTS\_TERM\_REG\_SUMM';

NT\_EVDO\_BTS\_TERM\_EMS\_SUMM varchar2(30) := 'NT\_EVDO\_BTS\_TERM\_EMS\_SUMM';

PROCEDURE populate\_cdma\_regional\_summary;

PROCEDURE populate\_cdma\_device\_summary;

PROCEDURE populate\_evdo\_regional\_summary;

PROCEDURE populate\_evdo\_device\_summary;

END; -- Package spec

/

--------------------------------------------------------

-- DDL for Package NTLS

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."NTLS" is

procedure home(advSearch in varchar2 default 'N', pwdWarning in varchar2 default 'N' ) ;

FUNCTION authorize return boolean;

end ntls;

/

--------------------------------------------------------

-- DDL for Package RPT\_IAN\_NOCC\_SITE\_TEST

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."RPT\_IAN\_NOCC\_SITE\_TEST"

IS

--

-- Purpose: NOCC SITE ALARAMS AGGREAGATE SUMMARY and DETAIL reports.

--

-- MODIFICATION HISTORY

-- Person Date Comments

-- ---------------------- ------------ -----------------------------------------

-- Krishna Gadde Oct 2005 Initial release.

-- Krihsna Gadde March 2006 Tunned...

-- Steve Ervin Mar 2016 Convert to Territory/Market/SubMarket

-- Steve Ervin Jun 2016 Add support for Lighting Test Period UDA

--/\* ddl

--CREATE TABLE xng\_reports.nocc\_last\_test\_summary

-- (area VARCHAR2(40) NOT NULL,

-- region VARCHAR2(80) NOT NULL,

-- domain\_id VARCHAR2(80) NOT NULL,

-- domain\_name VARCHAR2(100) NOT NULL,

-- sites NUMBER(9,0),

-- antennas NUMBER(9,0),

-- vzw\_owned NUMBER(9,0),

-- lights\_reqd NUMBER(9,0),

-- lights\_tested NUMBER(9,0),

-- lights\_test\_overdue NUMBER(9,0),

-- lights\_test\_ok NUMBER(9,0),

-- nocc\_monitored\_sites NUMBER(9,0),

-- env\_tested NUMBER(9,0),

-- env\_test\_overdue NUMBER(9,0),

-- env\_test\_ok NUMBER(9,0),

-- run\_date DATE NOT NULL)

--/

--

--CREATE INDEX xng\_reports.indx\_last\_test\_summary ON xng\_reports.nocc\_last\_test\_summary

-- (

-- "RUN\_DATE" DESC,

-- domain\_id ASC

-- )

--/

--

--

--CREATE TABLE xng\_reports.nocc\_last\_test\_details

-- (domain\_id NUMBER(9,0) NOT NULL,

-- domain\_name VARCHAR2(50) NOT NULL,

-- site\_inst\_id NUMBER(9,0) NOT NULL,

-- site\_name VARCHAR2(100) NOT NULL,

-- site\_id VARCHAR2(30),

-- site\_tech\_name VARCHAR2(100),

-- inventoried\_cells VARCHAR2(1000),

-- site\_type VARCHAR2(30),

-- nocc\_monitored VARCHAR2(30),

-- light\_last\_tested VARCHAR2(30),

-- light\_next\_test VARCHAR2(30),

-- light\_test\_overdue VARCHAR2(30),

-- env\_last\_tested VARCHAR2(30),

-- env\_next\_test VARCHAR2(30),

-- env\_test\_overdue VARCHAR2(30),

-- run\_date DATE)

--/

--

--CREATE INDEX xng\_reports.indx\_last\_test\_details ON xng\_reports.nocc\_last\_test\_details

-- (

-- "RUN\_DATE" DESC,

-- domain\_id ASC,

-- site\_inst\_id ASC

-- )

--/

--

---- Schedule an Oracle job to run everyday... every hour from 6am to 6pm in weekdays.

--

--BEGIN

-- DECLARE

-- v\_JobNum NUMBER;

-- BEGIN

-- DBMS\_JOB.SUBMIT (v\_JobNum,

-- 'RPT\_NOCC\_SITE\_TEST.getAllDomains();',

-- sysdate,

-- 'case when to\_char(sysdate, ''hh24'') between ''06'' and ''18'' then trunc(sysdate,''hh'')+1/24 when to\_char(SYSDATE, ''DAY'') = ''FRIDAY'' then trunc(sysdate+3)+6/24 else trunc(sysdate+1)+6/24 end',

-- TRUE );

-- commit;

--

-- EXCEPTION

-- WHEN OTHERS THEN

-- dbms\_output.put\_line(SubStr('Error '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

-- RAISE;

-- END;

-- END;

--

--

--

--\*/

PROCEDURE getAllDomains; -- This will load all the test data. Starting point.

PROCEDURE GETALLSITES (

P\_DOMAINID IN VZWNET.DOMAIN\_INST.DOMAIN\_INST\_ID%TYPE,

P\_DOMAINNAME IN VZWNET.DOMAIN\_INST.DOMAIN\_NAME%TYPE,

P\_Territory In Xng\_Reports.Vzw\_Network\_Org.Territory%Type,

P\_Market\_Territory In Xng\_Reports.Vzw\_Network\_Org.Market\_Territory%Type,

P\_SUB\_MARKET IN XNG\_REPORTS.VZW\_NETWORK\_ORG.SUB\_MARKET%TYPE);

FUNCTION getSiteTechName (

p\_SiteInstId IN vzwnet.site\_attr\_settings.site\_inst\_id%TYPE )

RETURN vzwnet.site\_attr\_settings.attr\_value%TYPE;

FUNCTION getCellEquipDesc(

p\_SiteInstId IN vzwnet.Site\_Inst.Site\_Inst\_Id%TYPE )

RETURN NOCC\_LAST\_TEST\_DETAILS.INVENTORIED\_CELLS%TYPE;

FUNCTION getIsNoccMonitored (

p\_SiteInstId IN vzwnet.site\_inst.site\_inst\_id%TYPE )

RETURN VARCHAR2;

FUNCTION getFailedFirstEnvAttempt (

p\_SiteInstId IN vzwnet.site\_inst.site\_inst\_id%TYPE )

RETURN VARCHAR2;

FUNCTION getENVLastTestedDate (

p\_SiteInstId IN vzwnet.site\_attr\_settings.site\_inst\_id%TYPE,

p\_SiteType IN vzwnet.val\_attr\_group.class\_category%TYPE )

RETURN vzwnet.site\_attr\_settings.attr\_value%TYPE;

FUNCTION getHasAntennaTower(

p\_SiteInstId IN vzwnet.Site\_Inst.Site\_Inst\_Id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE )

RETURN VARCHAR2;

FUNCTION getIsAntennaVZWOwned (

p\_SiteInstId IN vzwnet.equip\_inst.site\_inst\_id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE )

RETURN vzwnet.equip\_attr\_settings.attr\_value%TYPE;

FUNCTION getisLightingRequired (

p\_SiteInstId IN vzwnet.equip\_inst.site\_inst\_id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE )

RETURN vzwnet.equip\_attr\_settings.attr\_value%TYPE;

FUNCTION getisLightingVoluntary (

p\_SiteInstId IN vzwnet.equip\_inst.site\_inst\_id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE )

RETURN vzwnet.equip\_attr\_settings.attr\_value%TYPE;

FUNCTION getLightingTestPeriod (

p\_SiteInstId IN vzwnet.equip\_inst.site\_inst\_id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE )

RETURN vzwnet.equip\_attr\_settings.attr\_value%TYPE;

FUNCTION getTowerLightLastTestedDate (

p\_SiteInstId IN vzwnet.site\_attr\_settings.site\_inst\_id%TYPE,

p\_SiteType IN vzwnet.val\_attr\_group.class\_category%TYPE )

RETURN vzwnet.site\_attr\_settings.attr\_value%TYPE;

PROCEDURE INSERT\_NOCC\_LAST\_TEST\_SUMMARY(

P\_TERRITORY IN NOCC\_LAST\_TEST\_SUMMARY.TERRITORY%TYPE,

P\_MARKET\_TERRITORY IN NOCC\_LAST\_TEST\_SUMMARY.MARKET\_TERRITORY%TYPE,

P\_Sub\_Market In Nocc\_Last\_Test\_Summary.Sub\_Market%Type,

P\_DomainId In Nocc\_Last\_Test\_Summary.Domain\_Id%Type,

P\_DOMAINNAME IN NOCC\_LAST\_TEST\_SUMMARY.DOMAIN\_NAME%TYPE,

P\_SITES IN NOCC\_LAST\_TEST\_SUMMARY.SITES%TYPE,

P\_ANTENNAS IN NOCC\_LAST\_TEST\_SUMMARY.ANTENNAS%TYPE,

p\_VZWOwned IN NOCC\_LAST\_TEST\_SUMMARY.vzw\_owned%TYPE,

P\_LIGHTSREQD IN NOCC\_LAST\_TEST\_SUMMARY.LIGHTS\_REQD%TYPE,

p\_LightsTested IN NOCC\_LAST\_TEST\_SUMMARY.lights\_tested%TYPE,

P\_LIGHTSOVERDUE IN NOCC\_LAST\_TEST\_SUMMARY.LIGHTS\_TEST\_OVERDUE%TYPE,

P\_LIGHTSOK IN NOCC\_LAST\_TEST\_SUMMARY.LIGHTS\_TEST\_OK%TYPE,

P\_MONITORED IN NOCC\_LAST\_TEST\_SUMMARY.NOCC\_MONITORED\_SITES%TYPE,

P\_ENVTESTED IN NOCC\_LAST\_TEST\_SUMMARY.ENV\_TESTED%TYPE,

P\_ENVOVERDUE IN NOCC\_LAST\_TEST\_SUMMARY.ENV\_TEST\_OVERDUE%TYPE,

p\_EnvOK IN NOCC\_LAST\_TEST\_SUMMARY.env\_test\_ok%TYPE);

PROCEDURE insert\_NOCC\_Last\_Test\_Details(

p\_DomainId IN NOCC\_LAST\_TEST\_DETAILS.domain\_Id%TYPE,

p\_DomainName IN NOCC\_LAST\_TEST\_DETAILS.domain\_name%TYPE,

p\_SiteInstId IN NOCC\_LAST\_TEST\_DETAILS.site\_Inst\_Id%TYPE,

p\_SiteName IN NOCC\_LAST\_TEST\_DETAILS.site\_name%TYPE,

p\_SiteId IN NOCC\_LAST\_TEST\_DETAILS.site\_id%TYPE,

p\_Sitetype IN NOCC\_LAST\_TEST\_DETAILS.site\_type%TYPE,

p\_SiteTechName IN NOCC\_LAST\_TEST\_DETAILS.site\_tech\_name%TYPE,

p\_inventoried\_cells IN NOCC\_LAST\_TEST\_DETAILS.inventoried\_cells%TYPE,

p\_NOCC\_Monitored IN NOCC\_LAST\_TEST\_DETAILS.nocc\_Monitored%TYPE,

p\_vzw\_owned IN NOCC\_LAST\_TEST\_DETAILS.vzw\_owned%TYPE,

p\_lights\_reqd IN NOCC\_LAST\_TEST\_DETAILS.lights\_reqd%TYPE,

p\_LightLastTested IN NOCC\_LAST\_TEST\_DETAILS.Light\_last\_tested%TYPE,

p\_LightNextTest IN NOCC\_LAST\_TEST\_DETAILS.Light\_next\_test%TYPE,

p\_LightTestOverDue IN NOCC\_LAST\_TEST\_DETAILS.light\_test\_overdue%TYPE,

p\_ENVLastTested IN NOCC\_LAST\_TEST\_DETAILS.env\_last\_tested%TYPE,

p\_ENVNextTest IN NOCC\_LAST\_TEST\_DETAILS.env\_next\_test%TYPE,

p\_ENVTestOverDue IN NOCC\_LAST\_TEST\_DETAILS.env\_test\_overdue%TYPE,

p\_EquipInstid IN NOCC\_LAST\_TEST\_DETAILS.equip\_inst\_id%TYPE,

p\_LightsVoluntary IN NOCC\_LAST\_TEST\_DETAILS.lights\_voluntary%TYPE);

FUNCTION varchar2date( p\_StrDate IN VARCHAR2 ) RETURN DATE;

PROCEDURE removeDetails( p\_DomainId IN NOCC\_LAST\_TEST\_SUMMARY.domain\_id%TYPE );

END; -- Package spec

/

--------------------------------------------------------

-- DDL for Package SAM\_CSR\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."SAM\_CSR\_AUDIT" AS

PROCEDURE audit\_al\_csr\_equip;

PROCEDURE insert\_al\_csr\_equip\_issues;

PROCEDURE assign\_area\_region\_to\_csr;

PROCEDURE load\_csr\_clli\_summary;

PROCEDURE load\_csr\_regional\_summary;

PROCEDURE load\_csr\_vlan\_paths;

PROCEDURE audit\_al\_csr\_vlans;

PROCEDURE insert\_al\_CSR\_vlan\_issues;

PROCEDURE assign\_area\_region\_to\_csr\_vlan;

PROCEDURE load\_vlan\_csr\_clli\_summary;

PROCEDURE load\_csr\_vlan\_regional\_summ;

END SAM\_CSR\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package SITE\_PORTAL\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."SITE\_PORTAL\_AUDIT" AS

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

PROCEDURE UPDATESITEPORTALAUDIT;

PROCEDURE UPDATESITEPORTALAUDITSUMMARY;

PROCEDURE RESTORE\_SITEPORTALAUDIT;

AUDIT\_TABLE\_NAME CONSTANT varchar2(30) := 'NE\_VS\_NT\_SP\_AUDIT';

PROD\_AUDIT\_TABLE\_NAME CONSTANT varchar2(30) := 'NE\_VS\_NT\_SP\_AUDIT';

AUDIT\_SUMMARY\_TABLE\_NAME CONSTANT varchar2(30) := 'NE\_VS\_NT\_SP\_SUMMARY';

PROD\_AUDIT\_SUMMARY\_TABLE\_NAME CONSTANT varchar2(30) := 'NE\_VS\_NT\_SP\_SUMMARY';

END SITE\_PORTAL\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package XPERWEB

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."XPERWEB" is

procedure home(userName in varchar2 default NULL );

procedure path\_def(iPathInstId in varchar2, userName in varchar2 default NULL ) ;

FUNCTION authorize return boolean;

end xperweb;

/

--------------------------------------------------------

-- DDL for Package XWEB\_EQUIP

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."XWEB\_EQUIP" AS

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

procedure home(userName in varchar2 default NULL ) ;

procedure equip\_def(ieqInstId in varchar2, userName in varchar2 default NULL ) ;

FUNCTION authorize return boolean;

END XWEB\_EQUIP;

/

--------------------------------------------------------

-- DDL for Package XWEB\_SEG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."XWEB\_SEG" AS

/\* TODO enter package declarations (types, exceptions, methods etc) here \*/

procedure home(userName in varchar2 default NULL ) ;

procedure segment\_def(iSegInstId in varchar2, userName in varchar2 default NULL ) ;

FUNCTION authorize return boolean;

END XWEB\_seg;

/

--------------------------------------------------------

-- DDL for Package XWEB\_SITE

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE "NG\_REPORTS"."XWEB\_SITE" is

procedure home(userName in varchar2 default NULL ) ;

procedure site\_def(siteInstId in varchar2, userName in varchar2 default NULL ) ;

FUNCTION authorize return boolean;

end xweb\_site;

/

--------------------------------------------------------

-- DDL for Package Body BTOD\_AUDIT\_PKG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."BTOD\_AUDIT\_PKG" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: load btod ntls audit tables

\* Process:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* procedure that runs audit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Procedure runAudit is

v\_btod\_wrk\_ck number;

v\_btod\_date\_ck varchar2(1);

v\_ntls\_tbl\_ck number;

v\_btod\_tbl\_ck number;

BEGIN

BEGIN

--make sure Granite data is current - this is set up in nis\_extractor too

ng\_reports.load\_segments\_stripped;

v\_ntls\_tbl\_ck := 0 ;

v\_btod\_tbl\_ck := 0 ;

v\_ntls\_tbl\_ck := 0;

v\_btod\_tbl\_ck := 0;

select count(\*) into v\_ntls\_tbl\_ck from ng\_reports.stripped\_segment\_data;

select count(\*) into v\_btod\_tbl\_ck from ng\_reports.btod\_audit\_data;

if v\_ntls\_tbl\_ck <> 0 and v\_btod\_tbl\_ck <> 0 then

load\_btod\_ntls\_audit\_detail();

load\_btod\_ntls\_audit\_summary();

else

if v\_ntls\_tbl\_ck = 0 then

dbms\_output.put\_line(SubStr('Error: btod\_ntls\_audit\_upd(): stripped\_segment\_data table empty', 1, 255));

end if;

if v\_btod\_tbl\_ck = 0 then

dbms\_output.put\_line(SubStr('Error: btod\_ntls\_audit\_upd(): btod\_audit\_data table empty', 1, 255));

end if;

end if;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr('Error: btod\_ntls\_audit\_upd(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

END;

END runAudit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Loads BTOD NTLS Audit Detail

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Procedure load\_btod\_ntls\_audit\_detail is

v\_seg\_id\_hld ng\_reports.stripped\_segment\_data.ntls\_segment\_id%type;

v\_bd\_hld varchar2(9);

v\_cost\_hld ng\_reports.stripped\_segment\_data.ntls\_recur\_costs%type;

v\_ban\_hld ng\_reports.stripped\_segment\_data.ntls\_ban%type;

v\_costban\_upd char(1);

v\_seg\_cnt number;

v\_cnt number;

v\_cost\_upd\_cnt number;

v\_ban\_upd\_cnt number;

v\_date\_upd\_cnt number;

v\_date\_add\_cnt number;

v\_modby varchar2(10);

sql\_stmt VARCHAR2(32767);

v\_do\_upds char(1);

v\_upds\_done ng\_reports.btod\_ntls\_audit\_wrk.match\_status%type;

v\_sdate date;

cursor vend\_cursor is

select nvl(dom.ntls\_territory, 'Unknown') as ntls\_territory, nvl(dom.ntls\_market, 'Unknown') as ntls\_market,

nvl(dom.ntls\_sub\_market, 'Unknown') as ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, ntls\_segment\_id, ntls\_segment\_name, ntls\_vendor, ntls\_status, ntls.ntls\_seg\_name\_strip, ntls\_bandwidth,

ntls\_type, ntls\_recur\_costs, ntls\_ban, ntls\_bill\_date, ntls.ntls\_vendor\_strip, ntls\_bill\_date\_char, ntls\_curr\_path\_inst\_id,

ntls\_next\_path\_inst\_id

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and ntls\_seg\_unq = 'Y'

and processed is null;

vend\_row vend\_cursor%ROWTYPE;

cursor nomatch\_cursor is

select ntls\_territory, ntls\_market, ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip, ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban,

ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code, btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year,

manual\_entry, ban\_custom\_field\_3, serv\_estab\_date, seqno

from (select ntls\_territory, ntls\_market, ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip, ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban,

ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code, btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year,

manual\_entry, ban\_custom\_field\_3, serv\_estab\_date, seqno, max(ntlsck) as ntlsck

from (select nvl(dom.ntls\_territory, 'Unknown') as ntls\_territory, nvl(dom.ntls\_market, 'Unknown') as ntls\_market,

nvl(dom.ntls\_sub\_market, 'Unknown') as ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, nvl(ntls\_segment\_id,0) as ntlsck

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip (+)

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and processed is null)

group by ntls\_territory, ntls\_market, ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip, ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban,

ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code, btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year,

manual\_entry, ban\_custom\_field\_3, serv\_estab\_date, seqno)

where ntlsck = 0;

nomatch\_row nomatch\_cursor%ROWTYPE;

cursor multi\_cursor is

select distinct ntls\_territory, ntls\_market, ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, ntls\_segment\_id, ntls\_segment\_name, ntls\_vendor, ntls\_status, ntls\_seg\_name\_strip, ntls\_bandwidth,

ntls\_type, ntls\_recur\_costs, ntls\_ban, ntls\_bill\_date, ntls\_vendor\_strip, ntls\_bill\_date\_char, ntls\_curr\_path\_inst\_id,

ntls\_next\_path\_inst\_id

from (select distinct a.\*, nvl(n.btod\_circ\_strip,'nfd') as ckprc

from (select nvl(dom.ntls\_territory, 'Unknown') as ntls\_territory, nvl(dom.ntls\_market, 'Unknown') as ntls\_market,

nvl(dom.ntls\_sub\_market, 'Unknown') as ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, ntls\_segment\_id, ntls\_segment\_name, ntls\_vendor, ntls\_status, ntls.ntls\_seg\_name\_strip, ntls\_bandwidth,

ntls\_type, ntls\_recur\_costs, ntls\_ban, ntls\_bill\_date, ntls.ntls\_vendor\_strip, ntls\_bill\_date\_char, ntls\_curr\_path\_inst\_id,

ntls\_next\_path\_inst\_id

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.inventory\_vendor\_code = ntls.ntls\_vendor\_strip

and btod.btod\_ban\_strip = ntls.ntls\_ban\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and processed is null) a,

(select ntls\_seg\_name\_strip, ntls\_vendor\_strip, ntls\_ban\_strip, count(ntls\_segment\_id) as mcnt

from ng\_reports.stripped\_segment\_data

group by ntls\_seg\_name\_strip, ntls\_vendor\_strip, ntls\_ban\_strip

having count(ntls\_segment\_id) > 1) b,

(select btod\_circ\_strip, inventory\_vendor\_code, btod\_ban\_strip from ng\_reports.btod\_audit\_data\_process where processed <> 'M' and stripped\_ec\_circuit\_id = '15HCGS169138CTE') n

where a.btod\_circ\_strip = b.ntls\_seg\_name\_strip

and a.inventory\_vendor\_code = b.ntls\_vendor\_strip

and a.btod\_ban\_strip = b.ntls\_ban\_strip

and a.btod\_circ\_strip = n.btod\_circ\_strip (+)

and a.inventory\_vendor\_code = n.inventory\_vendor\_code (+)

and a.btod\_ban\_strip = n.btod\_ban\_strip (+))

where ckprc = 'nfd';

multi\_row multi\_cursor%ROWTYPE;

cursor multi1\_cursor is

select distinct ntls\_territory, ntls\_market, ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, ntls\_segment\_id, ntls\_segment\_name, ntls\_vendor, ntls\_status, ntls\_seg\_name\_strip, ntls\_bandwidth,

ntls\_type, ntls\_recur\_costs, ntls\_ban, ntls\_bill\_date, ntls\_vendor\_strip, ntls\_bill\_date\_char, ntls\_curr\_path\_inst\_id,

ntls\_next\_path\_inst\_id

from (select distinct a.\*, nvl(n.btod\_circ\_strip,'nfd') as ckprc

from (select nvl(dom.ntls\_territory, 'Unknown') as ntls\_territory, nvl(dom.ntls\_market, 'Unknown') as ntls\_market,

nvl(dom.ntls\_sub\_market, 'Unknown') as ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, ntls\_segment\_id, ntls\_segment\_name, ntls\_vendor, ntls\_status, ntls.ntls\_seg\_name\_strip, ntls\_bandwidth,

ntls\_type, ntls\_recur\_costs, ntls\_ban, ntls\_bill\_date, ntls.ntls\_vendor\_strip, ntls\_bill\_date\_char, ntls\_curr\_path\_inst\_id,

ntls\_next\_path\_inst\_id

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.inventory\_vendor\_code = ntls.ntls\_vendor\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and processed is null) a,

(select ntls\_seg\_name\_strip, ntls\_vendor\_strip, count(ntls\_segment\_id) as mcnt

from ng\_reports.stripped\_segment\_data

group by ntls\_seg\_name\_strip, ntls\_vendor\_strip

having count(ntls\_segment\_id) > 1) b,

(select btod\_circ\_strip, inventory\_vendor\_code, btod\_ban\_strip from ng\_reports.btod\_audit\_data\_process where processed <> 'M' and stripped\_ec\_circuit\_id = '15HCGS169138CTE') n

where a.btod\_circ\_strip = b.ntls\_seg\_name\_strip

and a.inventory\_vendor\_code = b.ntls\_vendor\_strip

and a.btod\_circ\_strip = n.btod\_circ\_strip (+)

and a.inventory\_vendor\_code = n.inventory\_vendor\_code (+)

and a.btod\_ban\_strip = n.btod\_ban\_strip (+))

where ckprc = 'nfd';

multi1\_row multi1\_cursor%ROWTYPE;

cursor multi2\_cursor is

select distinct ntls\_territory, ntls\_market, ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, ntls\_segment\_id, ntls\_segment\_name, ntls\_vendor, ntls\_status, ntls\_seg\_name\_strip, ntls\_bandwidth,

ntls\_type, ntls\_recur\_costs, ntls\_ban, ntls\_bill\_date, ntls\_vendor\_strip, ntls\_bill\_date\_char, ntls\_curr\_path\_inst\_id,

ntls\_next\_path\_inst\_id

from (select distinct a.\*, nvl(n.btod\_circ\_strip,'nfd') as ckprc

from (select nvl(dom.ntls\_territory, 'Unknown') as ntls\_territory, nvl(dom.ntls\_market, 'Unknown') as ntls\_market,

nvl(dom.ntls\_sub\_market, 'Unknown') as ntls\_sub\_market, stripped\_ec\_circuit\_id, btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, btod\_amt, btod\_ban\_strip, inventory\_vendor\_code,

btod\_bd, btod\_extract\_date, ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

serv\_estab\_date, seqno, ntls\_segment\_id, ntls\_segment\_name, ntls\_vendor, ntls\_status, ntls.ntls\_seg\_name\_strip, ntls\_bandwidth,

ntls\_type, ntls\_recur\_costs, ntls\_ban, ntls\_bill\_date, ntls.ntls\_vendor\_strip, ntls\_bill\_date\_char, ntls\_curr\_path\_inst\_id,

ntls\_next\_path\_inst\_id

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and processed is null) a,

(select ntls\_seg\_name\_strip, count(ntls\_segment\_id) as mcnt

from ng\_reports.stripped\_segment\_data

group by ntls\_seg\_name\_strip

having count(ntls\_segment\_id) > 1) b,

(select btod\_circ\_strip, inventory\_vendor\_code, btod\_ban\_strip from ng\_reports.btod\_audit\_data\_process where processed <> 'M' and stripped\_ec\_circuit\_id = '15HCGS169138CTE') n

where a.btod\_circ\_strip = b.ntls\_seg\_name\_strip

and a.btod\_circ\_strip = n.btod\_circ\_strip (+)

and a.inventory\_vendor\_code = n.inventory\_vendor\_code (+)

and a.btod\_ban\_strip = n.btod\_ban\_strip (+))

where ckprc = 'nfd';

multi2\_row multi2\_cursor%ROWTYPE;

BEGIN

v\_cnt := 0;

v\_seg\_id\_hld := 0;

v\_seg\_cnt := 0;

v\_sdate := sysdate;

execute immediate 'truncate table ng\_reports.btod\_audit\_data\_process reuse storage'; commit;

insert into ng\_reports.btod\_audit\_data\_process

(select stripped\_ec\_circuit\_id, ng\_reports.strip\_segment\_name(stripped\_ec\_circuit\_id) as btod\_circ\_strip,

ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban,

to\_number(replace(replace(charge\_amount,'$',''),',','')) as charge\_amount,

replace(translate(ban, '-/,.()[]\*$;?#&~:=`<>\^@!"'||chr(160), ' '), ' ', '') as btod\_ban\_strip,

inventory\_vendor\_code,

trunc(to\_date(replace(bill\_date,' 00:00:00',''),'yyyy/mm/dd hh:mi:ss')) as btod\_bd,

replace(bill\_date,' 00:00:00','') as btod\_bd\_chr,

to\_date(btod\_extract\_date,'yyyymmdd') as btod\_extract\_date,

ban\_custom\_field\_4, bill\_date\_month, bill\_date\_year, manual\_entry, ban\_custom\_field\_3,

trunc(to\_date(replace(serv\_estab\_date,' 00:00:00',''),'yyyy/mm/dd hh:mi:ss')) as serv\_estab\_date,

rownum as seqno, null

from ng\_reports.btod\_audit\_data

where stripped\_ec\_circuit\_id is not null);

commit;

execute immediate 'truncate table ng\_reports.btod\_ntls\_audit\_wrk reuse storage'; commit;

insert into ng\_reports.btod\_ntls\_audit\_wrk

(select ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban,

ban\_custom\_field\_4, stripped\_ec\_circuit\_id, inventory\_vendor\_code, btod\_amt,

bill\_date\_month, bill\_date\_year, manual\_entry, btod\_extract\_date,

'Y', nvl(dom.ntls\_territory, 'Unknown') as ntls\_territory, nvl(dom.ntls\_market, 'Unknown') as ntls\_market,

nvl(dom.ntls\_sub\_market, 'Unknown') as ntls\_sub\_market,

ntls\_segment\_id, ntls\_segment\_name, ntls\_type, ntls\_vendor, ntls\_bandwidth,

ntls\_status, ntls\_curr\_path\_inst\_id, ntls\_next\_path\_inst\_id, trunc(sysdate), 'Y', btod\_bd,

ban\_custom\_field\_3, serv\_estab\_date, 'Both', null, ntls\_ban

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.inventory\_vendor\_code = ntls.ntls\_vendor\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and ntls\_seg\_vendor\_unq = 'Y'

union

select ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban,

ban\_custom\_field\_4, stripped\_ec\_circuit\_id, inventory\_vendor\_code, btod\_amt,

bill\_date\_month, bill\_date\_year, manual\_entry, btod\_extract\_date,

'Y', nvl(dom.ntls\_territory, 'Unknown') as ntls\_territory, nvl(dom.ntls\_market, 'Unknown') as ntls\_market,

nvl(dom.ntls\_sub\_market, 'Unknown') as ntls\_sub\_market, ntls\_segment\_id, ntls\_segment\_name, ntls\_type, ntls\_vendor, ntls\_bandwidth,

ntls\_status, ntls\_curr\_path\_inst\_id, ntls\_next\_path\_inst\_id, trunc(sysdate), 'Y', btod\_bd, ban\_custom\_field\_3, serv\_estab\_date,

'Both', null, ntls\_ban

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.inventory\_vendor\_code = ntls.ntls\_vendor\_strip

and btod.btod\_ban\_strip = ntls.ntls\_ban\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and ntls\_seg\_vendor\_ban\_unq = 'Y');

commit;

update ng\_reports.btod\_audit\_data\_process bdp set processed = 'B'

where exists (select 1

from (select seqno

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.inventory\_vendor\_code = ntls.ntls\_vendor\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and ntls\_seg\_vendor\_unq = 'Y'

union

select seqno

from ng\_reports.btod\_audit\_data\_process btod,

ng\_reports.stripped\_segment\_data ntls,

(select bxd.btod\_ban\_category\_1, drr.territory as ntls\_territory, drr.market\_territory as ntls\_market,

drr.sub\_market as ntls\_sub\_market, drr.domain\_id

from ng\_reports.btod\_ntls\_domain\_map bxd, ng\_reports.domains\_regional\_reporting drr

where bxd.ntls\_domain\_id = drr.domain\_id (+)) dom

where btod.btod\_circ\_strip = ntls.ntls\_seg\_name\_strip

and btod.inventory\_vendor\_code = ntls.ntls\_vendor\_strip

and btod.btod\_ban\_strip = ntls.ntls\_ban\_strip

and btod.ban\_category\_1 = dom.btod\_ban\_category\_1 (+)

and ntls\_seg\_vendor\_ban\_unq = 'Y') upd

where bdp.seqno = upd.seqno);

commit;

OPEN vend\_cursor;

LOOP

FETCH vend\_cursor

INTO vend\_row;

EXIT WHEN vend\_cursor%NOTFOUND;

insert into ng\_reports.btod\_ntls\_audit\_wrk (ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, ban\_custom\_field\_4,

stripped\_ec\_circuit\_id, inventory\_vendor\_code, charge\_amount, bill\_date\_month, bill\_date\_year,

manual\_entry, btod\_extract\_date, bill\_date, ban\_custom\_field\_3, serv\_estab\_date, ntls\_territory,

ntls\_market, ntls\_sub\_market, ntls\_segment\_id, ntls\_segment\_name, ntls\_type, ntls\_vendor,

ntls\_bandwidth, ntls\_status, ntls\_curr\_path\_inst\_id, ntls\_ban, ntls\_next\_path\_inst\_id,

ntls\_extract\_date, btod\_ntls\_match, uniquely\_identified, match\_code, match\_status)

values (vend\_row.ban\_category\_1, vend\_row.ban\_category\_2, vend\_row.ban\_custom\_field\_1, vend\_row.master\_ban, vend\_row.ban,

vend\_row.ban\_custom\_field\_4, vend\_row.stripped\_ec\_circuit\_id, vend\_row.inventory\_vendor\_code, vend\_row.btod\_amt,

vend\_row.bill\_date\_month, vend\_row.bill\_date\_year, vend\_row.manual\_entry, vend\_row.btod\_extract\_date,

vend\_row.btod\_bd, vend\_row.ban\_custom\_field\_3, vend\_row.serv\_estab\_date, vend\_row.ntls\_territory, vend\_row.ntls\_market,

vend\_row.ntls\_sub\_market, vend\_row.ntls\_segment\_id, vend\_row.ntls\_segment\_name, vend\_row.ntls\_type, vend\_row.ntls\_vendor,

vend\_row.ntls\_bandwidth, vend\_row.ntls\_status, vend\_row.ntls\_curr\_path\_inst\_id, vend\_row.ntls\_ban, vend\_row.ntls\_next\_path\_inst\_id,

trunc(sysdate), 'N', 'Y','BTOD Only', 'Vendor Mismatch');

update ng\_reports.btod\_audit\_data\_process

set processed = 'V'

where seqno = vend\_row.seqno;

v\_cnt := v\_cnt + 1;

if v\_cnt > 5000 then

commit;

v\_cnt := 0;

end if;

END LOOP;

CLOSE vend\_cursor;

COMMIT WORK;

OPEN nomatch\_cursor;

LOOP

FETCH nomatch\_cursor

INTO nomatch\_row;

EXIT WHEN nomatch\_cursor%NOTFOUND;

insert into ng\_reports.btod\_ntls\_audit\_wrk (ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, ban\_custom\_field\_4,

stripped\_ec\_circuit\_id, inventory\_vendor\_code, charge\_amount, bill\_date\_month, bill\_date\_year,

manual\_entry, btod\_extract\_date, bill\_date, ban\_custom\_field\_3, serv\_estab\_date, ntls\_territory, ntls\_market,

ntls\_sub\_market, ntls\_extract\_date, btod\_ntls\_match, uniquely\_identified, match\_code, match\_status)

values (nomatch\_row.ban\_category\_1, nomatch\_row.ban\_category\_2, nomatch\_row.ban\_custom\_field\_1, nomatch\_row.master\_ban, nomatch\_row.ban,

nomatch\_row.ban\_custom\_field\_4, nomatch\_row.stripped\_ec\_circuit\_id, nomatch\_row.inventory\_vendor\_code, nomatch\_row.btod\_amt,

nomatch\_row.bill\_date\_month, nomatch\_row.bill\_date\_year, nomatch\_row.manual\_entry, nomatch\_row.btod\_extract\_date,

nomatch\_row.btod\_bd, nomatch\_row.ban\_custom\_field\_3, nomatch\_row.serv\_estab\_date, nomatch\_row.ntls\_territory, nomatch\_row.ntls\_market,

nomatch\_row.ntls\_sub\_market, trunc(sysdate), 'N', 'N','BTOD Only', 'No Matching Circuit in Granite');

update ng\_reports.btod\_audit\_data\_process

set processed = 'N'

where seqno = nomatch\_row.seqno;

v\_cnt := v\_cnt + 1;

if v\_cnt > 5000 then

commit;

v\_cnt := 0;

end if;

END LOOP;

CLOSE nomatch\_cursor;

COMMIT WORK;

--multiple circ/vendor/ban

OPEN multi\_cursor;

LOOP

FETCH multi\_cursor

INTO multi\_row;

EXIT WHEN multi\_cursor%NOTFOUND;

insert into ng\_reports.btod\_ntls\_audit\_wrk (ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, ban\_custom\_field\_4,

stripped\_ec\_circuit\_id, inventory\_vendor\_code, charge\_amount, bill\_date\_month, bill\_date\_year,

manual\_entry, btod\_extract\_date, bill\_date, ban\_custom\_field\_3, serv\_estab\_date, ntls\_territory, ntls\_market,

ntls\_sub\_market, ntls\_segment\_id, ntls\_segment\_name, ntls\_type, ntls\_vendor, ntls\_bandwidth, ntls\_status,

ntls\_curr\_path\_inst\_id, ntls\_ban, ntls\_next\_path\_inst\_id,

ntls\_extract\_date, btod\_ntls\_match, uniquely\_identified, match\_code, match\_status)

values (multi\_row.ban\_category\_1, multi\_row.ban\_category\_2, multi\_row.ban\_custom\_field\_1, multi\_row.master\_ban, multi\_row.ban,

multi\_row.ban\_custom\_field\_4, multi\_row.stripped\_ec\_circuit\_id, multi\_row.inventory\_vendor\_code, multi\_row.btod\_amt,

multi\_row.bill\_date\_month, multi\_row.bill\_date\_year, multi\_row.manual\_entry, multi\_row.btod\_extract\_date,

multi\_row.btod\_bd, multi\_row.ban\_custom\_field\_3, multi\_row.serv\_estab\_date, multi\_row.ntls\_territory, multi\_row.ntls\_market,

multi\_row.ntls\_sub\_market, multi\_row.ntls\_segment\_id, multi\_row.ntls\_segment\_name, multi\_row.ntls\_type, multi\_row.ntls\_vendor,

multi\_row.ntls\_bandwidth, multi\_row.ntls\_status, multi\_row.ntls\_curr\_path\_inst\_id, multi\_row.ntls\_ban, multi\_row.ntls\_next\_path\_inst\_id,

trunc(sysdate), 'Y', 'N','BTOD Only', 'Multiple Circuits Match in Granite');

update ng\_reports.btod\_audit\_data\_process

set processed = 'M'

where seqno = multi\_row.seqno;

v\_cnt := v\_cnt + 1;

if v\_cnt > 5000 then

commit;

v\_cnt := 0;

end if;

END LOOP;

CLOSE multi\_cursor;

COMMIT WORK;

--multiple circ/vendor

OPEN multi1\_cursor;

LOOP

FETCH multi1\_cursor

INTO multi1\_row;

EXIT WHEN multi1\_cursor%NOTFOUND;

insert into ng\_reports.btod\_ntls\_audit\_wrk (ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, ban\_custom\_field\_4,

stripped\_ec\_circuit\_id, inventory\_vendor\_code, charge\_amount, bill\_date\_month, bill\_date\_year,

manual\_entry, btod\_extract\_date, bill\_date, ban\_custom\_field\_3, serv\_estab\_date, ntls\_territory, ntls\_market,

ntls\_sub\_market, ntls\_segment\_id, ntls\_segment\_name, ntls\_type, ntls\_vendor, ntls\_bandwidth, ntls\_status,

ntls\_curr\_path\_inst\_id, ntls\_ban, ntls\_next\_path\_inst\_id,

ntls\_extract\_date, btod\_ntls\_match, uniquely\_identified, match\_code, match\_status)

values (multi1\_row.ban\_category\_1, multi1\_row.ban\_category\_2, multi1\_row.ban\_custom\_field\_1, multi1\_row.master\_ban, multi1\_row.ban,

multi1\_row.ban\_custom\_field\_4, multi1\_row.stripped\_ec\_circuit\_id, multi1\_row.inventory\_vendor\_code, multi1\_row.btod\_amt,

multi1\_row.bill\_date\_month, multi1\_row.bill\_date\_year, multi1\_row.manual\_entry, multi1\_row.btod\_extract\_date,

multi1\_row.btod\_bd, multi1\_row.ban\_custom\_field\_3, multi1\_row.serv\_estab\_date, multi1\_row.ntls\_territory, multi1\_row.ntls\_market,

multi1\_row.ntls\_sub\_market, multi1\_row.ntls\_segment\_id, multi1\_row.ntls\_segment\_name, multi1\_row.ntls\_type, multi1\_row.ntls\_vendor,

multi1\_row.ntls\_bandwidth, multi1\_row.ntls\_status, multi1\_row.ntls\_curr\_path\_inst\_id, multi1\_row.ntls\_ban, multi1\_row.ntls\_next\_path\_inst\_id,

trunc(sysdate), 'Y', 'N','BTOD Only', 'Multiple Circuits Match in Granite');

update ng\_reports.btod\_audit\_data\_process

set processed = 'M'

where seqno = multi1\_row.seqno;

v\_cnt := v\_cnt + 1;

if v\_cnt > 5000 then

commit;

v\_cnt := 0;

end if;

END LOOP;

CLOSE multi1\_cursor;

COMMIT WORK;

--mulitple circ

OPEN multi2\_cursor;

LOOP

FETCH multi2\_cursor

INTO multi2\_row;

EXIT WHEN multi2\_cursor%NOTFOUND;

insert into ng\_reports.btod\_ntls\_audit\_wrk (ban\_category\_1, ban\_category\_2, ban\_custom\_field\_1, master\_ban, ban, ban\_custom\_field\_4,

stripped\_ec\_circuit\_id, inventory\_vendor\_code, charge\_amount, bill\_date\_month, bill\_date\_year,

manual\_entry, btod\_extract\_date, bill\_date, ban\_custom\_field\_3, serv\_estab\_date, ntls\_territory, ntls\_market,

ntls\_sub\_market, ntls\_segment\_id, ntls\_segment\_name, ntls\_type, ntls\_vendor, ntls\_bandwidth, ntls\_status,

ntls\_curr\_path\_inst\_id, ntls\_ban, ntls\_next\_path\_inst\_id,

ntls\_extract\_date, btod\_ntls\_match, uniquely\_identified, match\_code, match\_status)

values (multi2\_row.ban\_category\_1, multi2\_row.ban\_category\_2, multi2\_row.ban\_custom\_field\_1, multi2\_row.master\_ban, multi2\_row.ban,

multi2\_row.ban\_custom\_field\_4, multi2\_row.stripped\_ec\_circuit\_id, multi2\_row.inventory\_vendor\_code, multi2\_row.btod\_amt,

multi2\_row.bill\_date\_month, multi2\_row.bill\_date\_year, multi2\_row.manual\_entry, multi2\_row.btod\_extract\_date,

multi2\_row.btod\_bd, multi2\_row.ban\_custom\_field\_3, multi2\_row.serv\_estab\_date, multi2\_row.ntls\_territory, multi2\_row.ntls\_market,

multi2\_row.ntls\_sub\_market, multi2\_row.ntls\_segment\_id, multi2\_row.ntls\_segment\_name, multi2\_row.ntls\_type, multi2\_row.ntls\_vendor,

multi2\_row.ntls\_bandwidth, multi2\_row.ntls\_status, multi2\_row.ntls\_curr\_path\_inst\_id, multi2\_row.ntls\_ban, multi2\_row.ntls\_next\_path\_inst\_id,

trunc(sysdate), 'Y', 'N','BTOD Only', 'Multiple Circuits Match in Granite');

update ng\_reports.btod\_audit\_data\_process

set processed = 'M'

where seqno = multi2\_row.seqno;

v\_cnt := v\_cnt + 1;

if v\_cnt > 5000 then

commit;

v\_cnt := 0;

end if;

END LOOP;

CLOSE multi2\_cursor;

COMMIT WORK;

execute immediate 'truncate table ng\_reports.btod\_ntls\_audit reuse storage'; commit;

insert into ng\_reports.btod\_ntls\_audit (select \* from ng\_reports.btod\_ntls\_audit\_wrk); commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put(SQLCODE);

dbms\_output.put(SQLERRM);

dbms\_output.put\_line(SubStr('Error: btod\_ntls\_audit\_detail(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

--raise;

END load\_btod\_ntls\_audit\_detail;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Loads BTOD NTLS Audit Summary

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Procedure load\_btod\_ntls\_audit\_summary is

BEGIN

execute immediate 'truncate table ng\_reports.btod\_ntls\_audit\_summary reuse storage'; commit;

insert into ng\_reports.btod\_ntls\_audit\_summary

select ntls\_territory,

case when ntls\_territory = 'NNO' then 'NNO' else ntls\_market end ntls\_market,

case when ntls\_territory = 'NNO' then 'NNO' else ntls\_sub\_market end ntls\_sub\_market,

ckt\_count, ckts\_matched, ckts\_excp,

ckts\_matched\_pct\_str, ckts\_matched\_pct\_num, sysdate,

case

when ntls\_sub\_market is null and ntls\_market is not null then ntls\_market

when ntls\_sub\_market is null and ntls\_market is null then ntls\_territory

else ntls\_sub\_market

end territory\_market\_sub

from (select ntls\_territory, ntls\_market, ntls\_sub\_market, ckt\_count, ckts\_matched, ckts\_excp,

ckts\_matched\_pct\_str, ckts\_matched\_pct\_num

from (select ntls\_territory, ntls\_market, ntls\_sub\_market, sum(tot\_cnt) as ckt\_count, sum(match\_cnt) as ckts\_matched, sum(excp\_cnt) as ckts\_excp,

to\_char((sum(match\_cnt)/sum(tot\_cnt) \* 100),'999.99') as ckts\_matched\_pct\_str,

round((sum(match\_cnt)/sum(tot\_cnt) \* 100), 2) as ckts\_matched\_pct\_num

from (select ntls\_territory, ntls\_market, ntls\_sub\_market, count(\*) tot\_cnt, sum(match\_cnt) match\_cnt, sum(excp\_cnt) excp\_cnt

from (select nvl(drr.territory, 'Unknown') as ntls\_territory, nvl(drr.market\_territory, 'Unknown') as ntls\_market,

nvl(drr.sub\_market, 'Unknown') as ntls\_sub\_market,

case when processed = 'B' then 1 else 0 end match\_cnt,

case when processed <> 'B' then 1 else 0 end excp\_cnt

from (select distinct ban\_category\_1, stripped\_ec\_circuit\_id, inventory\_vendor\_code,

ban, processed

from ng\_reports.btod\_audit\_data\_process) bdp,

ng\_reports.btod\_ntls\_domain\_map bxdm,

ng\_reports.domains\_regional\_reporting drr

where bdp.ban\_category\_1 = bxdm.btod\_ban\_category\_1 (+)

and bxdm.ntls\_domain\_id = drr.domain\_id (+))

group by ntls\_territory, ntls\_market, ntls\_sub\_market)

group by rollup (ntls\_territory, ntls\_market, ntls\_sub\_market)

order by ntls\_territory, ntls\_market, ntls\_sub\_market))

where ntls\_market != 'NNO' or ntls\_market is null;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put(SQLCODE);

dbms\_output.put(SQLERRM);

dbms\_output.put\_line(SubStr('Error: btod\_ntls\_audit\_summary(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

--raise;

END load\_btod\_ntls\_audit\_summary;

END;

/

--------------------------------------------------------

-- DDL for Package Body CIENA\_6500\_DISCOVERY

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."CIENA\_6500\_DISCOVERY" AS

/\*\*===========================================================================================\*

\* @procedure : main

\* @description : Entry point for the package which runs complete audit for Ciena6500 device.

\* @author : Latha Ramamurthy

\* @date : April 18 2016

\*===========================================================================================\*

\* Change History

\*===========================================================================================\*

\* @changeId :

\* @description :

\* @author :

\* @date :

\*===========================================================================================\*

\*\*===========================================================================================\*/

PROCEDURE CLEANUP\_PARSED\_DATA is

sqlStmt VARCHAR2(500);

BEGIN

dbms\_output.put\_line('Executing procedure: CLEANUP\_PARSED\_DATA');

sqlStmt := 'update NG\_REPORTS.ciena\_6500\_ne\_parsed\_data\_wk p set p.resp\_field\_1 = replace(p.resp\_field\_1,''\"'','''') ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update NG\_REPORTS.ciena\_6500\_ne\_parsed\_data\_wk p set p.resp\_field\_2 = replace(p.resp\_field\_2,''\"'','''') ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update NG\_REPORTS.ciena\_6500\_ne\_parsed\_data\_wk p set p.resp\_field\_3 = replace(p.resp\_field\_3,''\"'','''') ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update NG\_REPORTS.ciena\_6500\_ne\_parsed\_data\_wk p set p.resp\_field\_4 = replace(p.resp\_field\_4,''\"'','''') ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update NG\_REPORTS.ciena\_6500\_ne\_key\_value\_wk p set p.value = replace(p.value,''\"'','''') ';

EXECUTE IMMEDIATE sqlStmt;

commit;

dbms\_output.put\_line('Ending CleanUp Parsed Data' );

EXCEPTION

WHEN OTHERS

THEN

dbms\_output.put\_line(SubStr('Error in Cleanup\_Parsed\_data. Error Code: '|| TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END CLEANUP\_PARSED\_DATA;

PROCEDURE DELETE\_WK\_TABLES is

sqlStmt VARCHAR2(500);

BEGIN

dbms\_output.put\_line('Executing procedure: DELETE\_WK\_TABLES');

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_key\_value\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_parsed\_data\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_errors\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_equip\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

dbms\_output.put\_line('Ending Delete\_wk\_Tables' );

EXCEPTION

WHEN OTHERS

THEN

dbms\_output.put\_line(SubStr('Error in Delet\_wk\_Tables. Error Code: '|| TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END DELETE\_WK\_TABLES;

PROCEDURE DELETE\_STALE\_DATA\_PRD\_TABLES is

sqlStmt VARCHAR2(1000);

BEGIN

dbms\_output.put\_line('Executing procedure: DELETE\_STALE\_DATA\_PRD\_TABLES');

sqlStmt := ' delete from NG\_REPORTS.ciena\_6500\_ne\_key\_value kv where kv.parsed\_inst\_id in

( select parsed\_inst\_id

from NG\_REPORTS.ciena\_6500\_ne\_key\_value kv2,

NG\_REPORTS.ciena\_6500\_ne\_parsed\_data p , NG\_REPORTS.CIENA\_6500\_NE\_EQUIP eq

where EQ.ID = P.EQ\_ID

and P.INST\_ID = kv2.parsed\_inst\_id

and EQ.DEVICE\_NAME in (

select distinct e.device\_name from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP e where extract\_date < sysdate - ' || CUT\_OFF ||

' ))

';

EXECUTE IMMEDIATE sqlStmt; --using CUT\_OFF;

commit;

sqlStmt := ' delete from NG\_REPORTS.ciena\_6500\_ne\_parsed\_data p where p.eq\_id in

(

select distinct e.id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP e where extract\_date < sysdate - ' || CUT\_OFF ||

' ) ';

EXECUTE IMMEDIATE sqlStmt; --using CUT\_OFF;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_errors pd where pd.eq\_id in

(

select distinct e.id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP e where extract\_date < sysdate - ' || CUT\_OFF ||

' ) ';

EXECUTE IMMEDIATE sqlStmt; --using CUT\_OFF;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_equip where id in

(

select distinct e.id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP e where extract\_date < sysdate - ' || CUT\_OFF ||

' ) ';

EXECUTE IMMEDIATE sqlStmt; --using CUT\_OFF;

commit;

dbms\_output.put\_line('Ending DELETE\_STALE\_DATA\_PRD\_TABLES' );

EXCEPTION

WHEN OTHERS

THEN

dbms\_output.put\_line(SubStr('Error in Delet\_wk\_Tables. Error Code: '|| TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END DELETE\_STALE\_DATA\_PRD\_TABLES;

PROCEDURE DELETE\_DEVICES\_NOT\_IN\_TIDLIST is

sqlStmt VARCHAR2(1000);

BEGIN

dbms\_output.put\_line('Executing procedure: DELETE\_DEVICES\_NOT\_IN\_TIDLIST');

sqlStmt := ' delete from NG\_REPORTS.ciena\_6500\_ne\_key\_value kv where kv.parsed\_inst\_id in

( select parsed\_inst\_id

from NG\_REPORTS.ciena\_6500\_ne\_key\_value kv2,

NG\_REPORTS.ciena\_6500\_ne\_parsed\_data p , NG\_REPORTS.CIENA\_6500\_NE\_EQUIP eq

where EQ.ID = P.EQ\_ID

and P.INST\_ID = kv2.parsed\_inst\_id

and EQ.DEVICE\_NAME not in (

select distinct device\_name from ciena\_6500\_ne\_tid\_list ))

';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := ' delete from NG\_REPORTS.ciena\_6500\_ne\_parsed\_data p where p.eq\_id in

(

select distinct e.id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP e where e.device\_name

not in (select distinct device\_name from ciena\_6500\_ne\_tid\_list)

)

' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_errors pd where pd.eq\_id in

(

select distinct e.id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP e where e.device\_name

not in (select distinct device\_name from ciena\_6500\_ne\_tid\_list)

)

' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_equip where device\_name not in (select distinct device\_name from ciena\_6500\_ne\_tid\_list) ' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

dbms\_output.put\_line('Ending DELETE\_DEVICES\_NOT\_IN\_TIDLIST' );

EXCEPTION

WHEN OTHERS

THEN

dbms\_output.put\_line(SubStr('Error in Delet\_wk\_Tables. Error Code: '|| TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END DELETE\_DEVICES\_NOT\_IN\_TIDLIST;

PROCEDURE COPY\_WK\_TO\_PRD is

sqlStmt VARCHAR2(2000);

BEGIN

dbms\_output.put\_line('Executing procedure: COPY\_WK\_TO\_PRD');

EXECUTE immediate 'truncate table ciena\_6500\_ne\_equip\_save';

--save the devices that are in ne\_equip but not in ne\_equip\_wk

sqlStmt := ' insert into NG\_REPORTS.ciena\_6500\_ne\_equip\_save

with keep\_devices as

(select distinct E.DEVICE\_NAME

from NG\_REPORTS.ciena\_6500\_ne\_equip e

minus

select distinct E.DEVICE\_NAME

from NG\_REPORTS.ciena\_6500\_ne\_equip\_wk e

)select e.\* from keep\_devices k, NG\_REPORTS.ciena\_6500\_ne\_equip e

where e.device\_name=k.device\_name ';

EXECUTE IMMEDIATE sqlStmt;

commit;

-- delete data in prd tables that are not in the list to be saved

sqlStmt := ' delete from NG\_REPORTS.ciena\_6500\_ne\_key\_value kv where kv.parsed\_inst\_id in

( select parsed\_inst\_id

from NG\_REPORTS.ciena\_6500\_ne\_key\_value kv2,

NG\_REPORTS.ciena\_6500\_ne\_parsed\_data p , NG\_REPORTS.CIENA\_6500\_NE\_EQUIP eq

where EQ.ID = P.EQ\_ID

and P.INST\_ID = kv2.parsed\_inst\_id

and EQ.id in (

select distinct id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP minus select distinct id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP\_SAVE

))';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := ' delete from NG\_REPORTS.ciena\_6500\_ne\_parsed\_data p where p.eq\_id in

(

select distinct id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP minus select distinct id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP\_SAVE

) ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_errors pd where pd.eq\_id in

(

select distinct id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP minus select distinct id from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP\_SAVE

) ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'delete from NG\_REPORTS.ciena\_6500\_ne\_equip where device\_name in

(

select distinct device\_name from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP minus select distinct device\_name from NG\_REPORTS.CIENA\_6500\_NE\_EQUIP\_SAVE

) ';

EXECUTE IMMEDIATE sqlStmt;

commit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

sqlStmt := 'insert into NG\_REPORTS.ciena\_6500\_ne\_equip select \* from NG\_REPORTS.ciena\_6500\_ne\_equip\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'insert into NG\_REPORTS.ciena\_6500\_ne\_errors select \* from NG\_REPORTS.ciena\_6500\_ne\_errors\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'insert into NG\_REPORTS.ciena\_6500\_ne\_parsed\_data select \* from NG\_REPORTS.ciena\_6500\_ne\_parsed\_data\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'insert into NG\_REPORTS.ciena\_6500\_ne\_key\_value select \* from NG\_REPORTS.ciena\_6500\_ne\_key\_value\_wk ';

EXECUTE IMMEDIATE sqlStmt;

commit;

dbms\_output.put\_line('Ending COPY\_WK\_TO\_PRD' );

EXCEPTION

WHEN OTHERS

THEN

dbms\_output.put\_line(SubStr('Error in Delet\_wk\_Tables. Error Code: '|| TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END COPY\_WK\_TO\_PRD;

END CIENA\_6500\_DISCOVERY;

/

--------------------------------------------------------

-- DDL for Package Body CIRC\_SEG\_NEW\_DECOM\_RPT\_PKG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_RPT\_PKG" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: CIRC\_SEG\_NEW\_DECOM\_RPT\_PKG

PURPOSE:

REVISIONS:

Ver Date Author Description

------ ---------- ---------- ------------------------------------

1.0 7/15/2013 c0hedav 1. Created this package body.

1.1 7/19/2013 c0hedav 2. add 6 new fields per request into table CIRC\_SEG\_NEW\_DECOM\_DETAIL :

SEGMENT\_TYPE, CHG\_BY, DISCONNECT\_PON, DISCONNECT\_ORDER\_NUM, DISCONNECT\_ORDER\_TS, DISCONNECT\_TS

1.2 8/5/2013 c0hedav 3. add BILING\_CODE,

1.3 8/7/2013 c0hedav 4. exclude duplicated audit log for both NEW and DECOM

1.4 6/8/2016 ervist1 5. Convert to Territory/Market/SubMarket

1.5 12/20/2017 araje8y 6. Migrate report package into Nautilus

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE RUN\_DAILY\_DETAIL\_RPT

IS

v\_id NUMBER;

v\_cutoff\_time\_last DATE;

v\_cutoff\_time\_new DATE;

v\_group\_name VARCHAR2 (30);

BEGIN

SELECT NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN\_SEQ.NEXTVAL

INTO v\_id

FROM DUAL;

BEGIN

SELECT MAX (PROC\_START\_TS) INTO v\_cutoff\_time\_last FROM CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN;

EXCEPTION WHEN NO\_DATA\_FOUND THEN SELECT TO\_DATE ('01-01-1970', 'MM-DD-YYYY') INTO v\_cutoff\_time\_last FROM DUAL;

END;

SELECT SYSDATE INTO v\_cutoff\_time\_new FROM DUAL;

INSERT INTO CIRC\_SEG\_NEW\_DECOM\_DETAIL (RPT\_ID,

DOMAIN\_ID, --domain\_id / NG\_TOPOLOGY.SEGMENT\_DOMAIN\_MAP

DOMAIN\_NAME, --domain\_name / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

TERRITORY, --territory / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

MARKET\_TERRITORY, --market\_territory / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

SUB\_MARKET, --sub\_market / NG\_REPORTS.DOMAINS\_LEAF\_REPORTING

SEGMENT\_ID, --segment\_id / NG\_TOPOLOGY.TRAIL\_SEGMENT

SEGMENT\_NAME, --circ\_hum\_id / XNG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DET\_TESS\_VW = segment\_id / NG\_TOPOLOGY.TRAIL\_SEGMENT

SEGMENT\_TYPE, --type / NG\_TOPOLOGY.TRAIL\_SEGMENT

Z\_SITE\_ID, --z\_site\_id / NG\_TOPOLOGY.TRAIL\_SEGMENT

SITE\_NAME, --site\_hum\_id = site\_name / NG\_REPORTS.SITE - ng\_inventory.SITE

VENDOR, --vendor / NG\_TOPOLOGY.TRAIL\_SEGMENT

BILLING\_CODE, --billing\_code / NG\_TOPOLOGY.TRAIL\_SEGMENT

STATUS, --status / NG\_TOPOLOGY.TRAIL\_SEGMENT

ACTION\_TYPE, --part of the insert fn

COMMIT\_DATE, --commit\_date / NG\_TOPOLOGY.JV\_COMMIT

AUTHOR, --author / NG\_TOPOLOGY.JV\_COMMIT

BANDWIDTH, --bandwidth / NG\_TOPOLOGY.TRAIL\_SEGMENT

MRC --recur\_costs / VZWNET.CIRC\_INST = recur\_costs / NG\_TOPOLOGY.TRAIL\_SEGMENT

)

WITH INST\_AUDIT\_UPDATED AS (

SELECT

g.local\_id,

cc.qualified\_name,

c.commit\_date,

c.author,

s.type,

s.version

FROM

ng\_topology.jv\_global\_id g,

ng\_topology.jv\_cdo\_class cc,

ng\_topology.jv\_commit c,

ng\_topology.jv\_snapshot s

WHERE cc.cdo\_class\_pk = g.cdo\_class\_fk

AND g.global\_id\_pk = s.global\_id\_fk

AND s.commit\_fk = c.commit\_pk

AND c.commit\_date >= v\_cutoff\_time\_last

AND c.commit\_date < v\_cutoff\_time\_new

AND cc.qualified\_name = 'TrailSegment'

AND s.type = 'UPDATE'),

INST\_UPDATE\_LOG\_DECOM AS (

SELECT

g.local\_id,

cc.qualified\_name,

--s.changed\_properties,

s.version,

s.state

FROM

ng\_topology.jv\_global\_id g,

ng\_topology.jv\_cdo\_class cc,

ng\_topology.jv\_snapshot s

WHERE

cc.cdo\_class\_pk = g.cdo\_class\_fk

AND g.global\_id\_pk = s.global\_id\_fk

--AND s.changed\_properties LIKE '%"status"%'

--AND REGEXP\_LIKE (s.state, '"status": "Decommissioned"', 'i')),

AND REGEXP\_LIKE (state, '"FIELD\_CHANGED": "Status".\*"NEW\_VALUE": "Decommissioned"', 'i')),

AUDIT\_DECOM\_LOG AS (

SELECT

a.local\_id,

a.commit\_date,

a.author

FROM

INST\_AUDIT\_UPDATED a,

INST\_UPDATE\_LOG\_DECOM u

WHERE

u.local\_id = a.local\_id

AND u.qualified\_name = a.qualified\_name

AND u.version = a.version),

AUDIT\_DECOM AS (

SELECT

a.local\_id,

a.commit\_date,

a.author

FROM AUDIT\_DECOM\_LOG a

WHERE a.commit\_date = (

SELECT MAX (a.commit\_date)

FROM AUDIT\_DECOM\_LOG b

WHERE a.local\_id = b.local\_id)),

AUDIT\_INSERT AS (

SELECT

g.local\_id,

c.commit\_date,

c.author

FROM

ng\_topology.jv\_global\_id g,

ng\_topology.jv\_commit c,

ng\_topology.jv\_cdo\_class cc,

ng\_topology.jv\_snapshot s

WHERE

c.commit\_date >= v\_cutoff\_time\_last

AND c.commit\_date < v\_cutoff\_time\_new

AND cc.qualified\_name = 'TrailSegment'

AND s.type = 'INITIAL'),

AUDIT\_INSERT\_MAX AS (

SELECT

a.local\_id,

a.commit\_date,

a.author

FROM AUDIT\_INSERT a

WHERE a.commit\_date = (

SELECT MAX (a.commit\_date)

FROM AUDIT\_INSERT b

WHERE a.local\_id = b.local\_id)),

AUDIT\_INSERT\_DECOM AS (

SELECT

local\_id,

commit\_date,

author,

'NEW' AS action

FROM AUDIT\_INSERT\_MAX

UNION

SELECT

local\_id,

commit\_date,

author,

'DECOM' AS action

FROM AUDIT\_DECOM),

CIRC\_INST\_ALL AS (

SELECT

segment\_id,

type AS segment\_type,

name,

z\_site\_id,

vendor,

billing\_code,

status,

bandwidth,

recur\_costs

FROM ng\_topology.AR\_TRAIL\_SEGMENT

UNION

SELECT

segment\_id,

type AS segment\_type,

name,

z\_site\_id,

vendor,

billing\_code,

status,

bandwidth,

recur\_costs

FROM ng\_topology.TRAIL\_SEGMENT)

SELECT

v\_id,

rpt.domain\_id,

RPT.DOMAIN\_NAME,

RPT.TERRITORY,

RPT.MARKET\_TERRITORY,

RPT.SUB\_MARKET,

circ.segment\_id,

circ.name,

circ.segment\_type,

circ.z\_site\_id,

z.site\_name,

circ.vendor,

circ.billing\_code,

circ.status,

aud.action,

aud.commit\_date,

aud.author,

circ.bandwidth,

circ.recur\_costs

FROM

AUDIT\_INSERT\_DECOM aud,

CIRC\_INST\_ALL circ,

NG\_TOPOLOGY.SEGMENT\_DOMAIN\_MAP dm,

NG\_REPORTS.DOMAINS\_LEAF\_REPORTING rpt,

NG\_REPORTS.SITE z -- site is for detail only. ng\_inventory.SITE

WHERE

aud.local\_id = circ.segment\_id

AND circ.segment\_id = dm.segment\_id

AND dm.domain\_id = rpt.domain\_id

AND circ.z\_site\_id = z.site\_reference\_id;

-- Update 4 fields related to Disconnect

v\_group\_name := 'Disconnect Information';

DELETE FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL a

WHERE a.commit\_date <> (SELECT MAX (commit\_date) FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL b WHERE a.domain\_id = b.domain\_id AND a.segment\_id = b.segment\_id AND a.action\_type = b.action\_type);

UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

SET DISCONNECT\_PON = (SELECT ATTRIBUTE\_VALUE FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID AND sa.GROUP\_NAME = v\_group\_name AND sa.ATTRIBUTE\_NAME = 'Disconnect PON')

WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

SET DISCONNECT\_ORDER\_TS =

(SELECT ATTRIBUTE\_VALUE

FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa

WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID AND sa.GROUP\_NAME = v\_group\_name AND sa.ATTRIBUTE\_NAME = 'Disconnect Order Date')

WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

SET DISCONNECT\_ORDER\_NUM =

(SELECT ATTRIBUTE\_VALUE

FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa

WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID AND sa.GROUP\_NAME = v\_group\_name AND sa.ATTRIBUTE\_NAME = 'Disconnect Order Number')

WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

UPDATE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DETAIL c

SET DISCONNECT\_TS =

(SELECT ATTRIBUTE\_VALUE

FROM NG\_TOPOLOGY.SEGMENT\_ATTRIBUTES sa

WHERE c.SEGMENT\_ID = sa.SEGMENT\_ID AND sa.GROUP\_NAME = v\_group\_name AND sa.ATTRIBUTE\_NAME = 'Disconnect Date')

WHERE COMMIT\_DATE >= v\_cutoff\_time\_last AND COMMIT\_DATE < v\_cutoff\_time\_new;

UPDATE CIRC\_SEG\_NEW\_DECOM\_DETAIL D

SET D.SEGMENT\_NAME = (SELECT TS.NAME FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID) WHERE (D.SEGMENT\_NAME IS NULL) OR D.SEGMENT\_NAME != (SELECT TS.NAME

FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID);

UPDATE CIRC\_SEG\_NEW\_DECOM\_DETAIL D

SET D.MRC =

(SELECT TS.RECUR\_COSTS

FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS

WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID)

WHERE (D.MRC IS NULL)

OR D.MRC != (SELECT TS.RECUR\_COSTS

FROM NG\_TOPOLOGY.TRAIL\_SEGMENT TS

WHERE D.SEGMENT\_ID = TS.SEGMENT\_ID);

RUN\_ALL\_MONTHLY\_WEEKLY\_RPT (); --as we deleted some duplicated, so need recalculate all count in summary

--RUN\_ONE\_MONTHLY\_RPT (ADD\_MONTHS (v\_cutoff\_time\_new, 1)); --current month=next mon's previous mon

--RUN\_ONE\_WEEKLY\_RPT (v\_cutoff\_time\_new + 7); --current week=next week's previous week

INSERT

INTO CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN (RPT\_ID, PROC\_START\_TS, PROC\_END\_TS)

VALUES (v\_id, v\_cutoff\_time\_new, SYSDATE);

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

ROLLBACK;

END;

-------------------------------------------------------------

PROCEDURE RUN\_ONE\_MONTHLY\_RPT (p\_run\_date DATE)

IS

v\_first\_day\_this\_month DATE;

v\_first\_day\_last\_month DATE;

v\_last\_month VARCHAR2 (20);

v\_run\_date DATE;

BEGIN

v\_run\_date := p\_run\_date;

IF (v\_run\_date IS NULL)

THEN

SELECT SYSDATE INTO v\_run\_date FROM DUAL;

END IF;

--first day of the month contains v\_run\_date

SELECT TO\_DATE (TO\_CHAR (v\_run\_date, 'yyyymm'), 'yyyymm')

INTO v\_first\_day\_this\_month --at 00hr:00mm:00sec

FROM DUAL;

--first day of the previous month

SELECT TO\_DATE (TO\_CHAR (v\_first\_day\_this\_month - 1, 'yyyymm'),

'yyyymm')

INTO v\_first\_day\_last\_month --at 00hr:00mm:00sec

FROM DUAL;

SELECT TO\_CHAR (v\_first\_day\_last\_month, 'Mon, yyyy')

INTO v\_last\_month

FROM DUAL;

DELETE FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_SUMMARY

WHERE PERIOD = v\_last\_month AND PERIOD\_TYPE = 'MONTHLY';

INSERT INTO ng\_reports.circ\_seg\_new\_decom\_summary (

period,

period\_type,

territory,

market\_territory,

sub\_market,

count\_decom,

count\_new,

count\_new\_total,

count\_new\_by\_tess,

count\_new\_not\_tess\_contract,

count\_new\_not\_tess\_no\_contract,

percentage\_new\_by\_tess

)

WITH monthly AS (

SELECT

\*

FROM

ng\_reports.circ\_seg\_new\_decom\_det\_tess\_vw

WHERE

commit\_date >= v\_first\_day\_last\_month

AND commit\_date < v\_first\_day\_this\_month

),max\_commit\_date AS (

SELECT

\*

FROM

monthly a

WHERE

a.commit\_date = (

SELECT

MAX(commit\_date)

FROM

monthly b

WHERE

a.segment\_id = b.segment\_id

AND a.action\_type = b.action\_type

)

) SELECT

v\_last\_month,

'MONTHLY',

territory,

market\_territory,

sub\_market,

SUM(decom) decom\_count,

SUM(new) count\_new,

SUM(new\_total) count\_new\_total,

SUM(new\_by\_tess) count\_new\_by\_tess,

SUM(new\_not\_tess\_contract) count\_new\_not\_tess\_contract,

SUM(new\_not\_tess\_no\_contract) count\_new\_not\_tess\_no\_contract,

round(

CASE

WHEN SUM(new\_total) > 0 THEN SUM(new\_by\_tess) \* 100.0 / SUM(new\_total)

ELSE 0.00

END,2) percentage\_new\_by\_tess

FROM

max\_commit\_date

GROUP BY

territory,

market\_territory,

sub\_market

ORDER BY

territory,

market\_territory,

sub\_market;

end;

------------------------------------------------------------------------------

PROCEDURE RUN\_ONE\_WEEKLY\_RPT (p\_run\_date DATE)

IS

v\_first\_day\_this\_week DATE;

v\_last\_day\_last\_week DATE;

v\_first\_day\_last\_week DATE;

v\_run\_date DATE;

v\_period VARCHAR2 (50);

BEGIN

v\_run\_date := p\_run\_date;

IF (v\_run\_date IS NULL)

THEN

SELECT SYSDATE INTO v\_run\_date FROM DUAL;

END IF;

--Sunday after v\_run\_date

SELECT TRUNC (NEXT\_DAY (v\_run\_date - 7, 'SUN'))

INTO v\_first\_day\_this\_week

FROM DUAL;

--Saturday of the week contain v\_run\_date

v\_last\_day\_last\_week := v\_first\_day\_this\_week - 1;

--Sunday of the week contain v\_run\_date

SELECT TRUNC (NEXT\_DAY (v\_run\_date - 14, 'SUN'))

INTO v\_first\_day\_last\_week

FROM DUAL;

SELECT TO\_CHAR (v\_first\_day\_last\_week, 'mm/dd/yyyy')

|| ' - '

|| TO\_CHAR (v\_last\_day\_last\_week, 'mm/dd/yyyy')

INTO v\_period

FROM DUAL;

DELETE FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_SUMMARY

WHERE PERIOD = v\_period AND PERIOD\_TYPE = 'WEEKLY';

INSERT INTO ng\_reports.circ\_seg\_new\_decom\_summary (

period,

period\_type,

territory,

market\_territory,

sub\_market,

count\_decom,

count\_new,

count\_new\_total,

count\_new\_by\_tess,

count\_new\_not\_tess\_contract,

count\_new\_not\_tess\_no\_contract,

percentage\_new\_by\_tess

) WITH weekly AS (

SELECT

\*

FROM

ng\_reports.circ\_seg\_new\_decom\_det\_tess\_vw

WHERE

commit\_date >= v\_first\_day\_last\_week

AND commit\_date < v\_first\_day\_this\_week

),max\_commit\_date AS (

SELECT

\*

FROM

weekly a

WHERE

a.commit\_date = (

SELECT

MAX(commit\_date)

FROM

weekly b

WHERE

a.segment\_id = b.segment\_id

AND a.action\_type = b.action\_type

)

) SELECT

v\_period,

'WEEKLY',

territory,

market\_territory,

sub\_market,

SUM(decom) decom\_count,

SUM(new) count\_new,

SUM(new\_total) count\_new\_total,

SUM(new\_by\_tess) count\_new\_by\_tess,

SUM(new\_not\_tess\_contract) count\_new\_not\_tess\_contract,

SUM(new\_not\_tess\_no\_contract) count\_new\_not\_tess\_no\_contract,

round(

CASE

WHEN SUM(new\_total) > 0 THEN SUM(new\_by\_tess) \* 100.0 / SUM(new\_total)

ELSE 0.00

END,

2) percentage\_new\_by\_tess

FROM

max\_commit\_date

GROUP BY

territory,

market\_territory,

sub\_market

ORDER BY

territory,

market\_territory,

sub\_market;

end;

------------------------------------------------------------------------

PROCEDURE RUN\_ALL\_MONTHLY\_WEEKLY\_RPT

IS

v\_min\_time DATE;

v\_max\_time DATE;

v\_run\_time DATE;

v\_end\_time DATE;

BEGIN

SELECT MIN (COMMIT\_DATE) INTO v\_min\_time FROM CIRC\_SEG\_NEW\_DECOM\_DETAIL;

SELECT MAX (COMMIT\_DATE) INTO v\_max\_time FROM CIRC\_SEG\_NEW\_DECOM\_DETAIL;

--clean table

EXECUTE IMMEDIATE

'TRUNCATE TABLE NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_SUMMARY';

--run monthly for all

SELECT TO\_DATE (TO\_CHAR (v\_min\_time, 'yyyymm'), 'yyyymm')

INTO v\_run\_time -- 1st day of the month at 00hr:00mm:00sec for earliest record

FROM DUAL;

SELECT TO\_DATE (TO\_CHAR (v\_max\_time, 'yyyymm'), 'yyyymm')

INTO v\_end\_time -- 1st day of the month at 00hr:00mm:00sec for latest record

FROM DUAL;

SELECT ADD\_MONTHS (v\_end\_time, 1) INTO v\_end\_time FROM DUAL; --1st day of next month

/\*

IF (v\_end\_time > SYSDATE)

THEN

SELECT ADD\_MONTHS (v\_end\_time, -1) INTO v\_end\_time FROM DUAL; -- roll back to 1st day of this month, to avoid partial monthly report

END IF;

\*/

WHILE (v\_run\_time <= v\_end\_time)

LOOP

BEGIN

RUN\_ONE\_MONTHLY\_RPT (v\_run\_time);

SELECT ADD\_MONTHS (v\_run\_time, 1) INTO v\_run\_time FROM DUAL;

END;

END LOOP;

--run weekly for all

--Sunday after the earliest record

SELECT TRUNC (NEXT\_DAY (v\_min\_time, 'SUN')) INTO v\_run\_time FROM DUAL;

--Sunday after the latest record

SELECT TRUNC (NEXT\_DAY (v\_max\_time, 'SUN')) INTO v\_end\_time FROM DUAL;

/\*

IF (v\_end\_time > SYSDATE)

THEN -- last record is in this week which is in partial week

v\_end\_time := v\_end\_time - 7;

END IF;

\*/

WHILE (v\_run\_time <= v\_end\_time)

LOOP

BEGIN

RUN\_ONE\_WEEKLY\_RPT (v\_run\_time);

v\_run\_time := v\_run\_time + 7;

END;

END LOOP;

--run yearly for all

--first day(mid-night) of the year earlier than v\_min\_time

SELECT TRUNC(TO\_DATE ('01/01'||TO\_CHAR (v\_min\_time, 'yyyy'), 'mm/dd/yyyy'))

INTO v\_run\_time

FROM DUAL;

--first day(mid-night) of the next year after v\_max\_time

SELECT TRUNC(TO\_DATE ('12/31'||TO\_CHAR (v\_max\_time, 'yyyy'), 'mm/dd/yyyy')) + 1

INTO v\_end\_time

FROM DUAL;

WHILE (v\_run\_time <= v\_end\_time)

LOOP

BEGIN

RUN\_ONE\_YEARLY\_RPT (v\_run\_time);

-- add one year

SELECT TRUNC(TO\_DATE ('12/31'||TO\_CHAR (v\_run\_time, 'yyyy'), 'mm/dd/yyyy')) + 1

INTO v\_run\_time

FROM DUAL;

END;

END LOOP;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

ROLLBACK;

END;

----------------------------------------------------------------------

PROCEDURE RUN\_ONE\_YEARLY\_RPT (p\_run\_date DATE)

IS

v\_first\_day\_this\_year DATE;

v\_first\_day\_next\_year DATE;

v\_this\_year VARCHAR2 (20);

v\_run\_date DATE;

BEGIN

v\_run\_date := p\_run\_date;

IF (v\_run\_date IS NULL)

THEN

SELECT SYSDATE INTO v\_run\_date FROM DUAL;

END IF;

--first day(mid-night) of the year contains v\_run\_date

SELECT TRUNC(TO\_DATE ('01/01'||TO\_CHAR (v\_run\_date, 'yyyy'), 'mm/dd/yyyy'))

INTO v\_first\_day\_this\_year

FROM DUAL;

--this year

SELECT TO\_CHAR (v\_first\_day\_this\_year, 'yyyy')

INTO v\_this\_year

FROM DUAL;

--first day(mid-night) of the next year

SELECT TRUNC(TO\_DATE ('12/31'||v\_this\_year, 'mm/dd/yyyy')) + 1

INTO v\_first\_day\_next\_year

FROM DUAL;

DELETE FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_SUMMARY

WHERE PERIOD = v\_this\_year AND PERIOD\_TYPE = 'YEARLY';

INSERT INTO NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_SUMMARY (

PERIOD,

PERIOD\_TYPE,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

COUNT\_DECOM,

COUNT\_NEW,

COUNT\_NEW\_TOTAL,

COUNT\_NEW\_BY\_TESS,

COUNT\_NEW\_NOT\_TESS\_CONTRACT,

COUNT\_NEW\_NOT\_TESS\_NO\_CONTRACT,

PERCENTAGE\_NEW\_BY\_TESS)

WITH YEARLY AS (

SELECT \*

FROM NG\_REPORTS.CIRC\_SEG\_NEW\_DECOM\_DET\_TESS\_VW

WHERE

COMMIT\_DATE >= v\_first\_day\_this\_year

AND COMMIT\_DATE < v\_first\_day\_next\_year),

MAX\_COMMIT\_DATE AS (

SELECT \*

FROM YEARLY a

WHERE

a.COMMIT\_DATE = (

SELECT MAX (COMMIT\_DATE)

FROM YEARLY b

WHERE

a.SEGMENT\_ID = b.SEGMENT\_ID

AND a.ACTION\_TYPE = b.ACTION\_TYPE))

SELECT

v\_this\_year PERIOD,

'YEARLY' PERIOD\_TYPE,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

SUM (DECOM) COUNT\_DECOM,

SUM (NEW) COUNT\_NEW,

SUM (NEW\_TOTAL) COUNT\_NEW\_TOTAL,

SUM (NEW\_BY\_TESS) COUNT\_NEW\_BY\_TESS,

SUM (NEW\_NOT\_TESS\_CONTRACT) COUNT\_NEW\_NOT\_TESS\_CONTRACT,

SUM (NEW\_NOT\_TESS\_NO\_CONTRACT) COUNT\_NEW\_NOT\_TESS\_NO\_CONTRACT,

ROUND (

CASE

WHEN SUM (NEW\_TOTAL) > 0 THEN SUM (NEW\_BY\_TESS) \* 100.0 / SUM (NEW\_TOTAL)

ELSE

0.00 END, 2) PERCENTAGE\_NEW\_BY\_TESS

FROM

MAX\_COMMIT\_DATE

GROUP BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET

ORDER BY TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET;

END;

END CIRC\_SEG\_NEW\_DECOM\_RPT\_PKG;

/

--------------------------------------------------------

-- DDL for Package Body ENB\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."ENB\_AUDIT"

IS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: eNBAudit

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 10/25/2010 Neeharika 1. Created this package.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--- Foundation of eNB Audit

-- In order to avoid using oracle regex , pre parsing some stuff and keeping them in tables

-- pre parsed (/home2/wnea/Ng/perljobs/bin/parse\_Ng\_eNB on d1) all Ng eNB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Procedure eNbNeVsNgAudit is

CURSOR eNB\_Dups\_in\_Ng is

WITH

eNB\_Equip as

(

select distinct

ENB\_NUMBER,

eqp\_reference\_id,

domain\_id

from NTLS\_ENB\_PARSED\_V enb

)

select distinct ENB\_NUMBER, count(eqp\_reference\_id)

from eNB\_Equip enb

group by ENB\_NUMBER, domain\_id

having count(eqp\_reference\_id) > 1

;

/\* Declaring variables\*/

sql\_stmt varchar2(32000);

begin

sql\_stmt:='truncate table '||AUDIT\_TABLE\_NAME;

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr('Error: Could not truncate table: '||AUDIT\_TABLE\_NAME ||

'Error: '|| TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

-- create a full audit of discoverd eNB (e/// and alu) vs Ng

sql\_stmt := 'insert into '|| AUDIT\_TABLE\_NAME ||' WITH

ERIC\_ENB as

(

select enb.\*, ''OSS-RC'' from\_ems

from

'|| ERICSSON\_ENB\_WK ||'

enb

where regexp\_like( enb.NODE,''^[[:digit:]]{6}'' )

)

,

ALU\_ENB as

(

select enb.\*, ''SAM'' from\_ems

from

' || SAM\_ENB ||'

enb

where regexp\_like( enb.site\_id,''^[[:digit:]]{6}'' )

--and enb.enb\_status=''LIVE''

)

,

Ng\_EQUIP as

(

select xpe.\*

from NTLS\_ENB\_PARSED\_V xpe, market\_prefix\_enb\_naming mkt, clli\_domain\_map cli

where substr(xpe.enb\_number, 1, 3) = mkt.market\_id

and substr(mkt.clli, 1, 8) = cli.clli

and cli.leaf\_domain\_id = xpe.domain\_id

)

select

substr(enb.node, 1,3) ne\_market\_id

, enb.node

, from\_ems

, IP, connect\_status, sync\_status

, ''BOTH'' match\_code

, parse\_status match\_status

, eqp\_name Ng\_equip\_name

, case when eqp\_name is not null and enb\_number is null then ''Unparseable''

when eqp\_name is not null and enb\_number is not null then substr(enb\_number, 1,3)

end Ng\_market\_id

, ENB\_NUMBER Ng\_ENB\_NUMBER

, eqp\_reference\_id

'||

-- , eq\_class

-- , has\_multiples

'

, eqp\_vendor

, eqp\_model

, inv\_status

,case when enb.connect\_status = ''Connected'' and sync\_status = ''Synchronized'' then ''Up''

else ''Down''

end ne\_status

, ei.eqp\_type

, null comments

from ERIC\_ENB enb

--full outer

join Ng\_EQUIP ei

on ei.enb\_number = substr(enb.node, 1,6)

union

select substr(enb.site\_id, 1,3) ne\_market\_id

, enb.site\_id

, from\_ems

, null IP

, inv\_status connection\_status

, null synch\_status

, ''BOTH'' match\_code

, parse\_status match\_status

, eqp\_name Ng\_equip\_name

, case when eqp\_name is not null and enb\_number is null then ''Unparseable''

when eqp\_name is not null and enb\_number is not null then substr(enb\_number, 1,3)

end Ng\_market\_id

, ENB\_NUMBER Ng\_ENB\_NUMBER

, eqp\_reference\_id

'||

-- , eq\_class

-- , has\_multiples

'

, eqp\_vendor

, eqp\_model

, status

' ||

-- ,case when enb.inv\_status = ''LIVE'' then ''Up''

-- else ''Down''

-- end ne\_status

'

,''Up'' ne\_status

, ei.eqp\_type

, null comments

from ALU\_ENB enb

--full outer

join Ng\_EQUIP ei

on ei.enb\_number = substr(enb.site\_id, 1,6)

';

EXECUTE IMMEDIATE sql\_stmt;

-- update the ones that have any problem to NE ONLY

--or (Ng\_enb\_number is null and match\_status is not null )

-- eNB should be built as shelves in Ng. So anything that is not a shelf mark it bad match

-- i.e. NE Only

sql\_stmt:='

update '|| AUDIT\_TABLE\_NAME ||' aud

set match\_code = ''NE Only''

where

match\_code = ''BOTH''

and (aud.eq\_class <> ''S''

or has\_multiples is not null

)

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Updated the matched equip match\_code, which are not shelf, to NE Only');

-- eNB should be built as shelves in Ng. So anything that is not a shelf mark it bad match

-- i.e. NE Only

sql\_stmt:='

update '|| AUDIT\_TABLE\_NAME ||' aud

set match\_status = ''Equipment not built as Shelf in Ng''

where aud.eq\_class <> ''S'' AND match\_status is null

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Updated the matched equip match\_status, which are not shelf');

--insert NE ONLY from E///

sql\_stmt:='

insert into '|| AUDIT\_TABLE\_NAME ||'

select substr(enb.node, 1,3) ne\_market\_id

, enb.node

, ''OSS-RC''

, IP, connection\_status, synch\_status

, ''NE Only'' match\_code

, null match\_status

, null Ng\_equip\_name

, null Ng\_market\_id

, null Ng\_ENB\_NUMBER

, null eqp\_reference\_id

, null eq\_class

, null has\_multiples

, null vendor

, null model

, null status

,enb.server\_name

,case when enb.CONNECTION\_STATUS = ''Connected'' and synch\_status = ''Synchronized'' then ''Up''

else ''Down''

end ne\_status

, '''' type

, null comments

from

'|| ERICSSON\_ENB\_WK ||' enb

,

(select node

FROM '|| ERICSSON\_ENB\_WK ||' enb

where connection\_status = ''Connected''

and synch\_status = ''Synchronized''

and regexp\_like( NODE,''^[[:digit:]]{6}'' )

minus

select aud.NODE

from '|| AUDIT\_TABLE\_NAME ||' aud

) miss

where miss.node = enb.NODE

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Inserted NE Only e///');

--insert NE ONLY from ALU

sql\_stmt:='

insert into '|| AUDIT\_TABLE\_NAME ||'

select substr(enb.node, 1,3) ne\_market\_id

, enb.node

, ''SAM''

, null IP, enb\_status connection\_status, null synch\_status

, ''NE Only'' match\_code

, null match\_status

, null Ng\_equip\_name

, null Ng\_market\_id

, null Ng\_ENB\_NUMBER

, null eqp\_reference\_id

, null eq\_class

, null has\_multiples

, null vendor

, null model

, null status

,enb.server\_name

,case when enb.enb\_status = ''LIVE'' then ''Up''

else ''Down''

end ne\_status

, '''' type

, null comments

from

'|| SAM\_ENB ||' enb

,

(select node

FROM '|| SAM\_ENB ||' enb

--where enb.enb\_status=''LIVE''

where regexp\_like( NODE,''^[[:digit:]]{6}'' )

minus

select aud.NODE

from '|| AUDIT\_TABLE\_NAME ||' aud

) miss

where miss.node = enb.NODE

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Inserted NE Only ALU');

-- insert Ng\_only. here both ALU and E/// will be inserted

sql\_stmt:='

insert into '|| AUDIT\_TABLE\_NAME ||'

select null ne\_market\_id

, null node

, null from\_ems

, null IP

, null connection\_status

, null synch\_status

, ''Ng Only'' match\_code

, parse\_status match\_status

, Ng.descr Ng\_equip\_name

, substr(Ng.ENB\_NUMBER, 1,3) Ng\_market\_id

, Ng.ENB\_NUMBER Ng\_ENB\_NUMBER

, Ng.eqp\_reference\_id

, Ng.eq\_class

, has\_multiples

, Ng.vendor

, Ng.model

, Ng.STATUS

, null ne\_server\_name

, null ne\_status

, Ng.type

, null comments

from NTLS\_ENB\_PARSED\_V Ng,

(

select eqp\_reference\_id

from NTLS\_ENB\_PARSED\_V Ng

minus

select AUD.eqp\_reference\_id

from '|| AUDIT\_TABLE\_NAME ||' aud

) miss

where miss.eqp\_reference\_id = Ng.eqp\_reference\_id';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Inserted NE Only ALU and E///');

-- if a new market id is discovered then mark the record to reflect it

sql\_stmt:='

update '|| AUDIT\_TABLE\_NAME ||' aud

set match\_status = ''New Market Id was discovered''

where

exists ( select 1

from

(

select NE\_MARKET\_ID

from '|| AUDIT\_TABLE\_NAME ||'

minus

select market\_id

from MARKET\_PREFIX\_ENB\_NAMING m

)

where ne\_market\_id = aud.ne\_market\_id

)

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Updated rec to indicate that a new Mkt\_id was discovered');

-- an undiscovered Mkt\_id was found in Ng

sql\_stmt:='

update '|| AUDIT\_TABLE\_NAME ||' aud

set match\_status = ''Unknown Market Id found in Ng''

where

exists ( select 1

from

(

select Ng\_MARKET\_ID

from '|| AUDIT\_TABLE\_NAME ||'

minus

select market\_id

from MARKET\_PREFIX\_ENB\_NAMING m

)

where Ng\_market\_id = aud.Ng\_market\_id

and aud.Ng\_EQUIP\_NAME is not null

)

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Updated record to indicate that an undiscovered Mkt\_id was found in Ng');

-- multiple equip in Ng have the same enb id

/\* if the equipment is in > 1 domain then counting eqp\_reference\_id in the table

gives wrong answer . So count enb\_number, count(eqp\_reference\_id) \*/

sql\_stmt := 'update '|| AUDIT\_TABLE\_NAME ||' aud

set has\_multiples = ''Y''

, match\_status = nvl2(match\_status, match\_status||'', Multiple equipments have same eNB identfier in Ng'', ''Multiple equipments have same eNB identfier in Ng'')

, match\_code = ''NE Only''

where substr(node, 1,6) = :ENB\_NUMBER';

for rec in eNB\_Dups\_in\_Ng

LOOP

EXECUTE IMMEDIATE sql\_stmt using rec.ENB\_NUMBER;

END LOOP;

commit;

dbms\_output.put\_line('Updated records where discovered enb id was found on multiple equipments in Ng');

-- update the match\_status to indicate duplicates and match\_code NE Only

sql\_stmt:='

update '|| AUDIT\_TABLE\_NAME ||' aud

set match\_code = ''NE Only'',

match\_status = nvl2(match\_status, match\_status||'', Duplicate in NE'', ''Duplicate in NE'')

where exists (

select node, count(1) dups from '|| SAM\_ENB ||'

where node = aud.node group by node

having count(1) > 1

union all

select node, count(1) dups from '|| ERICSSON\_ENB\_WK ||'

where node = aud.node group by node

having count(1) > 1

)

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Updated the match\_code to NE Only [and match\_status] for DUPLICATEs');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in enb\_audit.eNbNeVsNgAudit(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

Procedure generateMarketIdSummary is

/\* Declaring variables\*/

sql\_stmt varchar2(32767);

begin

sql\_stmt := 'truncate table eNB\_SUMMARY\_BY\_Mkt\_id\_wrk' ;

execute immediate(sql\_stmt);

sql\_stmt:=' insert into eNB\_SUMMARY\_BY\_Mkt\_id\_wrk columns(territory, market\_TERRITORY, sub\_market, market\_id,lte\_market\_name, leaf\_domain\_name

, total\_eNB, Built\_as\_Shelf, Live, Other, Are\_Duplicated,total\_compliance)

WITH

eNB\_MKT as

(

-- eliminate reserved blocks from the defined eNB market codes

select e.MARKET\_ID, e.lte\_market\_name, e.clli

from MARKET\_PREFIX\_ENB\_NAMING e

)

,

REPORTING\_DOMAINS as

(

-- eliminate domains that will not own an eNB

-- and add Unknown JIC we find stuff that does not belong anywhere else

select nvl(territory, ''Unknown'') territory, nvl(market\_territory, ''Unknown'') market\_territory, nvl(sub\_market,''Unknown'') sub\_market ,

cdv.LEAF\_DOMAIN\_INST\_ID

, cdv.PARENT\_DOMAIN\_INST\_ID, cdv.LEAF\_DOMAIN\_NAME

, cdv.clli

from clli\_domain\_map\_v cdv

where cdv.territory not in (''NNO'', ''OSS'')

)

,

MKT\_DOMAIN\_MAP as

(

select market\_id, e.lte\_market\_name

, e.clli

, territory, market\_territory,sub\_market

, cdv.leaf\_domain\_inst\_id, cdv.leaf\_domain\_name

from eNB\_MKT e

-- OUR MASTER TABLE might be missing some records.

-- So do a full outer join to capture those

left outer join REPORTING\_DOMAINS cdv

on substr(cdv.clli, 1,6) = substr(e.clli, 1,6)

)

,

REPORTING\_MARKET\_IDS as

(

select case when market\_id is null then substr(e.NODE, 1, 3)

else market\_id

end market\_id

, m.lte\_market\_name

,nvl(territory, ''Unknown'') territory, nvl(market\_territory, ''Unknown'') market\_territory, nvl(sub\_market,''Unknown'') sub\_market

, leaf\_domain\_inst\_id, leaf\_domain\_name

from MKT\_DOMAIN\_MAP m

full OUTER join OSS\_RC\_ENB e

on substr(e.NODE, 1, 3) = m.market\_id

where

e.CONNECTION\_STATUS = ''Connected''

and e.SYNCH\_STATUS = ''Synchronized''

UNION

select case when market\_id is null then substr(e.NODE, 1, 3)

else market\_id

end market\_id

, m.lte\_market\_name

, nvl(territory, ''Unknown'') territory, nvl(market\_territory, ''Unknown'') market\_territory, nvl(sub\_market,''Unknown'') sub\_market

, leaf\_domain\_inst\_id, leaf\_domain\_name

from MKT\_DOMAIN\_MAP m

full OUTER join SAM\_ENB e

on substr(e.NODE, 1, 3) = m.market\_id

where e.enb\_status=''LIVE''

)

,

eNB\_MKT\_MATCH as

(

select case when market\_id is null then substr(e.NODE, 1, 3)

else market\_id

end market\_id

, m.lte\_market\_name

, leaf\_domain\_inst\_id, leaf\_domain\_name

, e.NODE, e.IP

from MKT\_DOMAIN\_MAP m

right OUTER join OSS\_RC\_ENB e

on substr(e.NODE, 1, 3) = m.market\_id

where

e.CONNECTION\_STATUS = ''Connected''

and e.SYNCH\_STATUS = ''Synchronized''

UNION

select distinct case when market\_id is null then substr(e.NODE, 1, 3)

else market\_id

end market\_id

, m.lte\_market\_name

, leaf\_domain\_inst\_id, leaf\_domain\_name

, e.NODE, NULL IP

from MKT\_DOMAIN\_MAP m

right OUTER join SAM\_ENB e

on substr(e.NODE, 1, 3) = m.market\_id

where e.enb\_status=''LIVE''

)

,

TOT as

(

-- since we might discover some new market\_ids that are not in our

-- master table, we need to do put nvl on area and region

select e.market\_id

, count(1) total

from eNB\_MKT\_MATCH e

group by e.market\_id

)

,

BUILT\_AS\_SHELF as

(

-- since we are reporting the discovered spans that matched

-- we dont need to worry about "mkt ids, that were not discovered"

select enb.market\_id, count(1) as\_shelf

from NE\_VS\_Ng\_ENB\_AUDIT\_WRK aud

join eNB\_MKT\_MATCH enb

on enb.node = aud.NODE

where match\_code = ''BOTH''

and eq\_class = ''S''

and ne\_status=''Up''

group by enb.market\_id

)

,

MATCHED\_LIVE as

(

-- since we are reporting the discovered spans that matched

-- we dont need to worry about "mkt ids, that were not discovered"

select enb.market\_id

, count(1) live

from NE\_VS\_Ng\_ENB\_AUDIT\_WRK aud

join eNB\_MKT\_MATCH enb

on enb.node = aud.NODE

-- no need to call check that these are shelves since non-shelf matches

-- are not marked BOTH in the audit

where match\_code = ''BOTH''

and aud.STATUS = ''Live''

and ne\_status=''Up''

group by enb.market\_id

)

,

MATCHED\_N\_LIVE as

(

-- since we are reporting the discovered spans that matched

-- we dont need to worry about "mkt ids, that were not discovered"

select enb.market\_id

, count(1) n\_live

from NE\_VS\_Ng\_ENB\_AUDIT\_WRK aud

join eNB\_MKT\_MATCH enb

on enb.node = aud.node

where match\_code = ''BOTH''

and aud.STATUS <> ''Live''

and ne\_status=''Up''

group by enb.market\_id

)

,

DUP as

(

-- since we are reporting the discovered spans that matched

-- we dont need to worry about "mkt ids, that were not discovered"

select market\_id

, count(1) multiples

from

(

select distinct enb.market\_id, aud.NODE

from NE\_VS\_Ng\_ENB\_AUDIT\_WRK aud

join eNB\_MKT\_MATCH enb

on enb.node = aud.node

where match\_code = ''NE Only''

and aud.HAS\_MULTIPLES = ''Y''

-- and match\_status = ''Multiple equipments have same eNB identfier''

)

group by market\_id

)

,

STATS\_BY\_MKT\_ID as

(

select territory, market\_territory, sub\_market

, mkt.market\_id

, mkt.lte\_market\_name

, mkt.leaf\_domain\_name

, sum(total) Total\_eNB

, sum(as\_shelf) shelves, sum(live) Live\_shelves

, sum(n\_live) n\_live\_shelves, sum(multiples) multiple\_shelves

--, round( nvl(sum(live)-sum(multiples),0)/( sum(total)-sum(n\_live)),4)"% Compliant"

from REPORTING\_MARKET\_IDS mkt

left outer join TOT t

on t.market\_id = mkt.market\_id

left outer join BUILT\_AS\_SHELF bas

on bas.market\_id = mkt.market\_id

left outer join MATCHED\_LIVE ml

on ml.market\_id = mkt.market\_id

left outer join MATCHED\_N\_LIVE mnl

on mnl.market\_id = mkt.market\_id

left outer join DUP d

on d.market\_id = mkt.market\_id

group by territory, market\_territory, sub\_market, mkt.MARKET\_ID, mkt.lte\_market\_name, mkt.leaf\_domain\_name

)

select territory, market\_territory, sub\_market, '''''''' || market\_id

, lte\_market\_name

, leaf\_domain\_name

, total\_eNB

, shelves Built\_as\_Shelf

, live\_shelves Live

, n\_live\_shelves Other

, multiple\_shelves Are\_Duplicated

, case when (nvl(TOTAL\_ENB, 0) - nvl(n\_live\_shelves, 0) > 0) then round (nvl(live\_shelves, 0)/nullif (nvl(TOTAL\_ENB, 0) - nvl(n\_live\_shelves, 0), 0) \*100, 2) else 100 end total\_compliance

from STATS\_BY\_MKT\_ID

ORDER BY territory, market\_territory, sub\_market, market\_id

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in enb\_audit.generateMarketIdSummary(): '

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

End generateMarketIdSummary;

Procedure generateRegionalSummary is

/\* Declaring variables\*/

sql\_stmt varchar2(32767);

begin

sql\_stmt := 'truncate table ENB\_SUMMARY\_BY\_REGION\_WRK' ;

execute immediate(sql\_stmt);

sql\_stmt := 'insert into ENB\_SUMMARY\_BY\_REGION\_WRK columns

(territory, market\_territory, sub\_market, total\_enb, built\_as\_shelf, live, other, are\_duplicated, enb\_reported, total\_compliance)

select replace(TERRITORY, ''\_Grand'', ''Grand Total'') TERRITORY, market\_territory, sub\_market,TOTAL\_ENB, BUILT\_AS\_SHELF, LIVE, OTHER, ARE\_DUPLICATED, enb\_reported,total\_compliance from(

select nvl(territory,''\_Grand'')territory, market\_territory, sub\_market, sum( mkt.TOTAL\_ENB) TOTAL\_ENB

, sum(mkt.BUILT\_AS\_SHELF ) BUILT\_AS\_SHELF, sum(mkt.LIVE ) LIVE, sum(mkt.OTHER) OTHER, sum(mkt.ARE\_DUPLICATED) ARE\_DUPLICATED

, sum( nvl(mkt.TOTAL\_ENB, 0)) - sum(nvl(mkt.OTHER, 0)) enb\_reported

,round (sum(nvl(mkt.LIVE, 0))/decode (sum( nvl(mkt.TOTAL\_ENB, 0)) - sum(nvl(mkt.OTHER, 0)), 0, 1, sum(nvl(mkt.TOTAL\_ENB, 0)) - sum(nvl(mkt.OTHER, 0))) \*100, 2) total\_compliance

from ENB\_SUMMARY\_BY\_MKT\_ID\_WRK mkt

group by rollup(territory, market\_territory, sub\_market)

order by territory, market\_territory, sub\_market)

where market\_territory != ''Unknown'' or market\_territory is null

';

EXECUTE IMMEDIATE sql\_stmt;

-- Added so missing area/regions will show with 0 count

sql\_stmt := 'insert into ENB\_SUMMARY\_BY\_REGION\_WRK columns (territory, market\_territory, sub\_market, total\_enb, built\_as\_shelf, live, other, are\_duplicated, enb\_reported, total\_compliance)

SELECT d.territory, d.market\_territory, d.sub\_market, 0, 0, 0, 0, 0, 0, 0

FROM DOMAINS\_REGIONAL\_REPORTING d

left join enb\_summary\_by\_region\_wrk e

on d.territory = e.territory and d.market\_territory = e.market\_territory

and d.sub\_market = e.sub\_market

WHERE d.territory NOT IN (''OSS'', ''NNO'') and e.territory is null';

EXECUTE IMMEDIATE sql\_stmt;

commit;

sql\_stmt:= 'update ENB\_SUMMARY\_BY\_REGION\_WRK wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sql\_stmt;

commit;

sql\_stmt:= 'update ENB\_SUMMARY\_BY\_REGION\_WRK wk set wk.TERRITORY\_MARKET\_SUB = market\_territory where sub\_market is null and wk.market\_territory is not null';

EXECUTE IMMEDIATE sql\_stmt;

commit;

sql\_stmt:= 'update ENB\_SUMMARY\_BY\_REGION\_WRK wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null and wk.market\_territory is null' ;

EXECUTE IMMEDIATE sql\_stmt;

commit;

sql\_stmt:= 'update ENB\_SUMMARY\_BY\_REGION\_WRK wk set wk.market\_territory =''Unknown'',wk.sub\_market=''Unknown'' where territory=''Unknown''' ;

EXECUTE IMMEDIATE sql\_stmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in enb\_audit.generateRegionalSummary(): '

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

End generateRegionalSummary;

--PROCEDURE copy\_enb\_wrk IS

---- sqlStmt varchar2(32000);

---- v\_ready Ng\_reports.all\_processes.is\_ready%type;

-- BEGIN

---- select is\_ready

---- into v\_ready

---- from Ng\_reports.all\_processes

---- where process\_name = 'ENB\_AUDIT';

----

---- IF v\_ready = 'Y' then

---- sqlStmt:='DELETE FROM NE\_VS\_Ng\_ENB\_AUDIT';

---- execute immediate sqlStmt;

---- sqlStmt := 'insert into NE\_VS\_Ng\_ENB\_AUDIT select \* from '||AUDIT\_TABLE\_NAME;

---- execute immediate sqlStmt;

---- commit;

---- dbms\_output.put\_line('Done populating NE\_VS\_Ng\_ENB\_AUDIT');

----

---- sqlStmt := 'truncate table ENB\_SUMMARY\_BY\_REGION';

---- execute immediate sqlStmt;

---- sqlStmt := 'insert into ENB\_SUMMARY\_BY\_REGION select \* from ENB\_SUMMARY\_BY\_REGION\_WRK';

---- execute immediate sqlStmt;

---- commit;

---- dbms\_output.put\_line('Done populating ENB\_SUMMARY\_BY\_REGION');

----

---- sqlStmt := 'truncate table eNB\_SUMMARY\_BY\_Mkt\_id';

---- execute immediate sqlStmt;

---- sqlStmt := 'insert into eNB\_SUMMARY\_BY\_Mkt\_id select \* from eNB\_SUMMARY\_BY\_Mkt\_id\_wrk';

---- execute immediate sqlStmt;

---- commit;

---- dbms\_output.put\_line('Done populating eNB\_SUMMARY\_BY\_Mkt\_id');

---- END IF;

---- EXCEPTION

---- WHEN OTHERS THEN

---- WATCHDOG.logerror('ENB\_AUDIT',4000,SubStr('Error in copy\_enb\_wrk; '

---- ||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255),'Y');

---- WATCHDOG.updateprocessend('ENB\_AUDIT','STATUS\_FAILURE','N');

PROCEDURE truncate\_enb\_wrk IS

sqlStmt varchar2(32000);

BEGIN

sqlStmt := 'delete from '||AUDIT\_TABLE\_NAME;

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting '||AUDIT\_TABLE\_NAME);

sqlStmt := 'delete from ENB\_SUMMARY\_BY\_REGION\_WRK';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting ENB\_SUMMARY\_BY\_REGION\_WRK');

sqlStmt := 'delete from eNB\_SUMMARY\_BY\_Mkt\_id\_wrk';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting eNB\_SUMMARY\_BY\_Mkt\_id\_wrk');

END;

--PROCEDURE restore\_enb\_wrk IS

-- sqlStmt varchar2(32000);

-- BEGIN

-- sqlStmt:='DELETE FROM '||AUDIT\_TABLE\_NAME;

-- execute immediate sqlStmt;

-- sqlStmt := 'insert into '||AUDIT\_TABLE\_NAME||' select \* from NE\_VS\_Ng\_ENB\_AUDIT ';

-- execute immediate sqlStmt;

-- commit;

-- dbms\_output.put\_line('Done populating NE\_VS\_Ng\_ENB\_AUDIT');

--

-- sqlStmt := 'truncate table ENB\_SUMMARY\_BY\_REGION';

-- execute immediate sqlStmt;

-- sqlStmt := 'insert into ENB\_SUMMARY\_BY\_REGION select \* from ENB\_SUMMARY\_BY\_REGION\_WRK';

-- execute immediate sqlStmt;

-- commit;

-- dbms\_output.put\_line('Done populating nortel\_cdma\_audit\_reg\_summ\_wrk');

--

-- sqlStmt := 'truncate table eNB\_SUMMARY\_BY\_Mkt\_id';

-- execute immediate sqlStmt;

-- sqlStmt := 'insert into eNB\_SUMMARY\_BY\_Mkt\_id select \* from eNB\_SUMMARY\_BY\_Mkt\_id\_wrk';

-- execute immediate sqlStmt;

-- commit;

-- dbms\_output.put\_line('Done populating eNB\_SUMMARY\_BY\_Mkt\_id');

-- EXCEPTION

-- WHEN OTHERS THEN

-- WATCHDOG.logerror('ENB\_AUDIT',4000,SubStr('Error in restore\_enb\_wrk; '

-- ||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255),'Y');

-- WATCHDOG.updateprocessend('ENB\_AUDIT','STATUS\_FAILURE','N');

--

--END;

END eNB\_Audit;

/

--------------------------------------------------------

-- DDL for Package Body LOAD\_NTLS\_CSR\_GIGE\_TABLES

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."LOAD\_NTLS\_CSR\_GIGE\_TABLES"

IS

Procedure runLoad is

BEGIN

BEGIN

get\_csr\_gige\_trails();

get\_csr\_gige\_parents();

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr('Error: btod\_ntls\_audit\_upd(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

END;

END runLoad;

-- to get csr gige trails

procedure get\_csr\_gige\_trails is

sql\_stmt varchar2(200);

BEGIN

sql\_stmt:='truncate table ntls\_csr\_gige\_trails\_wk';

EXECUTE IMMEDIATE sql\_stmt; commit;

insert into NTLS\_CSR\_GIGE\_TRAILS\_WK

select \* from (

with GET\_CSR\_PORTS as (

select te.z\_site\_id, te.equipment\_id, te.element\_ref\_id port\_id, te.port\_name, te.BANDWIDTH,

tc.trail\_id, tc.bandwidth path\_bw, tc.trail\_name,

tn.next\_revision\_trail\_id, tn.bandwidth next\_bw, tn.trail\_name next\_trail\_name,

cp.csr\_live\_in\_xng, case when tc.STATUS = 'LIVE' then 'Y' else 'N' end gige\_live\_in\_xng

from ntls\_csr\_parsed\_v cp

join ng\_topology.trail\_element te

on te.equipment\_id = cp.eqp\_reference\_id

left outer join ng\_topology.trail tc

on te.trail\_id = tc.trail\_id

left outer join ng\_topology.trail tn

on te.trail\_id = tn.next\_revision\_trail\_id

where

te.element\_type = 'E' and te.port\_name is not null and te.port\_name <> 'PORT\_NAME'

and ((tc.TYPE in ( 'EBH','UNI' ) AND tc.BANDWIDTH <> 'VLAN')

or

(tn.TYPE in ( 'EBH','UNI' ) AND tn.BANDWIDTH <> 'VLAN'))

),

GigE\_W\_eNB as (

select te.trail\_id, te.equipment\_id, te.element\_ref\_id port\_id, te.port\_name

from ng\_topology.trail\_element te

join GET\_CSR\_PORTS csr

on te.trail\_id in (csr.trail\_id, csr.next\_revision\_trail\_id)

join ntls\_enb\_parsed\_v enb

on enb.eqp\_reference\_id = te.equipment\_id

where te.element\_type = 'E' and te.port\_name is not null and te.port\_name <> 'PORT\_NAME')

select distinct p.equipment\_id, p.port\_id, p.trail\_id, p.next\_revision\_trail\_id, csr\_live\_in\_xng, gige\_live\_in\_xng

from GET\_CSR\_PORTS p

where not exists (select 1

from GigE\_W\_eNB e

where e.port\_id = p.port\_id

and e.equipment\_id = p.equipment\_id

)

);

commit;

end get\_csr\_gige\_trails;

-- for giges with no vlans, then store its parents

procedure get\_csr\_gige\_parents is

CURSOR all\_giges is

(select distinct trail\_id gige\_trail\_id

from ntls\_csr\_gige\_trails\_wk

union

select distinct next\_revision\_trail\_id gige\_trail\_id

from ntls\_csr\_gige\_trails\_wk)

;

sql\_stmt varchar2(200);

BEGIN

-- get all vlans on these gige trails

sql\_stmt:='truncate table ntls\_csr\_gige\_parents\_wk';

EXECUTE IMMEDIATE sql\_stmt; commit;

for rec in all\_giges

loop

get\_gige\_parents\_in\_1st\_leg(rec.gige\_trail\_id, rec.gige\_trail\_id, '', 0);

commit;

end loop;

end get\_csr\_gige\_parents;

/\* this gets all parents ONLY in the first leg of the trail

the code is written to not go past 10 parents up.

This is helpful in the case where due to problem-modelling in xng, there is a loop in data

without this you get an error "maximum open cursors exceeded"

\*/

procedure get\_gige\_parents\_in\_1st\_leg (start\_trail in number, current\_trail in number, p\_sys\_conn\_by\_trail in varchar2, deep in number) is

cursor PARENTS\_IN\_1ST\_LEG (ARG\_TRAIL number) is

select \*

from (

select distinct tc.trail\_id gige\_trail\_id, te.parent\_trail\_id,

tc.component\_id trail\_component\_id,

tc.name trail\_component\_name, tc.sequence,

rank () over (partition by tc.trail\_id order by sequence) as first\_leg,

t.topology, te.element\_type

from ng\_topology.trail t

join ng\_topology.trail\_element te

on t.trail\_ID = te.trail\_id

join ng\_topology.trail\_component tc

on tc.trail\_id = te.trail\_id

where te.trail\_id = ARG\_TRAIL

and te.element\_type in ('P', 'K')

)

where first\_leg = 1

;

sql\_stmt varchar2(2000) := 'insert into ntls\_csr\_gige\_parents\_wk columns

(gige\_trail\_id, parent\_trail\_id, trail\_component\_id, trail\_component\_name, sys\_conn\_by\_trail

)

values(:gige\_trail\_id, :parent\_trail\_id, :trail\_component\_id, :trail\_component\_name, :sys\_conn\_by\_trail

)';

sys\_conn\_by\_trail varchar2(1000) := p\_sys\_conn\_by\_trail;

BEGIN

sys\_conn\_by\_trail := sys\_conn\_by\_trail||'/'||current\_trail;

for rec in PARENTS\_IN\_1ST\_LEG(current\_trail)

loop

EXECUTE IMMEDIATE sql\_stmt using start\_trail, rec.parent\_trail\_id,

rec.trail\_component\_id, rec.trail\_component\_name,

sys\_conn\_by\_trail||'/'||rec.parent\_trail\_id

;

--dbms\_output.PUT\_LINE (sys\_conn\_by\_trail);

if (upper(rec.topology) <> 'U' -- Uni-Ring'

and upper(rec.topology) <> 'R' -- 'Bi-Ring'

and rec.element\_type <> 'K' --trail link

)

then

if (deep <= RECURSIVE\_DEPTH)

THEN

--recursively call the same procedure for the parent\_trail

get\_gige\_parents\_in\_1st\_leg(start\_trail, rec.parent\_trail\_id, sys\_conn\_by\_trail, deep+1);

end if;

END IF;

end loop;

end get\_gige\_parents\_in\_1st\_leg;

-- to get csr gige trails

procedure get\_csr\_gige\_delivery is

sql\_stmt varchar2(200);

BEGIN

sql\_stmt:='truncate table ntls\_csr\_gige\_delivery\_wk';

EXECUTE IMMEDIATE sql\_stmt; commit;

insert into ntls\_csr\_gige\_delivery\_wk

(select \* from (

WITH PARENTS\_OF\_GIGE\_W\_SEG as

(

-- get only those parents of GigE that have a segment in them

select pp.gige\_trail\_id, pp.parent\_trail\_id, te.element\_ref\_id segment\_id, te.group\_number seg\_order

from ntls\_csr\_gige\_parents\_wk pp

join ng\_topology.trail\_element te

on te.trail\_id = pp.parent\_trail\_id

and te.element\_type = 'S'

)

select distinct gige.gige\_trail\_id, ts.segment\_id, gige.seg\_order,

case when upper(ts.bandwidth) like 'OC%' then 'OPTICAL BW'

else ts.bandwidth

end delivered\_on

from PARENTS\_OF\_GIGE\_W\_SEG gige

join ng\_topology.trail pt

on pt.trail\_id = gige.parent\_trail\_id

join ng\_topology.trail gt

on gt.trail\_id = gige.gige\_trail\_id

join ng\_reports.ntls\_csr\_gige\_trails\_wk ncgt

on gige.gige\_trail\_id in (ncgt.trail\_id, ncgt.next\_revision\_trail\_id)

join ng\_topology.trail\_segment ts

on ts.segment\_id = gige.segment\_id

join ng\_reports.equipment e

on ncgt.equipment\_id = e.eqp\_reference\_id

--where e.site\_reference\_id in ( pt.a\_site\_id, pt.z\_site\_id)

--where gt.z\_site\_id in (pt.a\_site\_id, pt.z\_site\_id)

));

commit;

end get\_csr\_gige\_delivery;

END;

/

--------------------------------------------------------

-- DDL for Package Body LUCENT\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."LUCENT\_AUDIT" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: lucent\_audit

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 2/4/2011 1. Created this package body.

2.0 2/15/2016 VenkatP 2. Market structure changes.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE auditNgVsProspect

IS

BEGIN

BEGIN

auditLucentBTSTerminations();

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr('Error in auditLucentBTSTerminations(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

BEGIN

generateLucentSwitchSummary();

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr('Error in generateLucentSwitchSummary(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

BEGIN

generateLucentRegSummary();

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr('Error in generateLucentRegSummary(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

END;

PROCEDURE auditLucentBTSTerminations

IS

sql\_stmt varchar2(32676);

-- One Live and N non Live match

CURSOR t1\_nl\_match IS

SELECT sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num, ne\_ds1\_num

FROM Lucent\_ne\_vs\_ng\_audit

WHERE match\_code='BOTH'

AND ng\_path\_status='LIVE'

AND termination\_type = 'T1';

-- One Live and N non Live match

CURSOR ebh\_nl\_match IS

SELECT sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num

FROM Lucent\_ne\_vs\_ng\_audit

WHERE match\_code='BOTH'

AND ng\_path\_status='LIVE'

AND termination\_type = 'Ethernet';

cursor upd\_ethernet\_both\_multiples is

select sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num, count(1)

from Lucent\_ne\_vs\_ng\_audit a

where a.MATCH\_CODE = 'BOTH'

and a.TERMINATION\_TYPE = 'Ethernet'

and ng\_path\_status = 'LIVE'

group by sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num

having count(1) > 1

;

cursor upd\_t1\_both\_multiples is

select sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num, ne\_ds1\_num, count(1)

from Lucent\_ne\_vs\_ng\_audit a

where a.MATCH\_CODE = 'BOTH'

and a.TERMINATION\_TYPE = 'T1'

and ng\_path\_status = 'LIVE'

group by sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num, ne\_ds1\_num

having count(1) > 1

;

cursor upd\_ethernet\_ngonly\_multiples is

select sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num, count(1)

from Lucent\_ne\_vs\_ng\_audit a

where a.MATCH\_CODE = 'Ng Only'

and a.TERMINATION\_TYPE = 'Ethernet'

and ng\_path\_status = 'LIVE'

group by sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num

having count(1) > 1

;

cursor upd\_t1\_ngonly\_multiples is

select sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num, ne\_ds1\_num, count(1)

from Lucent\_ne\_vs\_ng\_audit a

where a.MATCH\_CODE = 'Ng Only'

and a.TERMINATION\_TYPE = 'T1'

and ng\_path\_status = 'LIVE'

group by sys\_id, ecp\_sid, ne\_cell\_num, ne\_urc\_num, ne\_ds1\_num

having count(1) > 1

;

--JC

cursor update\_lucent\_wrk\_no\_csr\_port is

WITH

LUXA\_PATHS as

(

select lu.TRAIL\_ID

from LUCENT\_NE\_VS\_NG\_audit lu

where lu.TERMINATION\_TYPE = 'Ethernet'

AND lu.BANDWIDTH LIKE '%Mbps'

and lu.match\_code = 'BOTH'

and lu.NG\_PATH\_STATUS = 'LIVE'

)

,

PATHS\_W\_CSR as

(

select cpi.TRAIL\_ID

from LUXA\_PATHS cpi

join NG\_TOPOLOGY.TRAIL\_ELEMENT cpe

on cpi.trail\_id = cpe.trail\_id

join ng\_reportS.PORT p

on CPE.ELEMENT\_REF\_ID = P.PORT\_REFERENCE\_ID and cpe.ELEMENT\_TYPE='E'

join ng\_reportS.EQUIPMENT ei

on p.eqp\_reference\_id = ei.eqp\_reference\_id

and ei.eqp\_type = 'CSR'

)

select LUXA.TRAIL\_ID

from LUXA\_PATHS luxa

MINUS

select pc.TRAIL\_ID

from PATHS\_W\_CSR pc;

BEGIN

EXECUTE IMMEDIATE 'drop index LUCENT\_NE\_VS\_XNG\_AUDT\_IDX';

EXECUTE IMMEDIATE 'drop index LNVXA\_BTS\_URC\_DS1\_IDX ';

EXECUTE IMMEDIATE 'drop index LNVXA\_PATHINST\_STATUS\_TYPE\_IDX';

sql\_stmt:= 'truncate table '||AUDIT\_TBL;

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in auditngVsProspect(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

commit;

sql\_stmt:='insert into '|| AUDIT\_TBL ||'

WITH

NG\_SPANS\_PER\_SWITCH as

(

select distinct vcp.switch\_id,

vcp.cell\_number ng\_cell\_num,

vcp.urc\_crc as ng\_urc\_num,

vcp.ds1\_number as ng\_ds1\_num,

vcp.TRAIL\_ID TRAIL\_ID,

vcp.PATH\_NAME,

vcp.ng\_path\_status,

vcp.ng\_path\_type,

vcp.parse\_status

from VZW\_LUCENT\_CELL\_PATHS vcp

)

,

BTS\_ETHER\_TERMINATIONS as

(

select ne.\*, esm.switch\_id, ''Ethernet'' termination\_type

from PROS\_ECP\_CELL\_EBH\_MAP ne

join ECP\_SWITCH\_MAP esm

on esm.SYS\_ID = ne.sys\_id

and esm.ecp\_sid = ne.ecp\_sid

)

,

BTS\_T1\_TERMINATIONS as

(

select ne.\*, esm.switch\_id, ''T1'' termination\_type

from PROS\_ECP\_CELL\_DS1\_MAP ne

join ECP\_SWITCH\_MAP esm

on esm.SYS\_ID = ne.sys\_id

and esm.ecp\_sid = ne.ecp\_sid

)

,

ETHERNET\_AUDIT AS

(

SELECT bet.bts\_name ne\_bts\_name

, bet.cell\_type ne\_cell\_type

, bet.sys\_id, bet.ecp\_sid

, bet.cell\_number ne\_cell\_num

, bet.urc\_num ne\_urc\_num

, null ne\_ds1\_num

, bet.version ne\_version

, bet.switch\_id

, bet.service

, bet.termination\_type

, cpi.ng\_cell\_num

, cpi.ng\_urc\_num

, cpi.ng\_ds1\_num

, cpi.TRAIL\_ID

, cpi.path\_name

, cpi.ng\_path\_status

, cpi.ng\_path\_type

, vcpi.bandwidth

, CASE

WHEN cpi.switch\_id IS NULL

THEN ''NE Only''

WHEN cpi.ng\_ds1\_num IS not NULL and upper(vcpi.bandwidth) like ''%MBPS%''

THEN ''NE Only''

WHEN cpi.ng\_ds1\_num IS not NULL and upper(vcpi.bandwidth) not like ''%MBPS%''

THEN ''NE Only''

WHEN cpi.ng\_ds1\_num IS NULL and upper(vcpi.bandwidth) not like ''%MBPS%''

THEN ''NE Only''

WHEN cpi.ng\_ds1\_num IS NULL and upper(vcpi.bandwidth) like ''%MBPS%''

THEN ''BOTH''

END match\_code

, CASE

WHEN cpi.switch\_id IS NULL

THEN ''NE Only''

WHEN cpi.ng\_ds1\_num IS not NULL and upper(vcpi.bandwidth) like ''%MBPS%''

THEN ''Pathname Invalid Format''

WHEN cpi.ng\_ds1\_num IS not NULL and upper(vcpi.bandwidth) not like ''%MBPS%''

THEN ''Pathname Invalid Format, Path Invalid Bandwidth''

WHEN cpi.ng\_ds1\_num IS NULL and upper(vcpi.bandwidth) not like ''%MBPS%''

THEN ''Path Invalid Bandwidth''

WHEN cpi.ng\_ds1\_num IS NULL and upper(vcpi.bandwidth) like ''%MBPS%''

THEN ''BOTH''

else parse\_status

END match\_status

FROM BTS\_ETHER\_TERMINATIONS bet

LEFT OUTER JOIN NG\_SPANS\_PER\_SWITCH cpi

ON bet.switch\_id = cpi.switch\_id

AND bet.cell\_number = cpi.ng\_cell\_num

AND bet.urc\_num = cpi.ng\_urc\_num

LEFT OUTER JOIN NG\_TOPOLOGY.TRAIL vcpi

on cpi.TRAIL\_ID = vcpi.TRAIL\_ID

)

,

T1\_AUDIT AS

(

SELECT bet.bts\_name ne\_bts\_name

, bet.cell\_type ne\_cell\_type

, bet.sys\_id, bet.ecp\_sid

, bet.cell\_number ne\_cell\_num

, bet.urc\_num ne\_urc\_num

, bet.ds1\_num ne\_ds1\_num

, bet.version ne\_version

, bet.switch\_id

, bet.service

, bet.termination\_type

, cpi.ng\_cell\_num

, cpi.ng\_urc\_num

, cpi.ng\_ds1\_num

, cpi.TRAIL\_ID

, cpi.path\_name

, cpi.ng\_path\_status

, cpi.ng\_path\_type

, vcpi.bandwidth

, CASE

WHEN cpi.switch\_id IS NULL

THEN ''NE Only''

WHEN upper(vcpi.bandwidth) not like ''%DS1%''

THEN ''NE Only''

WHEN upper(vcpi.bandwidth) like ''%DS1%''

THEN ''BOTH''

END match\_code

, CASE

WHEN cpi.switch\_id IS NULL

THEN null -- ''NE Only''

WHEN upper(vcpi.bandwidth) not like ''%DS1%''

THEN ''Path Invalid Bandwidth''

WHEN upper(vcpi.bandwidth) like ''%DS1%''

THEN null -- ''BOTH''

else parse\_status

END match\_status

FROM BTS\_T1\_TERMINATIONS bet

LEFT OUTER JOIN NG\_SPANS\_PER\_SWITCH cpi

ON bet.switch\_id = cpi.switch\_id

AND bet.cell\_number = cpi.ng\_cell\_num

AND bet.urc\_num = cpi.ng\_urc\_num

AND bet.ds1\_num = cpi.ng\_ds1\_num

LEFT OUTER JOIN NG\_TOPOLOGY.TRAIL vcpi

on cpi.TRAIL\_ID = vcpi.TRAIL\_ID

)

,

NE\_ONLY\_AND\_BOTH AS

(

SELECT \*

FROM ETHERNET\_AUDIT ea

UNION

SELECT \*

FROM T1\_AUDIT t1a

)

,

NG\_ONLY\_PATHS AS

(

SELECT TRAIL\_ID

FROM NG\_SPANS\_PER\_SWITCH cpi

MINUS

SELECT TRAIL\_ID

FROM NE\_ONLY\_AND\_BOTH

)

SELECT a.\*, TRUNC (SYSDATE) AS audit\_date

FROM NE\_ONLY\_AND\_BOTH A

UNION

SELECT null ne\_bts\_name

, null ne\_cell\_type

, null sys\_id

, null ecp\_sid

, null ne\_cell\_num

, null ne\_urc\_num

, null ne\_ds1\_num

, null ne\_version

, cpi.switch\_id

, null service

, null termination\_type

, cpi.ng\_cell\_num

, cpi.ng\_urc\_num

, cpi.ng\_ds1\_num

, cpi.TRAIL\_ID

, cpi.path\_name

, cpi.ng\_path\_status

, cpi.ng\_path\_type

, vcpi.bandwidth

, ''Ng Only'' match\_code

, parse\_status match\_status

, TRUNC (SYSDATE) AS audit\_date

FROM NG\_ONLY\_PATHS a

join NG\_SPANS\_PER\_SWITCH cpi

on a.TRAIL\_ID = cpi.TRAIL\_ID

LEFT OUTER JOIN NG\_TOPOLOGY.TRAIL vcpi

on cpi.TRAIL\_ID = vcpi.TRAIL\_ID

';

EXECUTE IMMEDIATE sql\_stmt;

commit;

sql\_stmt:='CREATE INDEX LUCENT\_NE\_VS\_XNG\_AUDT\_IDX ON '|| AUDIT\_TBL ||'

(sys\_id, ecp\_sid, NE\_CELL\_NUM, NE\_URC\_NUM, NE\_DS1\_NUM)';

EXECUTE IMMEDIATE sql\_stmt;

commit;

sql\_stmt:='CREATE INDEX LNVXA\_BTS\_URC\_DS1\_IDX ON '|| AUDIT\_TBL ||'

(SWITCH\_ID, NG\_CELL\_NUM, NG\_URC\_NUM, NG\_DS1\_NUM)';

EXECUTE IMMEDIATE sql\_stmt;

commit;

sql\_stmt:='CREATE INDEX LNVXA\_PATHINST\_STATUS\_TYPE\_IDX ON '|| AUDIT\_TBL ||'

(TRAIL\_ID, NG\_PATH\_STATUS, NG\_PATH\_TYPE)';

EXECUTE IMMEDIATE sql\_stmt;

commit;

-- Ng Only and BW = DS1 then it is T1

sql\_stmt:=' update '|| AUDIT\_TBL ||' lt1 set lt1.TERMINATION\_TYPE=''T1''

where exists (

select lt.TRAIL\_ID

from '|| AUDIT\_TBL ||' lt, NG\_TOPOLOGY.TRAIL cpi

where lt.MATCH\_CODE=''Ng Only''

and cpi.TRAIL\_ID = lt.TRAIL\_ID

and cpi.BANDWIDTH=''DS1''

and lt1.TRAIL\_ID = lt.TRAIL\_ID

)';

EXECUTE IMMEDIATE sql\_stmt;

commit;

-- Ng Only and BW <> DS1 then it is Ethernet

sql\_stmt:=' update '|| AUDIT\_TBL ||' lt1 set lt1.TERMINATION\_TYPE=''Ethernet''

where exists (

select lt.TRAIL\_ID

from '|| AUDIT\_TBL ||' lt, NG\_TOPOLOGY.TRAIL cpi

where lt.MATCH\_CODE=''Ng Only''

and cpi.TRAIL\_ID = lt.TRAIL\_ID

and cpi.BANDWIDTH<>''DS1''

and lt1.TRAIL\_ID = lt.TRAIL\_ID

)';

EXECUTE IMMEDIATE sql\_stmt;

commit;

FOR cursorRec IN upd\_ethernet\_both\_multiples LOOP

UPDATE Lucent\_ne\_vs\_ng\_audit u

set match\_status = match\_status || ', Multiple spans found in Ng for same tokens'

, match\_code='NE Only'

WHERE --rowid = cursorRec.ROWID

u.sys\_id = cursorRec.sys\_id

and u.ecp\_sid = cursorRec.ecp\_sid

and u.ne\_cell\_num = cursorRec.ne\_cell\_num

and u.ne\_urc\_num = cursorRec.ne\_urc\_num

;

END LOOP;

commit;

FOR cursorRec IN upd\_t1\_both\_multiples LOOP

UPDATE Lucent\_ne\_vs\_ng\_audit u

set match\_status = match\_status || ', Multiple spans found in Ng for same tokens'

, match\_code='NE Only'

WHERE --rowid = cursorRec.ROWID

u.sys\_id = cursorRec.sys\_id

and u.ecp\_sid = cursorRec.ecp\_sid

and u.ne\_cell\_num = cursorRec.ne\_cell\_num

and u.ne\_urc\_num = cursorRec.ne\_urc\_num

and u.ne\_ds1\_num = cursorRec.ne\_ds1\_num

;

END LOOP;

commit;

FOR cursorRec IN upd\_ethernet\_ngonly\_multiples LOOP

UPDATE Lucent\_ne\_vs\_ng\_audit u

set match\_status = ', Multiple spans found in Ng for same tokens'

WHERE --rowid = cursorRec.ROWID

u.sys\_id = cursorRec.sys\_id

and u.ecp\_sid = cursorRec.ecp\_sid

and u.ne\_cell\_num = cursorRec.ne\_cell\_num

and u.ne\_urc\_num = cursorRec.ne\_urc\_num

;

END LOOP;

commit;

FOR cursorRec IN upd\_t1\_ngonly\_multiples LOOP

UPDATE Lucent\_ne\_vs\_ng\_audit u

set match\_status = nvl2(match\_status, 'Multiple spans found in Ng for same tokens,'||match\_status, 'Multiple spans found in Ng for same tokens')

WHERE --rowid = cursorRec.ROWID

u.sys\_id = cursorRec.sys\_id

and u.ecp\_sid = cursorRec.ecp\_sid

and u.ne\_cell\_num = cursorRec.ne\_cell\_num

and u.ne\_urc\_num = cursorRec.ne\_urc\_num

and u.ne\_ds1\_num = cursorRec.ne\_ds1\_num

;

END LOOP;

commit;

sql\_stmt:='update Lucent\_ne\_vs\_ng\_audit u

set match\_code = ''Ng Only''

where u.NG\_PATH\_TYPE=''AMPS''

'

;

EXECUTE IMMEDIATE sql\_stmt;

commit;

FOR cursorRec IN t1\_nl\_match LOOP

UPDATE Lucent\_ne\_vs\_ng\_audit mnvxa

SET match\_status = nvl2(match\_status, 'Live Path Exists, '||match\_status ,'Live Path Exists')

WHERE mnvxa.sys\_id = cursorRec.sys\_id

AND mnvxa.ecp\_sid = cursorRec.ecp\_sid

AND mnvxa.ne\_cell\_num = cursorRec.ne\_cell\_num

AND mnvxa.ne\_urc\_num = cursorRec.ne\_urc\_num

AND mnvxa.ne\_ds1\_num = cursorRec.ne\_ds1\_num

AND mnvxa.termination\_type='T1'

AND mnvxa.ng\_path\_status <> 'LIVE'

-- AND mnvxa.match\_code='BOTH'

;

END LOOP;

COMMIT;

FOR cursorRec IN ebh\_nl\_match LOOP

UPDATE Lucent\_ne\_vs\_ng\_audit mnvxa

SET match\_status = nvl2(match\_status, 'Live Path Exists, '||match\_status ,'Live Path Exists')

WHERE mnvxa.sys\_id = cursorRec.sys\_id

AND mnvxa.ecp\_sid = cursorRec.ecp\_sid

AND mnvxa.ne\_cell\_num = cursorRec.ne\_cell\_num

AND mnvxa.ne\_urc\_num = cursorRec.ne\_urc\_num

AND mnvxa.termination\_type='Ethernet'

AND mnvxa.ng\_path\_status <> 'LIVE'

--AND mnvxa.match\_code='BOTH'

;

END LOOP;

COMMIT;

--JC

FOR cursorRec IN update\_lucent\_wrk\_no\_csr\_port LOOP

UPDATE Lucent\_ne\_vs\_ng\_audit lu

SET MATCH\_CODE = 'NE Only',

match\_status =

case

when MATCH\_STATUS = 'BOTH'

then 'No CSR port on the e-pipe'

else

nvl2(match\_status, 'No CSR port on the e-pipe, '||match\_status, 'No CSR port on the e-pipe')

end

WHERE

lu.TRAIL\_ID = cursorRec.TRAIL\_ID

;

END LOOP;

commit;

--JC

dbms\_output.put\_line('Lucent audit completed successfully');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in auditLucentBTSTerminations(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

end auditLucentBTSTerminations;

procedure generateLucentRegSummary as

sqlstmt varchar2(32767);

TBL\_NAME varchar2(30) := 'LUCENT\_AUDIT\_REG\_SUMM';

BEGIN

sqlstmt := 'truncate table '||TBL\_NAME;

begin

execute immediate sqlstmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlstmt);

dbms\_output.put\_line(SubStr('Error in generateLucentRegSummary(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

sqlstmt := ' insert into '||TBL\_NAME ||' columns(territory, market\_territory, sub\_market, t1\_discovered, t1\_matched\_live, t1\_matched\_nonlive, ebh\_discovered,

ebh\_matched\_live, ebh\_matched\_nonlive, t1\_comp, ebh\_CSR\_matched\_live, ebh\_CSR\_matched\_nonlive, ebh\_csr\_reported,

ebh\_csr\_comp,overall\_csr\_comp,ebh\_comp,overall\_comp)

select drr.territory, drr.market\_territory, drr.sub\_market, sum(NVL (t1\_discovered, 0)) t1\_discovered, sum(NVL (t1\_matched\_live, 0)) t1\_matched\_live,

sum(NVL (t1\_matched\_nonlive, 0)) t1\_matched\_nonlive, sum(NVL (ebh\_discovered, 0)) ebh\_discovered,

sum(NVL (ebh\_matched\_live, 0)) ebh\_matched\_live, sum(NVL (ebh\_matched\_nonlive, 0)) ebh\_matched\_nonlive,

case when SUM (NVL (t1\_discovered, 0)) - SUM (NVL (t1\_matched\_nonlive, 0)) <> 0

then round (SUM (NVL (t1\_matched\_live, 0)) / (SUM (NVL (t1\_discovered, 0)) - SUM (NVL (t1\_matched\_nonlive, 0)))\*100, 2)

else 0

end t1\_comp,

sum(NVL (ebh\_CSR\_matched\_live, 0)) ebh\_CSR\_matched\_live,

sum(NVL (ebh\_CSR\_matched\_nonlive, 0)) ebh\_CSR\_matched\_nonlive, sum(NVL (ebh\_csr\_reported, 0)) ebh\_csr\_reported,

case when SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_CSR\_matched\_nonlive, 0)) <> 0

then round (SUM (NVL (ebh\_csr\_matched\_live, 0)) / (SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_CSR\_matched\_nonlive, 0)))\*100, 2)

else 0

end ebh\_csr\_comp,

case when SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0)) <> 0

then round (((sum (nvl(t1\_matched\_live,0)+nvl(ebh\_csr\_matched\_live,0)))/(SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0))- SUM(NVL (ebh\_CSR\_matched\_nonlive, 0)+NVL (t1\_matched\_nonlive, 0))))\*100,2)

else 0

end overall\_csr\_comp,

case when SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_matched\_nonlive, 0)) <> 0

then round (SUM (NVL (ebh\_matched\_live, 0)) / (SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_matched\_nonlive, 0)))\*100, 2)

else 0

end ebh\_comp,

case when SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0)) <> 0

then round (((sum (nvl(t1\_matched\_live,0)+nvl(ebh\_matched\_live,0)))/(SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0))- SUM(NVL (ebh\_matched\_nonlive, 0)+NVL (t1\_matched\_nonlive, 0))))\*100,2)

else 0

end overall\_comp

from ng\_reports.LUCENT\_AUDIT\_SWITCH\_SUMM mrm

right outer join domains\_regional\_reporting drr on DRR.market\_territory = mrm.market\_territory and DRR.sub\_market = mrm.sub\_market

where territory not in (''NNO'', ''OSS'')

group by rollup (drr.territory, drr.market\_territory, drr.sub\_market )

order by drr.territory, drr.market\_territory, drr.sub\_market ';

execute immediate sqlstmt;

commit;

sqlstmt:= 'update LUCENT\_AUDIT\_REG\_summ wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlstmt:= 'update LUCENT\_AUDIT\_REG\_summ wk set wk.TERRITORY\_MARKET\_SUB = MARKET\_TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlstmt:= 'update LUCENT\_AUDIT\_REG\_summ wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is null' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlstmt:= 'update LUCENT\_AUDIT\_REG\_summ wk set wk.MARKET\_TERRITORY =''Unknown'',wk.sub\_market=''Unknown'' where territory=''Unknown''' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlstmt);

dbms\_output.put\_line(SubStr('Error in auditLucentBTSTerminations(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

procedure generateLucentSwitchSummary as

sqlstmt varchar2(32767);

TBL\_NAME varchar2(30) := 'LUCENT\_AUDIT\_SWITCH\_summ';

BEGIN

sqlstmt := 'drop table '||TBL\_NAME;

begin

execute immediate sqlstmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlstmt);

dbms\_output.put\_line(SubStr('Error in generateLucentRegSummary(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

end;

sqlstmt := ' create table '||TBL\_NAME ||' as

WITH

ECPs as

(

SELECT \*

FROM ng\_reports.ecp\_switch\_map

)

,

ecp\_region\_map AS

(

SELECT DISTINCT cdmp.territory, cdmp.market\_territory, cdmp.sub\_market, switch\_id, cdmp.leaf\_domain\_name

FROM ng\_reports.clli\_domain\_map\_v cdmp

JOIN ECPs m

ON SUBSTR (switch\_id, 1, 6) = SUBSTR (cdmp.clli, 1, 6)

WHERE cdmp.territory <> ''NNO''

),

t1\_discovered AS

(

SELECT DISTINCT switch\_id, nvxa.ecp\_sid, nvxa.ds1\_num, NVXA.CELL\_NUMBER, NVXA.URC\_NUM

FROM ng\_reports.PROS\_ECP\_CELL\_DS1\_MAP nvxa

join ECPs e

on nvxa.sys\_id = e.sys\_id and nvxa.ecp\_sid = e.ecp\_sid

),

t1\_discovered\_cnt AS

(

SELECT DISTINCT switch\_id, COUNT (1) t1\_discovered

FROM t1\_discovered

group by switch\_id

),

ethernet\_discovered AS

(

SELECT DISTINCT switch\_id, nvxa.ecp\_sid, NVXA.CELL\_NUMBER, NVXA.URC\_NUM

FROM ng\_reports.PROS\_ECP\_CELL\_EBH\_MAP nvxa

join ECPs e

on nvxa.sys\_id = e.sys\_id and nvxa.ecp\_sid = e.ecp\_sid

),

ethernet\_discovered\_cnt AS

(

SELECT DISTINCT switch\_id, count(1) ebh\_discovered

FROM ethernet\_discovered

group by switch\_id

),

t1\_matched\_live as

(

SELECT DISTINCT nvxa.switch\_id, nvxa.NE\_CELL\_NUM, nvxa.NE\_URC\_NUM

, nvxa.NE\_DS1\_NUM

FROM ng\_reports.Lucent\_ne\_vs\_ng\_audit nvxa

WHERE nvxa.termination\_type = ''T1''

AND nvxa.match\_code IN (''BOTH'')

and nvxa.NG\_PATH\_STATUS = ''LIVE''

),

t1\_matched\_live\_cnt AS

(

SELECT switch\_id, COUNT (1) t1\_matched\_live

FROM t1\_matched\_live

GROUP BY switch\_id

)

,

ethernet\_matched\_live as

(

SELECT DISTINCT nvxa.switch\_id, nvxa.NE\_CELL\_NUM, nvxa.NE\_URC\_NUM

FROM ng\_reports.Lucent\_ne\_vs\_ng\_audit nvxa

WHERE nvxa.termination\_type = ''Ethernet''

AND nvxa.match\_code IN (''BOTH'')

and nvxa.NG\_PATH\_STATUS = ''LIVE''

)

,

ethernet\_matched\_live\_cnt AS

(

SELECT switch\_id, COUNT (1) ebh\_matched\_live

FROM ethernet\_matched\_live

GROUP BY switch\_id

)

,

t1\_matched\_n\_live\_cnt AS

(

SELECT switch\_id, COUNT (1) t1\_matched\_n\_live

FROM (

-- good matched non live minus all live ones

SELECT nvxa.switch\_id, nvxa.NE\_CELL\_NUM, nvxa.NE\_URC\_NUM

, nvxa.NE\_DS1\_NUM

FROM ng\_reports.Lucent\_ne\_vs\_ng\_audit nvxa

WHERE nvxa.termination\_type = ''T1''

AND nvxa.match\_code IN (''BOTH'')

and nvxa.NG\_PATH\_STATUS <> ''LIVE''

minus

SELECT nvxa.switch\_id, nvxa.NE\_CELL\_NUM, nvxa.NE\_URC\_NUM

, nvxa.NE\_DS1\_NUM

from t1\_matched\_live nvxa

)

GROUP BY switch\_id

)

,

ethernet\_matched\_n\_live\_cnt AS

(

SELECT switch\_id, COUNT (1) ebh\_matched\_n\_live

FROM (

-- good matched non live minus all live ones

SELECT nvxa.switch\_id, nvxa.NE\_CELL\_NUM, nvxa.NE\_URC\_NUM

FROM ng\_reports.Lucent\_ne\_vs\_ng\_audit nvxa

WHERE nvxa.termination\_type = ''Ethernet''

AND nvxa.match\_code IN (''BOTH'')

and nvxa.NG\_PATH\_STATUS <> ''LIVE''

minus

SELECT nvxa.switch\_id, nvxa.NE\_CELL\_NUM, nvxa.NE\_URC\_NUM

from ethernet\_matched\_live nvxa

)

GROUP BY switch\_id

)

,

--JC

ethernet\_csr\_matched\_live as

(

select DISTINCT lu.switch\_id, lu.NE\_CELL\_NUM, lu.NE\_URC\_NUM

from LUCENT\_NE\_VS\_NG\_audit lu

join NG\_TOPOLOGY.TRAIL cpi on LU.TRAIL\_ID = CPI.TRAIL\_ID

join ng\_reportS.PORT p on cpi.TRAIL\_ID = P.CKT\_PATH\_REFERENCE\_ID

join ng\_reportS.EQUIPMENT ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where lu.TERMINATION\_TYPE = ''Ethernet''

AND lu.BANDWIDTH LIKE ''%Mbps''

and lu.match\_code = ''BOTH''

and lu.match\_status = ''BOTH''

and lu.NG\_PATH\_STATUS = ''LIVE''

and EI.EQP\_TYPE = ''CSR''

GROUP BY lu.switch\_id, lu.NE\_CELL\_NUM, lu.NE\_URC\_NUM

)

,

ethernet\_csr\_matched\_live\_cnt AS

(

SELECT switch\_id, COUNT (1) ebh\_csr\_matched\_live

FROM ethernet\_csr\_matched\_live

GROUP BY switch\_id

)

,

ethernet\_csr\_matched\_non\_live as

(

select DISTINCT lu.switch\_id, lu.NE\_CELL\_NUM, lu.NE\_URC\_NUM

from LUCENT\_NE\_VS\_NG\_audit lu

join NG\_TOPOLOGY.TRAIL cpi on LU.TRAIL\_ID = CPI.TRAIL\_ID

join ng\_reportS.PORT p on cpi.TRAIL\_ID = P.CKT\_PATH\_REFERENCE\_ID

join ng\_reportS.EQUIPMENT ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where lu.TERMINATION\_TYPE = ''Ethernet''

AND lu.BANDWIDTH LIKE ''%Mbps''

and lu.match\_code = ''BOTH''

and lu.match\_status = ''BOTH''

and lu.NG\_PATH\_STATUS <> ''LIVE''

and EI.EQP\_TYPE = ''CSR''

GROUP BY lu.switch\_id, lu.NE\_CELL\_NUM, lu.NE\_URC\_NUM

)

,

ethernet\_csr\_matched\_nlv\_cnt AS

(

SELECT switch\_id, COUNT (1) ebh\_csr\_matched\_nonlive

FROM ethernet\_csr\_matched\_non\_live

GROUP BY switch\_id

)

--JC

SELECT mrm.market\_territory, mrm.sub\_market, mrm.leaf\_domain\_name, mrm.switch\_id,

SUM (NVL (t1\_discovered, 0)) AS t1\_discovered,

SUM (NVL (t1\_matched\_live, 0)) AS t1\_matched\_live,

SUM (NVL (t1\_matched\_n\_live, 0)) AS t1\_matched\_nonlive,

case when SUM (NVL (t1\_discovered, 0)) - SUM (NVL (t1\_matched\_n\_live, 0)) <> 0

then round (SUM (NVL (t1\_matched\_live, 0)) / (SUM (NVL (t1\_discovered, 0)) - SUM (NVL (t1\_matched\_n\_live, 0)))\*100, 2)

else 0

end t1\_comp,

SUM (NVL (ebh\_discovered, 0)) AS ebh\_discovered,

SUM (NVL (ebh\_matched\_live, 0)) AS ebh\_matched\_live,

SUM (NVL (ebh\_matched\_n\_live, 0)) AS ebh\_matched\_nonlive,

SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_matched\_n\_live, 0)) ebh\_reported,

--JC

SUM (NVL (ebh\_csr\_matched\_live, 0)) AS ebh\_CSR\_matched\_live,

SUM (NVL (ebh\_csr\_matched\_nonlive, 0)) AS ebh\_CSR\_matched\_nonlive,

SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_csr\_matched\_nonlive, 0)) ebh\_csr\_reported,

case when SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_CSR\_matched\_nonlive, 0)) <> 0

then round (SUM (NVL (ebh\_csr\_matched\_live, 0)) / (SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_CSR\_matched\_nonlive, 0)))\*100, 2)

else 0

end ebh\_csr\_comp,

case when SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0)) <> 0

then round (((sum (nvl(t1\_matched\_live,0)+nvl(ebh\_csr\_matched\_live,0)))/(SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0))- SUM(NVL (ebh\_CSR\_matched\_nonlive, 0)+NVL (t1\_matched\_n\_live, 0))))\*100,2)

else 0

end overall\_csr\_comp,

--JC

case when SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_matched\_n\_live, 0)) <> 0

then round (SUM (NVL (ebh\_matched\_live, 0)) / (SUM (NVL (ebh\_discovered, 0)) - SUM (NVL (ebh\_matched\_n\_live, 0)))\*100, 2)

else 0

end ebh\_comp,

case when SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0)) <> 0

then round (((sum (nvl(t1\_matched\_live,0)+nvl(ebh\_matched\_live,0)))/(SUM(nvl(t1\_discovered,0) + nvl(ebh\_discovered,0))- SUM(NVL (ebh\_matched\_n\_live, 0)+NVL (t1\_matched\_n\_live, 0))))\*100,2)

else 0

end overall\_comp

FROM ecp\_region\_map mrm

LEFT OUTER JOIN t1\_discovered\_cnt td ON mrm.switch\_id = td.switch\_id

LEFT OUTER JOIN t1\_matched\_live\_cnt t1\_l ON mrm.switch\_id = t1\_l.switch\_id

LEFT OUTER JOIN t1\_matched\_n\_live\_cnt t1\_n\_l ON mrm.switch\_id = t1\_n\_l.switch\_id

LEFT OUTER JOIN ethernet\_discovered\_cnt ed ON mrm.switch\_id = ed.switch\_id

LEFT OUTER JOIN ethernet\_matched\_live\_cnt e\_l ON mrm.switch\_id = e\_l.switch\_id

LEFT OUTER JOIN ethernet\_matched\_n\_live\_cnt e\_n\_l ON mrm.switch\_id = e\_n\_l.switch\_id

--JC

LEFT OUTER JOIN ethernet\_csr\_matched\_live\_cnt e\_c\_l ON mrm.switch\_id = e\_c\_l.switch\_id

LEFT OUTER JOIN ethernet\_csr\_matched\_nlv\_cnt e\_c\_nl ON mrm.switch\_id = e\_c\_nl.switch\_id

--JC

GROUP BY mrm.market\_territory, mrm.sub\_market, mrm.leaf\_domain\_name, mrm.switch\_id

ORDER BY mrm.market\_territory, mrm.sub\_market, mrm.leaf\_domain\_name, mrm.switch\_id';

execute immediate sqlstmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlstmt);

dbms\_output.put\_line(SubStr('Error in generateLucentSwitchSummary(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

PROCEDURE truncate\_lucent\_wrk IS

sqlStmt varchar2(32000);

BEGIN

sqlStmt := 'delete from lucent\_ne\_vs\_ng\_audit';

execute immediate sqlStmt;

dbms\_output.put\_line('Done deleting lucent\_ne\_vs\_ng\_audit');

sqlStmt := 'delete from lucent\_audit\_reg\_summ';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting lucent\_audit\_reg\_summ');

sqlStmt := 'delete from lucent\_audit\_switch\_summ';

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Done deleting lucent\_audit\_switch\_summ');

END;

END lucent\_audit;

/

--------------------------------------------------------

-- DDL for Package Body MOTOROLA\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."MOTOROLA\_AUDIT" AS

FUNCTION set\_message (

fullmsg IN VARCHAR2,

message IN VARCHAR2

) RETURN VARCHAR2 IS

status VARCHAR2(256);

BEGIN

IF

( fullmsg IS NULL )

THEN

status := message;

ELSE

status := fullmsg

|| ', '

|| message;

END IF;

RETURN status;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error in SET\_MESSAGE(): '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

END set\_message;

FUNCTION get\_ubs\_match\_status (

ds1\_number IN NUMBER,

bandwidth IN VARCHAR,

bts\_type IN VARCHAR,

ist1 IN INTEGER,

isubs IN INTEGER,

parsestats IN VARCHAR

) RETURN VARCHAR2 IS

status VARCHAR2(256) := parsestats;

BEGIN

IF

ist1 = t1\_type

THEN

IF

upper(bandwidth) NOT LIKE '%DS1%'

THEN

status := set\_message(status,'Path Invalid Bandwidth');

END IF;

ELSE

IF

ds1\_number IS NULL

THEN

IF

upper(bandwidth) LIKE '%DS1%'

THEN

status := set\_message(status,'Path Invalid Bandwidth');

END IF;

ELSE

status := set\_message(status,'Path Invalid Format');

IF

upper(bandwidth) LIKE '%DS1%'

THEN

status := set\_message(status,'Path Invalid Bandwidth');

END IF;

END IF;

END IF;

IF

isubs = ubs\_type

THEN

IF

bts\_type IS NOT NULL AND bts\_type NOT LIKE 'UB%'

THEN

status := set\_message(status,'Invalid UBS Pathname');

END IF;

ELSE

IF

bts\_type LIKE 'UB%'

THEN

status := set\_message(status,'Invalid UBS Pathname');

END IF;

END IF;

RETURN status;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error in GET\_EVDO\_T1\_MATCH\_STATUS(): '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

END get\_ubs\_match\_status;

PROCEDURE moto\_cdma\_audit AS

-- One Live and N non Live match

CURSOR t1\_nl\_match IS SELECT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_span\_number

FROM

moto\_cdma\_ne\_vs\_ng\_audit

WHERE

match\_code = 'BOTH'

AND upper(ng\_path\_status) = 'LIVE'

AND termination\_type = 'T1';

-- One Live and N non Live match

CURSOR ebh\_nl\_match IS SELECT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number

FROM

moto\_cdma\_ne\_vs\_ng\_audit

WHERE

match\_code = 'BOTH'

AND upper(ng\_path\_status) = 'LIVE'

AND termination\_type = 'Ethernet';

-- T1 spans matching multiple live paths in xng or matching multiple non live paths in xng in case a live match does not exist are marked NE Only

CURSOR t1\_cdma\_multiples\_xng IS WITH dup AS (

SELECT DISTINCT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_span\_number,

COUNT(

CASE

WHEN upper(ng\_path\_status) = 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_span\_number

) AS live\_count,

COUNT(

CASE

WHEN upper(Ng\_path\_status) != 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_span\_number

) AS non\_live\_count

FROM

moto\_cdma\_ne\_vs\_ng\_audit src

WHERE

src.match\_code = 'BOTH'

AND src.termination\_type = 'T1'

AND src.ne\_bts\_status IN (

'OOS',

'INS',

'INS\_ACTIVE'

)

) SELECT

tgt.rowid

FROM

moto\_cdma\_ne\_vs\_ng\_audit tgt

JOIN dup ON dup.ne\_omcr = tgt.ne\_omcr

AND dup.ne\_bts\_number = tgt.ne\_bts\_number

AND dup.ne\_cluster\_number = tgt.ne\_cluster\_number

AND dup.ne\_span\_number = tgt.ne\_span\_number

AND (

dup.live\_count > 1

OR (

dup.live\_count = 0

AND dup.non\_live\_count > 1

)

);

-- Ethernet spans matching multiple live paths in xng or matching multiple non live paths in xng incase a live match does not exist are marked NE Only

CURSOR ethernet\_cdma\_multiples\_xng IS WITH dup AS (

SELECT DISTINCT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

COUNT(

CASE

WHEN upper(ng\_path\_status) = 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number

) AS live\_count,

COUNT(

CASE

WHEN upper(ng\_path\_status) != 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number

) AS non\_live\_count

FROM

moto\_cdma\_ne\_vs\_ng\_audit src

WHERE

src.match\_code = 'BOTH'

AND src.termination\_type = 'Ethernet'

AND src.ne\_bts\_status IN (

'OOS',

'INS',

'INS\_ACTIVE'

)

) SELECT

tgt.rowid

FROM

moto\_cdma\_ne\_vs\_ng\_audit tgt

JOIN dup ON dup.ne\_omcr = tgt.ne\_omcr

AND dup.ne\_bts\_number = tgt.ne\_bts\_number

AND dup.ne\_cluster\_number = tgt.ne\_cluster\_number

AND (

dup.live\_count > 1

OR (

dup.live\_count = 0

AND dup.non\_live\_count > 1

)

);

--JC

-- Update those paths that dont have a CSR port

CURSOR update\_mcdma\_wrk\_no\_csr\_port IS WITH mxa\_paths AS (

SELECT

mxa.ng\_path\_inst\_id

FROM

moto\_cdma\_ne\_vs\_ng\_audit mxa

WHERE

mxa.termination\_type = 'Ethernet'

AND mxa.ng\_bandwidth LIKE '%bps%'

AND mxa.match\_code = 'BOTH'

AND upper(mxa.ng\_path\_status) = 'LIVE'

),paths\_w\_csr AS (

SELECT

cpi.ng\_path\_inst\_id

FROM

mxa\_paths cpi

JOIN ng\_topology.trail\_element cpe ON cpi.ng\_path\_inst\_id = cpe.trail\_id

where cpe.equipment\_type = 'CSR'

) SELECT

mxap.ng\_path\_inst\_id

FROM

mxa\_paths mxap

MINUS

SELECT

pc.ng\_path\_inst\_id

FROM

paths\_w\_csr pc;

-- mis-matched BTS Type

CURSOR bts\_type\_match IS SELECT

ROWID rid

--NE\_OMCR,NE\_BTS\_NUMBER,NE\_CLUSTER\_NUMBER,NE\_SPAN\_NUMBER

FROM

moto\_cdma\_ne\_vs\_ng\_audit

WHERE

match\_code = 'BOTH'

AND substr(ne\_bts\_type,1,2) = 'UB';

-- find the duplicated entries and remove the NE Only entry

CURSOR ne\_ethernet\_multiples IS SELECT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number

FROM

moto\_cdma\_ne\_vs\_ng\_audit a

WHERE

a.match\_code = 'BOTH'

AND a.termination\_type LIKE '%Eth%';

CURSOR ne\_t1\_multiples IS SELECT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_span\_number

FROM

moto\_cdma\_ne\_vs\_ng\_audit a

WHERE

a.match\_code = 'BOTH'

AND a.termination\_type = 'T1';

sql\_stmt VARCHAR2(32000);

BEGIN

sql\_stmt := 'truncate TABLE MOTO\_CDMA\_NE\_VS\_NG\_AUDIT';

BEGIN

EXECUTE IMMEDIATE sql\_stmt;

-- dbms\_output.put\_line(sql\_stmt);

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(substr('Error in moto\_cdma\_audit();: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

END;

COMMIT;

INSERT INTO moto\_cdma\_ne\_vs\_ng\_audit (

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_bts\_router\_group,

ne\_bandwidth,

ne\_span\_number,

ne\_span\_type,

ne\_access\_node\_number,

ne\_ag\_node\_number,

ne\_ag\_span\_number,

ne\_t1\_line\_number,

ne\_bts\_type,

ne\_bts\_status,

termination\_type,

switch\_id,

mtx,

ng\_path\_name,

ng\_path\_inst\_id,

ng\_ipbscdo\_number,

ng\_bts\_number,

ng\_cluster\_number,

ng\_span\_number,

ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

match\_code,

match\_status,

extract\_date,

audit\_date

)

WITH bts\_status AS (

SELECT DISTINCT

\*

FROM

mt\_omcr\_bts\_status stat

WHERE

stat.bts\_mlppp\_number IS NULL

AND stat.pbts\_number IS NULL

),moto\_ether\_terminations AS (

SELECT

inv.vsm\_device\_name\_omcr,

inv.bts\_number,

inv.cluster\_number,

inv.router\_grp,

inv.bandwidth,

inv.extract\_date,

substr(bmm.vsm\_device\_name\_mtx,1,6)

|| substr(bmm.vsm\_device\_name\_mtx,11,2) switch\_id,

'Ethernet' termination\_type,

bmm.vsm\_device\_name\_mtx,

mp.bts\_type,

stat.tel\_state ne\_bts\_status

FROM

moto\_cdma\_ethernet\_inv inv

JOIN ng\_reports.bsm\_mtx\_map bmm ON inv.vsm\_device\_name\_omcr = bmm.vsm\_device\_name\_bsm

AND upper(bmm.mtx\_status) = 'LIVE'

AND upper(bmm.bsm\_status) = 'LIVE'

JOIN moto\_bts\_location\_map mp ON inv.vsm\_device\_name\_omcr = mp.vsm\_device\_name\_omcr

AND inv.bts\_number = mp.bts\_number

JOIN bts\_status stat ON inv.vsm\_device\_name\_omcr = stat.vsm\_device\_name\_omcr

AND inv.bts\_number = stat.bts\_number

),moto\_t1\_terminations AS (

SELECT

mcs.bts\_number,

mcs.cluster\_number,

mcs.bts\_router\_group,

mcs.span\_number,

mcs.access\_node\_number,

mcs.ag\_node\_number,

mcs.ag\_span\_number,

mcs.t1\_line\_number,

mcs.extract\_date,

mcs.vsm\_device\_name\_omcr,

substr(bmm.vsm\_device\_name\_mtx,1,6)

|| substr(bmm.vsm\_device\_name\_mtx,11,2) switch\_id,

'T1' termination\_type,

bmm.vsm\_device\_name\_mtx,

mp.bts\_type,

stat.tel\_state ne\_bts\_status

FROM

moto\_cdma\_t1\_inv mcs

JOIN ng\_reports.bsm\_mtx\_map bmm ON mcs.vsm\_device\_name\_omcr = bmm.vsm\_device\_name\_bsm

AND upper(bmm.mtx\_status) = 'LIVE'

AND upper(bmm.bsm\_status) = 'LIVE'

JOIN moto\_bts\_location\_map mp ON mcs.vsm\_device\_name\_omcr = mp.vsm\_device\_name\_omcr

AND mcs.bts\_number = mp.bts\_number

JOIN bts\_status stat ON mcs.vsm\_device\_name\_omcr = stat.vsm\_device\_name\_omcr

AND mcs.bts\_number = stat.bts\_number

),moto\_ether\_audit AS (

SELECT

met.vsm\_device\_name\_omcr ne\_omcr,

met.bts\_number ne\_bts\_number,

met.cluster\_number ne\_cluster\_number,

met.router\_grp ne\_bts\_router\_group,

met.bandwidth ne\_bandwidth,

NULL ne\_span\_number,

'VOICE' ne\_span\_type,

NULL ne\_access\_node\_number,

NULL ne\_ag\_node\_number,

NULL ne\_ag\_span\_number,

NULL ne\_t1\_line\_number,

met.bts\_type ne\_bts\_type,

met.ne\_bts\_status,

met.termination\_type,

met.switch\_id,

met.vsm\_device\_name\_mtx mtx,

cpi.trail\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

NULL ng\_ipbscdo\_number,

cpi.bts\_number ng\_bts\_number,

cpi.cluster\_number ng\_cluster\_number,

cpi.span\_number ng\_span\_number,

cpi.bandwidth ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

CASE

WHEN cpi.switch\_id IS NULL THEN 'NE Only'

ELSE 'BOTH'

END

match\_code,

CASE

WHEN cpi.switch\_id IS NULL THEN 'Not found in Granite'

ELSE get\_ubs\_match\_status(cpi.span\_number,cpi.bandwidth,met.bts\_type,ebh\_type,non\_ubs\_type,cpi.parse\_status)

END

match\_status,

met.extract\_date,

cpi.rowid cpi\_rowid

FROM

moto\_ether\_terminations met

LEFT OUTER JOIN (

SELECT

\*

FROM

vzw\_moto\_voice\_cell\_paths cp

WHERE

cp.bts\_number IS NOT NULL

AND cp.cluster\_number IS NOT NULL

) cpi ON met.switch\_id = cpi.switch\_id

AND met.bts\_number = cpi.bts\_number

AND met.cluster\_number = cpi.cluster\_number

),moto\_t1\_audit AS (

SELECT

mtt.vsm\_device\_name\_omcr ne\_omcr,

mtt.bts\_number ne\_bts\_number,

mtt.cluster\_number ne\_cluster\_number,

mtt.bts\_router\_group ne\_bts\_router\_group,

NULL ne\_bandwidth,

mtt.span\_number ne\_span\_number,

'VOICE' ne\_span\_type,

mtt.access\_node\_number ne\_access\_node\_number,

mtt.ag\_node\_number ne\_ag\_node\_number,

mtt.ag\_span\_number ne\_ag\_span\_number,

mtt.t1\_line\_number ne\_t1\_line\_number,

mtt.bts\_type ne\_bts\_type,

mtt.ne\_bts\_status,

mtt.termination\_type,

mtt.switch\_id,

mtt.vsm\_device\_name\_mtx mtx,

cpi.trail\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

NULL ng\_ipbscdo\_number,

cpi.bts\_number ng\_bts\_number,

cpi.cluster\_number ng\_cluster\_number,

cpi.span\_number ng\_span\_number,

cpi.bandwidth ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

CASE

WHEN cpi.switch\_id IS NULL THEN 'NE Only'

ELSE 'BOTH'

END

match\_code,

CASE

WHEN cpi.switch\_id IS NULL THEN 'Not found in Granite'

ELSE get\_ubs\_match\_status(cpi.span\_number,cpi.bandwidth,mtt.bts\_type,t1\_type,non\_ubs\_type,cpi.parse\_status)

END

match\_status,

mtt.extract\_date,

cpi.rowid cpi\_rowid

FROM

moto\_t1\_terminations mtt

LEFT OUTER JOIN (

SELECT

\*

FROM

vzw\_moto\_voice\_cell\_paths cp

WHERE

cp.bts\_number IS NOT NULL

AND cp.cluster\_number IS NOT NULL

AND cp.span\_number IS NOT NULL

) cpi ON mtt.switch\_id = cpi.switch\_id

AND mtt.bts\_number = cpi.bts\_number

AND mtt.cluster\_number = cpi.cluster\_number

AND mtt.span\_number = cpi.span\_number

),moto\_ne\_only\_and\_ng AS (

SELECT

\*

FROM

moto\_ether\_audit ea

UNION

SELECT

\*

FROM

moto\_t1\_audit t1a

),xng\_only\_path\_id AS (

SELECT

ROWID cpi\_rowid

FROM

vzw\_moto\_voice\_cell\_paths cpi

MINUS

SELECT

cpi\_rowid

FROM

moto\_ne\_only\_and\_ng

),moto\_xng\_only AS (

SELECT

NULL ne\_omcr,

NULL ne\_bts\_number,

NULL ne\_cluster\_number,

NULL ne\_bts\_router\_group,

NULL ne\_bandwidth,

NULL ne\_span\_number,

NULL ne\_span\_type,

NULL ne\_access\_node\_number,

NULL ne\_ag\_node\_number,

NULL ne\_ag\_span\_number,

NULL ne\_t1\_line\_number,

NULL ne\_bts\_type,

NULL ne\_bts\_status,

CASE

WHEN cpi.bandwidth = 'DS1' THEN 'T1'

ELSE 'Ethernet'

END

termination\_type,

cpi.switch\_id,

bmm.vsm\_device\_name\_mtx mtx,

cpi.trail\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

NULL ng\_ipbscdo\_number,

cpi.bts\_number ng\_bts\_number,

cpi.cluster\_number ng\_cluster\_number,

cpi.span\_number ng\_span\_number,

cpi.bandwidth ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

'ng Only' match\_code,

parse\_status match\_status,

NULL extract\_date,

cpi.rowid cpi\_rowid

FROM

vzw\_moto\_voice\_cell\_paths cpi,

ng\_reports.bsm\_mtx\_map bmm,

xng\_only\_path\_id xopi

WHERE

cpi.switch\_id = ( substr(bmm.vsm\_device\_name\_mtx,1,6)

|| substr(bmm.vsm\_device\_name\_mtx,11,2) )

AND upper(bmm.mtx\_status) = 'LIVE'

AND cpi.rowid = xopi.cpi\_rowid

),full\_audit AS (

SELECT

\*

FROM

moto\_ne\_only\_and\_ng

UNION

SELECT

\*

FROM

moto\_xng\_only

) SELECT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_bts\_router\_group,

ne\_bandwidth,

ne\_span\_number,

ne\_span\_type,

ne\_access\_node\_number,

ne\_ag\_node\_number,

ne\_ag\_span\_number,

ne\_t1\_line\_number,

ne\_bts\_type,

ne\_bts\_status,

termination\_type,

switch\_id,

mtx,

ng\_path\_name,

ng\_path\_inst\_id,

ng\_ipbscdo\_number,

ng\_bts\_number,

ng\_cluster\_number,

ng\_span\_number,

ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

match\_code,

match\_status,

extract\_date,

trunc(SYSDATE) AS audit\_date

FROM

full\_audit fa;

COMMIT;

dbms\_output.put\_line('Motorola CDMA audit table created');

-- perform UBS Type audit

create\_ubs\_audit ();

-- remove the NE Only entry if BOTH exists in UBS path

-- due to multiple comparison for UBS

FOR cursorrec IN ne\_ethernet\_multiples LOOP

DELETE FROM moto\_cdma\_ne\_vs\_ng\_audit u

WHERE

u.match\_code LIKE 'NE%'

AND u.termination\_type LIKE 'Eth%'

AND u.ne\_omcr = cursorrec.ne\_omcr

AND u.ne\_bts\_number = cursorrec.ne\_bts\_number

AND u.ne\_cluster\_number = cursorrec.ne\_cluster\_number;

END LOOP;

COMMIT;

FOR cursorrec IN ne\_t1\_multiples LOOP

DELETE FROM moto\_cdma\_ne\_vs\_ng\_audit u

WHERE

u.match\_code LIKE 'NE%'

AND u.termination\_type = 'T1'

AND u.ne\_omcr = cursorrec.ne\_omcr

AND u.ne\_bts\_number = cursorrec.ne\_bts\_number

AND u.ne\_span\_number = cursorrec.ne\_span\_number

AND u.ne\_cluster\_number = cursorrec.ne\_cluster\_number;

END LOOP;

COMMIT;

-- check duplication

FOR cursorrec IN t1\_cdma\_multiples\_xng LOOP

UPDATE moto\_cdma\_ne\_vs\_ng\_audit

SET

match\_status = nvl2(match\_status,match\_status

|| ',Multiple spans found in Granite for same tokens','Multiple spans found in Granite for same tokens'),

match\_code = 'NE Only'

WHERE

ROWID = cursorrec.rowid;

END LOOP;

COMMIT;

FOR cursorrec IN ethernet\_cdma\_multiples\_xng LOOP

UPDATE moto\_cdma\_ne\_vs\_ng\_audit

SET

match\_status = nvl2(match\_status,match\_status

|| ',Multiple spans found in Granite for same tokens','Multiple spans found in Granite for same tokens'),

match\_code = 'NE Only'

WHERE

ROWID = cursorrec.rowid;

END LOOP;

COMMIT;

UPDATE moto\_cdma\_ne\_vs\_ng\_audit

SET

match\_code = 'NE Only'

WHERE

match\_code = 'BOTH'

AND match\_status IS NOT NULL;

COMMIT;

FOR cursorrec IN t1\_nl\_match LOOP

UPDATE moto\_cdma\_ne\_vs\_ng\_audit mnvxa

SET

match\_status = nvl2(match\_status,'Live Path Exists, '

|| match\_status,'Live Path Exists')

WHERE

mnvxa.ne\_omcr = cursorrec.ne\_omcr

AND mnvxa.ne\_bts\_number = cursorrec.ne\_bts\_number

AND mnvxa.ne\_cluster\_number = cursorrec.ne\_cluster\_number

AND mnvxa.ne\_span\_number = cursorrec.ne\_span\_number

AND mnvxa.termination\_type = 'T1'

AND upper(mnvxa.ng\_path\_status) <> 'LIVE';

END LOOP;

COMMIT;

FOR cursorrec IN ebh\_nl\_match LOOP

UPDATE moto\_cdma\_ne\_vs\_ng\_audit mnvxa

SET

match\_status = nvl2(match\_status,'Live Path Exists, '

|| match\_status,'Live Path Exists')

WHERE

mnvxa.ne\_omcr = cursorrec.ne\_omcr

AND mnvxa.ne\_bts\_number = cursorrec.ne\_bts\_number

AND mnvxa.ne\_cluster\_number = cursorrec.ne\_cluster\_number

AND mnvxa.termination\_type = 'Ethernet'

AND upper(mnvxa.ng\_path\_status) <> 'LIVE';

END LOOP;

COMMIT;

--JC

FOR cursorrec IN update\_mcdma\_wrk\_no\_csr\_port LOOP

UPDATE moto\_cdma\_ne\_vs\_ng\_audit mxa

SET

match\_code = 'NE Only',

match\_status =

CASE

WHEN match\_status = 'BOTH' THEN 'No CSR port on the e-pipe'

ELSE nvl2(match\_status,'No CSR port on the e-pipe, '

|| match\_status,'No CSR port on the e-pipe')

END

WHERE

mxa.ng\_path\_inst\_id = cursorrec.ng\_path\_inst\_id;

END LOOP;

COMMIT;

--JC

dbms\_output.put\_line('Motorola audit completd successfully');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error processing moto\_cdma\_audit(): '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END moto\_cdma\_audit;

PROCEDURE moto\_cdma\_regional\_summary AS

CURSOR updatecompliance IS

SELECT

summ.rowid rd,

summ.\*

FROM

moto\_cdma\_audit\_reg\_summ summ;

sqlstmt VARCHAR2(32767);

sqlstmtt1comp VARCHAR2(32767);

sqlstmtebhcomp VARCHAR2(32767);

sqlstmtovlcomp VARCHAR2(32767);

t1comp NUMBER;

ebhcomp NUMBER;

overallcomp NUMBER;

t1reported NUMBER;

ebhreported NUMBER;

BEGIN

sqlstmt := 'truncate table '

|| cdma\_region\_summ;

EXECUTE IMMEDIATE sqlstmt;

sqlstmt := 'INSERT INTO '

|| cdma\_region\_summ

|| '

(TERRITORY, SUB\_MARKET, T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE, EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,ebh\_matched\_live\_csr,

ebh\_matched\_nonlive\_csr, t1\_reported, ebh\_reported, t1\_comp, ebh\_comp, overall\_comp, ubs\_count, non\_ubs\_count)

WITH mtx\_region\_map AS

(SELECT DISTINCT cdmp.TERRITORY, cdmp.SUB\_MARKET, m.VSM\_DEVICE\_NAME\_MTX mtx, ( SUBSTR (VSM\_DEVICE\_NAME\_MTX, 1, 6) || SUBSTR (VSM\_DEVICE\_NAME\_MTX, 11, 2)) switch\_id

FROM NG\_REPORTS.clli\_domain\_map\_v cdmp

LEFT OUTER JOIN

(SELECT \* FROM NG\_REPORTS.bsm\_mtx\_map bmm WHERE bmm.VSM\_DEVICE\_NAME\_BSM LIKE ''%OMCR%'' AND upper(bmm.MTX\_STATUS) = ''LIVE'') m ON SUBSTR (m.VSM\_DEVICE\_NAME\_MTX, 1, 6) = SUBSTR (cdmp.CLLI, 1, 6)

WHERE cdmp.TERRITORY not in (''NNO'', ''OSS''))

SELECT mrm.TERRITORY, mrm.SUB\_MARKET, SUM (NVL (cds.t1\_discovered, 0)) AS t1\_discovered, SUM (NVL (cds.t1\_matched\_live, 0)) AS t1\_matched\_live,

SUM (NVL (cds.t1\_matched\_nonlive, 0)) AS t1\_matched\_nonlive, SUM (NVL (cds.ebh\_discovered, 0)) AS ebh\_discovered, SUM (NVL (cds.ebh\_matched\_live, 0)) AS ebh\_matched\_live,

SUM (NVL (cds.ebh\_matched\_nonlive, 0)) AS ebh\_matched\_nonlive, SUM (NVL (cds.ebh\_matched\_live\_csr, 0)) AS ebh\_matched\_live\_csr, SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)) AS ebh\_matched\_nonlive\_csr,

SUM (NVL (cds.t1\_discovered, 0)) - SUM (NVL (cds.t1\_matched\_nonlive, 0)) t1\_reported, SUM (NVL (cds.ebh\_discovered, 0)) - SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)) ebh\_reported,

case when SUM (NVL (cds.t1\_discovered, 0)) - SUM (NVL (cds.t1\_matched\_nonlive, 0)) > 0

then SUM (NVL (cds.t1\_matched\_live, 0)) / (SUM (NVL (cds.t1\_discovered, 0)) - SUM (NVL (cds.t1\_matched\_nonlive, 0)))\*100

else 0

end t1\_comp,

case when SUM (NVL (cds.ebh\_discovered, 0)) - SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)) > 0

then SUM (NVL (cds.ebh\_matched\_live\_csr, 0)) / (SUM (NVL (cds.ebh\_discovered, 0)) - SUM (NVL (cds.ebh\_matched\_nonlive\_csr, 0)))\*100

else 0

end ebh\_comp,

case when SUM(nvl(cds.t1\_discovered,0) + nvl(cds.ebh\_discovered,0))- SUM(NVL (cds.ebh\_matched\_nonlive\_csr, 0)+NVL (cds.t1\_matched\_nonlive, 0)) > 0

then ((sum (nvl(cds.t1\_matched\_live,0)+nvl(cds.ebh\_matched\_live\_csr,0)))/(SUM(nvl(cds.t1\_discovered,0) + nvl(cds.ebh\_discovered,0))- SUM(NVL (cds.ebh\_matched\_nonlive\_csr, 0)+NVL (cds.t1\_matched\_nonlive, 0))))\*100

else 0

end overall\_comp,

SUM (NVL (cds.ubs\_count, 0)) AS ubs\_count,

SUM (NVL (cds.non\_ubs\_count, 0)) AS non\_ubs\_count

FROM mtx\_region\_map mrm

LEFT OUTER JOIN '

|| cdma\_device\_summ

|| ' cds ON mrm.mtx = cds.VSM\_DEVICE\_NAME\_MTX

GROUP BY ROLLUP (mrm.TERRITORY, mrm.SUB\_MARKET)

ORDER BY mrm.TERRITORY, mrm.SUB\_MARKET'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

COMMIT;

sqlstmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is null and sub\_market is not null'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt := 'update MOTO\_CDMA\_AUDIT\_REG\_SUMM wk set wk.SUB\_MARKET=''Unknown'' where territory=''Unknown''';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlstmt);

dbms\_output.put\_line(substr('Error in populate\_cdma\_regional\_summary();: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END moto\_cdma\_regional\_summary;

PROCEDURE moto\_cdma\_device\_summary AS

CURSOR updatecompliance IS SELECT

summ.rowid rd,

summ.\*

FROM

moto\_cdma\_audit\_dev\_summ summ;

sqlstmt VARCHAR2(32767);

t1comp NUMBER;

ebhcomp NUMBER;

overallcomp NUMBER;

BEGIN

sqlstmt := 'truncate table '

|| cdma\_device\_summ;

EXECUTE IMMEDIATE sqlstmt;

INSERT INTO moto\_cdma\_audit\_dev\_summ (

territory,

sub\_market,

leaf\_domain,

vsm\_device\_name\_mtx,

switch\_id,

t1\_discovered,

t1\_matched\_live,

t1\_matched\_nonlive,

ebh\_discovered,

ebh\_matched\_live,

ebh\_matched\_nonlive,

ebh\_matched\_live\_csr,

ebh\_matched\_nonlive\_csr,

ubs\_count,

non\_ubs\_count

)

WITH mtx\_region\_map AS (

SELECT DISTINCT

cdmp.territory,

cdmp.sub\_market,

cdmp.leaf\_domain\_name,

m.vsm\_device\_name\_mtx mtx,

( substr(vsm\_device\_name\_mtx,1,6)

|| substr(vsm\_device\_name\_mtx,11,2) ) switch\_id

FROM

clli\_domain\_map\_v cdmp

RIGHT OUTER JOIN (

SELECT

\*

FROM

ng\_reports.bsm\_mtx\_map bmm

WHERE

bmm.vsm\_device\_name\_bsm LIKE '%OMCR'

AND UPPER(bmm.mtx\_status) = 'LIVE'

) m ON substr(m.vsm\_device\_name\_mtx,1,6) = substr(cdmp.clli,1,6)

WHERE

cdmp.territory NOT IN (

'NNO',

'OSS'

)

),audit\_details AS (

SELECT

\*

FROM

ng\_reports.moto\_cdma\_ne\_vs\_ng\_audit

WHERE

ne\_bts\_status IN (

'OOS',

'INS',

'INS\_ACTIVE'

)

AND match\_code IN (

'BOTH',

'NE Only'

)

),t1\_details AS (

SELECT

\*

FROM

audit\_details mnvxa

WHERE

mnvxa.termination\_type = 'T1'

),t1\_discovered AS (

SELECT

mtx,

COUNT(1) t1\_discovered

FROM

(

SELECT DISTINCT

mtx,

ne\_bts\_number,

ne\_cluster\_number,

ne\_span\_number

FROM

t1\_details

)

GROUP BY

mtx

),ebh\_details AS (

SELECT

\*

FROM

audit\_details mnvxa

WHERE

mnvxa.termination\_type = 'Ethernet'

),ethernet\_discovered AS (

SELECT

mtx,

COUNT(1) ethernet\_discovered

FROM

(

SELECT DISTINCT

mtx,

ne\_bts\_number,

ne\_cluster\_number

FROM

ebh\_details

)

GROUP BY

mtx

),t1\_matched\_live AS (

SELECT

mtx,

COUNT(1) t1\_matched\_live

FROM

(

SELECT DISTINCT

mtx,

ne\_bts\_number,

ne\_cluster\_number,

ne\_span\_number

FROM

t1\_details

WHERE

UPPER(ng\_path\_status) = 'LIVE'

--AND match\_code = 'BOTH'

)

GROUP BY

mtx

),t1\_matched\_nonlive AS (

SELECT

mtx,

COUNT(1) t1\_matched\_nonlive

FROM

(

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_cluster\_number,

mnvxa.ne\_span\_number

FROM

t1\_details mnvxa

WHERE

UPPER(mnvxa.ng\_path\_status) NOT LIKE 'LIVE'

--AND mnvxa.match\_code = 'BOTH'

MINUS

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_cluster\_number,

mnvxa.ne\_span\_number

FROM

t1\_details mnvxa

WHERE

UPPER(mnvxa.ng\_path\_status) = 'LIVE'

--AND mnvxa.match\_code = 'BOTH'

)

GROUP BY

mtx

),

-- ethernet\_matched\_live AS

-- (SELECT mtx, COUNT (1) ethernet\_matched\_live

-- FROM (SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

-- from ebh\_details mnvxa

-- WHERE MNVXA.NG\_PATH\_STATUS = 'Live'

-- AND mnvxa.MATCH\_CODE = 'BOTH')

-- GROUP BY mtx),

ethernet\_matched\_live\_csr AS (

SELECT

mtx,

COUNT(1) ethernet\_matched\_live\_csr

FROM

(

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_cluster\_number

FROM

ebh\_details mnvxa

join ng\_topology.trail\_element tre on mnvxa.ng\_path\_inst\_id = tre.trail\_id

WHERE

upper(mnvxa.ng\_path\_status) = 'LIVE'

AND mnvxa.match\_code = 'BOTH'

AND mnvxa.ng\_bandwidth LIKE '%bps'

AND tre.equipment\_type = 'CSR'

)

GROUP BY

mtx

),

-- ethernet\_matched\_nonlive AS

-- (SELECT MTX, COUNT (1) ETHERNET\_MATCHED\_NONLIVE

-- FROM (SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

-- FROM ebh\_details mnvxa

-- WHERE mnvxa.NG\_PATH\_STATUS <> 'Live'

-- AND mnvxa.MATCH\_CODE = 'BOTH'

-- MINUS

-- SELECT DISTINCT MNVXA.MTX, MNVXA.NE\_BTS\_NUMBER, MNVXA.NE\_CLUSTER\_NUMBER

-- FROM ebh\_details mnvxa

-- WHERE mnvxa.NG\_PATH\_STATUS = 'Live'

-- AND mnvxa.MATCH\_CODE = 'BOTH')

-- GROUP BY mtx),

ethernet\_matched\_nonlive\_csr AS (

SELECT

mtx,

COUNT(1) ethernet\_matched\_nonlive\_csr

FROM

(

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_cluster\_number

FROM

ebh\_details mnvxa

join ng\_topology.trail\_element tre on mnvxa.ng\_path\_inst\_id = tre.trail\_id

WHERE

UPPER(mnvxa.ng\_path\_status) <> 'LIVE'

AND mnvxa.match\_code = 'BOTH'

AND mnvxa.ng\_bandwidth LIKE '%bps'

AND tre.equipment\_type = 'CSR'

MINUS

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_cluster\_number

FROM

ebh\_details mnvxa

join ng\_topology.trail\_element tre on mnvxa.ng\_path\_inst\_id = tre.trail\_id

WHERE

UPPER(mnvxa.ng\_path\_status) = 'LIVE'

AND mnvxa.match\_code = 'BOTH'

AND mnvxa.ng\_bandwidth LIKE '%bps'

AND tre.equipment\_type = 'CSR'

)

GROUP BY

mtx

),conf\_both\_count AS -- for CONCONF BOTH

(

SELECT

mtx,

COUNT(1) conf\_both\_count

FROM

(

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_span\_number

FROM

t1\_details mnvxa

WHERE

mnvxa.ne\_bts\_type LIKE 'UB%'

AND mnvxa.ne\_cluster\_number IS NULL

AND match\_code = 'BOTH'

)

GROUP BY

mtx

),conf\_ne\_count AS -- for CONCONF NE only

(

SELECT

mtx,

COUNT(1) conf\_ne\_count

FROM

(

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_span\_number

FROM

t1\_details mnvxa

WHERE

mnvxa.ne\_bts\_type LIKE 'UB%'

AND mnvxa.ne\_cluster\_number IS NULL

AND match\_code LIKE 'NE%'

)

GROUP BY

mtx

),ubs\_count AS (

SELECT

mtx,

COUNT(1) ubs\_count

FROM

(

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_cluster\_number,

mnvxa.ne\_span\_number

FROM

audit\_details mnvxa

WHERE

mnvxa.ne\_bts\_type LIKE 'UB%'

)

GROUP BY

mtx

),non\_ubs\_count AS (

SELECT

mtx,

COUNT(1) non\_ubs\_count

FROM

(

SELECT DISTINCT

mnvxa.mtx,

mnvxa.ne\_bts\_number,

mnvxa.ne\_cluster\_number,

mnvxa.ne\_span\_number

FROM

audit\_details mnvxa

WHERE

mnvxa.ne\_bts\_type NOT LIKE 'UB%'

)

GROUP BY

mtx

) SELECT

mrm.territory,

mrm.sub\_market,

mrm.leaf\_domain\_name,

mrm.mtx,

mrm.switch\_id,

nvl(t1\_discovered,0) AS t1\_discovered,

nvl(t1\_matched\_live,0) AS t1\_matched\_live,

nvl(t1\_matched\_nonlive,0) AS t1\_matched\_nonlive,

nvl(ethernet\_discovered,0) AS ebh\_discovered,

nvl(conf\_both\_count,0) AS ebh\_matched\_live,

nvl(conf\_ne\_count,0) AS ebh\_matched\_nonlive,

nvl(ethernet\_matched\_live\_csr,0) AS ebh\_matched\_live\_csr,

nvl(ethernet\_matched\_nonlive\_csr,0) AS ebh\_matched\_nonlive\_csr,

nvl(ubs\_count,0) AS ubs\_count,

nvl(non\_ubs\_count,0) AS non\_ubs\_count

FROM

mtx\_region\_map mrm

LEFT OUTER JOIN t1\_discovered td ON mrm.mtx = td.mtx

LEFT OUTER JOIN t1\_matched\_live tm ON mrm.mtx = tm.mtx

LEFT OUTER JOIN t1\_matched\_nonlive tum ON mrm.mtx = tum.mtx

LEFT OUTER JOIN ethernet\_discovered ed ON mrm.mtx = ed.mtx

LEFT OUTER JOIN conf\_both\_count em ON mrm.mtx = em.mtx

LEFT OUTER JOIN conf\_ne\_count eum ON mrm.mtx = eum.mtx

LEFT OUTER JOIN ethernet\_matched\_live\_csr emc ON mrm.mtx = emc.mtx

LEFT OUTER JOIN ethernet\_matched\_nonlive\_csr eumc ON mrm.mtx = eumc.mtx

LEFT OUTER JOIN ubs\_count ubs ON mrm.mtx = ubs.mtx

LEFT OUTER JOIN non\_ubs\_count nubs ON mrm.mtx = nubs.mtx

ORDER BY

mrm.mtx;

COMMIT;

sqlstmt := 'update '

|| cdma\_device\_summ

|| ' set t1\_comp = :t, ebh\_comp = :e, overall\_comp = :o where rowid = :a';

FOR cur IN updatecompliance LOOP

t1comp := nvl(cur.t1\_discovered,0) - nvl(cur.t1\_matched\_nonlive,0);

ebhcomp := nvl(cur.ebh\_discovered,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

overallcomp := nvl(cur.t1\_discovered,0) + nvl(cur.ebh\_discovered,0);

overallcomp := overallcomp - nvl(cur.t1\_matched\_nonlive,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

IF

( t1comp > 0 )

THEN

t1comp := ( nvl(cur.t1\_matched\_live,0) / t1comp ) \* 100;

END IF;

IF

( ebhcomp > 0 )

THEN

ebhcomp := ( nvl(cur.ebh\_matched\_live\_csr,0) / ebhcomp ) \* 100;

END IF;

IF

( overallcomp > 0 )

THEN

overallcomp := ( ( nvl(cur.t1\_matched\_live,0) + nvl(cur.ebh\_matched\_live\_csr,0) ) / overallcomp ) \* 100;

--dbms\_output.put\_line('good EBH '||overallComp);

END IF;

EXECUTE IMMEDIATE sqlstmt

USING t1comp,ebhcomp,overallcomp,cur.rd;

END LOOP;

COMMIT;

moto\_cdma\_regional\_summary ();

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlstmt);

dbms\_output.put\_line(substr('Error in moto\_cdma\_device\_summary();: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END moto\_cdma\_device\_summary;

PROCEDURE create\_ubs\_audit

AS

BEGIN

INSERT INTO moto\_cdma\_ne\_vs\_ng\_audit (

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_bts\_router\_group,

ne\_bandwidth,

ne\_span\_number,

ne\_span\_type,

ne\_access\_node\_number,

ne\_ag\_node\_number,

ne\_ag\_span\_number,

ne\_t1\_line\_number,

ne\_bts\_type,

ne\_bts\_status,

termination\_type,

switch\_id,

mtx,

ng\_path\_name,

ng\_path\_inst\_id,

ng\_ipbscdo\_number,

ng\_bts\_number,

ng\_cluster\_number,

ng\_span\_number,

ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

match\_code,

match\_status,

extract\_date,

audit\_date

)

WITH bts\_status AS (

SELECT DISTINCT

\*

FROM

mt\_omcr\_bts\_status stat

WHERE

stat.bts\_mlppp\_number IS NULL

AND stat.pbts\_number IS NULL

),moto\_ether\_terminations AS (

SELECT

inv.vsm\_device\_name\_omcr,

inv.bts\_number,

inv.cluster\_number,

inv.router\_grp,

inv.bandwidth,

inv.extract\_date,

substr(bmm.vsm\_device\_name\_mtx,1,6)

|| substr(bmm.vsm\_device\_name\_mtx,11,2) switch\_id,

'Ethernet' termination\_type,

bmm.vsm\_device\_name\_mtx,

mp.bts\_type,

stat.tel\_state ne\_bts\_status

FROM

moto\_cdma\_ethernet\_inv inv

JOIN bsm\_mtx\_map bmm ON inv.vsm\_device\_name\_omcr = bmm.vsm\_device\_name\_bsm

AND upper(bmm.mtx\_status) = 'LIVE'

AND upper(bmm.bsm\_status) = 'LIVE'

JOIN moto\_bts\_location\_map mp ON inv.vsm\_device\_name\_omcr = mp.vsm\_device\_name\_omcr

AND inv.bts\_number = mp.bts\_number

JOIN bts\_status stat ON inv.vsm\_device\_name\_omcr = stat.vsm\_device\_name\_omcr

AND inv.bts\_number = stat.bts\_number

),moto\_t1\_terminations AS (

SELECT

mcs.bts\_number,

mcs.cluster\_number,

mcs.bts\_router\_group,

mcs.span\_number,

mcs.access\_node\_number,

mcs.ag\_node\_number,

mcs.ag\_span\_number,

mcs.t1\_line\_number,

mcs.extract\_date,

mcs.vsm\_device\_name\_omcr,

substr(bmm.vsm\_device\_name\_mtx,1,6)

|| substr(bmm.vsm\_device\_name\_mtx,11,2) switch\_id,

'T1' termination\_type,

bmm.vsm\_device\_name\_mtx,

mp.bts\_type,

stat.tel\_state ne\_bts\_status

FROM

moto\_cdma\_t1\_inv mcs

JOIN bsm\_mtx\_map bmm ON mcs.vsm\_device\_name\_omcr = bmm.vsm\_device\_name\_bsm

AND upper(bmm.mtx\_status) = 'LIVE'

AND upper(bmm.bsm\_status) = 'LIVE'

JOIN moto\_bts\_location\_map mp ON mcs.vsm\_device\_name\_omcr = mp.vsm\_device\_name\_omcr

AND mcs.bts\_number = mp.bts\_number

JOIN bts\_status stat ON mcs.vsm\_device\_name\_omcr = stat.vsm\_device\_name\_omcr

AND mcs.bts\_number = stat.bts\_number

),evdo\_t1 AS (

SELECT

substr(bmm.vsm\_device\_name\_mtx,1,6)

|| substr(bmm.vsm\_device\_name\_mtx,11,2) switch\_id,

mtt.vsm\_device\_name\_omcr,

mtt.bts\_number ne\_bts\_number,

nvl(mtt.bts\_con\_number,mtt.pbts\_span\_number) ne\_span\_number,

mtt.access\_node\_number ne\_access\_node\_number,

mtt.ag\_node\_number ne\_ag\_node\_number,

mtt.ag\_span\_number ne\_ag\_span\_number,

mtt.t1\_line\_number ne\_t1\_line\_number,

bmm.vsm\_device\_name\_mtx,

mtt.extract\_date,

mp.bts\_type,

stat.tel\_state bts\_status

FROM

mt\_omcr\_bts\_conconf mtt

JOIN bts\_status stat ON mtt.vsm\_device\_name\_omcr = stat.vsm\_device\_name\_omcr

AND mtt.bts\_number = stat.bts\_number

JOIN bsm\_mtx\_map bmm ON mtt.vsm\_device\_name\_omcr = bmm.vsm\_device\_name\_bsm

AND upper(bmm.mtx\_status) = 'LIVE'

AND upper(bmm.bsm\_status) = 'LIVE'

JOIN moto\_bts\_location\_map mp ON mtt.vsm\_device\_name\_omcr = mp.vsm\_device\_name\_omcr

AND mtt.bts\_number = mp.bts\_number

),evdo\_t1\_audit AS (

SELECT

mtt.vsm\_device\_name\_omcr ne\_omcr,

mtt.ne\_bts\_number,

NULL ne\_cluster\_number,

NULL ne\_bts\_router\_group,

NULL ne\_bandwidth,

mtt.ne\_span\_number,

'DATA' ne\_span\_type,

mtt.ne\_access\_node\_number,

mtt.ne\_ag\_node\_number,

mtt.ne\_ag\_span\_number,

mtt.ne\_t1\_line\_number,

mtt.bts\_type ne\_bts\_type,

mtt.bts\_status ne\_bts\_status,

'T1' termination\_type,

mtt.switch\_id,

mtt.vsm\_device\_name\_mtx mtx,

cpi.trail\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

cpi.ipbscdo\_number ng\_ipbscdo\_number,

cpi.bts\_number ng\_bts\_number,

NULL ng\_cluster\_number,

cpi.span\_number ng\_span\_number,

cpi.bandwidth ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

CASE

WHEN cpi.switch\_id IS NULL

OR ( mtt.switch\_id <> cpi.switch\_id ) THEN 'NE Only'

ELSE 'BOTH'

END

match\_code,

CASE

WHEN cpi.switch\_id IS NULL THEN 'Not found in Granite'

ELSE get\_ubs\_match\_status(cpi.span\_number,cpi.bandwidth,mtt.bts\_type,t1\_type,ubs\_type,cpi.parse\_status)

END

match\_status,

mtt.extract\_date,

cpi.rowid cpi\_rowid

FROM

evdo\_t1 mtt

LEFT OUTER JOIN vzw\_moto\_ubs\_cell\_paths cpi ON mtt.ne\_bts\_number = cpi.bts\_number

AND mtt.ne\_span\_number = cpi.span\_number

AND mtt.switch\_id = cpi.switch\_id

),ubs\_t1\_audit AS (

SELECT

mtt.vsm\_device\_name\_omcr ne\_omcr,

mtt.bts\_number ne\_bts\_number,

mtt.cluster\_number ne\_cluster\_number,

mtt.bts\_router\_group ne\_bts\_router\_group,

NULL ne\_bandwidth,

mtt.span\_number ne\_span\_number,

'VOICE' ne\_span\_type,

mtt.access\_node\_number ne\_access\_node\_number,

mtt.ag\_node\_number ne\_ag\_node\_number,

mtt.ag\_span\_number ne\_ag\_span\_number,

mtt.t1\_line\_number ne\_t1\_line\_number,

mtt.bts\_type ne\_bts\_type,

ne\_bts\_status,

mtt.termination\_type,

mtt.switch\_id,

mtt.vsm\_device\_name\_mtx mtx,

cpi.trail\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

cpi.ipbscdo\_number ng\_ipbscdo\_number,

cpi.bts\_number ng\_bts\_number,

NULL ng\_cluster\_number,

cpi.span\_number ng\_span\_number,

cpi.bandwidth ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

CASE

WHEN cpi.switch\_id IS NULL THEN 'NE Only'

ELSE 'BOTH'

END

match\_code,

CASE

WHEN cpi.switch\_id IS NULL THEN 'Not found in Granite'

ELSE get\_ubs\_match\_status(cpi.span\_number,cpi.bandwidth,mtt.bts\_type,t1\_type,ubs\_type,cpi.parse\_status)

END

match\_status,

mtt.extract\_date,

cpi.rowid cpi\_rowid

FROM

moto\_t1\_terminations mtt

JOIN vzw\_moto\_ubs\_cell\_paths cpi ON mtt.switch\_id = cpi.switch\_id

AND mtt.bts\_number = cpi.bts\_number

AND mtt.span\_number = cpi.span\_number

),ubs\_ether\_audit AS (

SELECT

met.vsm\_device\_name\_omcr ne\_omcr,

met.bts\_number ne\_bts\_number,

met.cluster\_number ne\_cluster\_number,

met.router\_grp ne\_bts\_router\_group,

met.bandwidth ne\_bandwidth,

NULL ne\_span\_number,

NULL ne\_span\_type,

NULL ne\_access\_node\_number,

NULL ne\_ag\_node\_number,

NULL ne\_ag\_span\_number,

NULL ne\_t1\_line\_number,

met.bts\_type ne\_bts\_type,

met.ne\_bts\_status,

met.termination\_type,

met.switch\_id,

met.vsm\_device\_name\_mtx mtx,

cpi.trail\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

cpi.ipbscdo\_number ng\_ipbscdo\_number,

cpi.bts\_number ng\_bts\_number,

NULL ng\_cluster\_number,

cpi.span\_number ng\_span\_number,

cpi.bandwidth ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

CASE

WHEN cpi.switch\_id IS NULL THEN 'NE Only'

WHEN met.bts\_type IS NULL

OR substr(met.bts\_type,1,2) <> 'UB' THEN 'NE Only'

ELSE 'BOTH'

END

match\_code,

CASE

WHEN cpi.switch\_id IS NULL THEN 'Not found in Nautilus'

ELSE get\_ubs\_match\_status(cpi.span\_number,cpi.bandwidth,met.bts\_type,ebh\_type,ubs\_type,cpi.parse\_status)

END

match\_status,

met.extract\_date,

cpi.rowid cpi\_rowid

FROM

moto\_ether\_terminations met

JOIN vzw\_moto\_ubs\_cell\_paths cpi ON met.switch\_id = cpi.switch\_id

AND met.bts\_number = cpi.bts\_number

),ubs\_ne\_only\_and\_both AS (

SELECT

\*

FROM

ubs\_ether\_audit

UNION

SELECT

\*

FROM

ubs\_t1\_audit

UNION

SELECT

\*

FROM

evdo\_t1\_audit

),ubs\_ng\_only\_path\_id AS (

SELECT

ROWID cpi\_rowid

FROM

vzw\_moto\_ubs\_cell\_paths cpi

MINUS

SELECT

cpi\_rowid

FROM

ubs\_ne\_only\_and\_both

),ubs\_ng\_only AS (

SELECT

NULL ne\_omcr,

NULL ne\_bts\_number,

NULL ne\_cluster\_number,

NULL ne\_bts\_router\_group,

NULL ne\_bandwidth,

NULL ne\_span\_number,

NULL ne\_span\_type,

NULL ne\_access\_node\_number,

NULL ne\_ag\_node\_number,

NULL ne\_ag\_span\_number,

NULL ne\_t1\_line\_number,

NULL ne\_bts\_type,

NULL ne\_bts\_status,

CASE

WHEN cpi.bandwidth = 'DS1' THEN 'T1'

ELSE 'Ethernet'

END

termination\_type,

cpi.switch\_id,

bmm.vsm\_device\_name\_mtx mtx,

cpi.trail\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

cpi.ipbscdo\_number ng\_ipbscdo\_number,

cpi.bts\_number ng\_bts\_number,

NULL ng\_cluster\_number,

cpi.span\_number ng\_span\_number,

cpi.bandwidth ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

'Xng Only' match\_code,

parse\_status match\_status,

NULL extract\_date,

cpi.rowid cpi\_rowid

FROM

vzw\_moto\_ubs\_cell\_paths cpi

JOIN ubs\_ng\_only\_path\_id xopi ON cpi.rowid = xopi.cpi\_rowid

JOIN ng\_reports.bsm\_mtx\_map bmm ON cpi.switch\_id = substr(bmm.vsm\_device\_name\_mtx,1,6)

|| substr(bmm.vsm\_device\_name\_mtx,11,2)

AND upper(bmm.mtx\_status) = 'LIVE'

WHERE

cpi.ng\_path\_type <> 'EVDO'

OR (

cpi.ng\_path\_type = 'EVDO'

AND cpi.bandwidth = 'DS1'

)

),full\_audit AS (

SELECT

\*

FROM

ubs\_ne\_only\_and\_both unb

UNION

SELECT

\*

FROM

ubs\_ng\_only ux

) SELECT

ne\_omcr,

ne\_bts\_number,

ne\_cluster\_number,

ne\_bts\_router\_group,

ne\_bandwidth,

ne\_span\_number,

ne\_span\_type,

ne\_access\_node\_number,

ne\_ag\_node\_number,

ne\_ag\_span\_number,

ne\_t1\_line\_number,

ne\_bts\_type,

ne\_bts\_status,

termination\_type,

switch\_id,

mtx,

ng\_path\_name,

ng\_path\_inst\_id,

ng\_ipbscdo\_number,

ng\_bts\_number,

ng\_cluster\_number,

ng\_span\_number,

ng\_bandwidth,

ng\_path\_status,

ng\_path\_type,

match\_code,

match\_status,

extract\_date,

trunc(SYSDATE) AS audit\_date

FROM

full\_audit fa;

COMMIT;

dbms\_output.put\_line('create\_ubs\_audit() completd successfully');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error in create\_ubs\_audit(): '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END create\_ubs\_audit;

PROCEDURE moto\_evdo\_audit AS

-- One Live and N non Live match

CURSOR t1\_nl\_match IS SELECT

aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier,

ne\_span\_number

FROM

moto\_evdo\_ne\_vs\_ng\_audit

WHERE

match\_code = 'BOTH'

AND UPPER(ng\_path\_status) = 'LIVE'

AND termination\_type = 'T1';

-- One Live and N non Live match

CURSOR ebh\_nl\_match IS SELECT

aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier

FROM

moto\_evdo\_ne\_vs\_ng\_audit

WHERE

match\_code = 'BOTH'

AND UPPER(ng\_path\_status) = 'LIVE'

AND termination\_type = 'Ethernet';

-- T1 spans matching multiple live paths in ng or matching multiple non live paths in ng incase a live match does not exist are marked NE Only

CURSOR t1\_evdo\_multiples\_ng IS WITH dup AS (

SELECT DISTINCT

aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier,

ne\_span\_number,

COUNT(

CASE

WHEN UPPER(ng\_path\_status) = 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier,

ne\_span\_number

) AS live\_count,

COUNT(

CASE

WHEN UPPER(ng\_path\_status) != 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier,

ne\_span\_number

) AS non\_live\_count

FROM

moto\_evdo\_ne\_vs\_ng\_audit src

WHERE

src.match\_code = 'BOTH'

AND src.termination\_type = 'T1'

) SELECT

tgt.rowid

FROM

moto\_evdo\_ne\_vs\_ng\_audit tgt

JOIN dup ON dup.aems = tgt.aems

AND dup.ne\_ipbscdo\_number = tgt.ne\_ipbscdo\_number

AND dup.ne\_mccdo\_identifier = tgt.ne\_mccdo\_identifier

AND dup.ne\_span\_number = tgt.ne\_span\_number

AND (

dup.live\_count > 1

OR (

dup.live\_count = 0

AND dup.non\_live\_count > 1

)

);

-- Ethernet spans matching multiple live paths in ng or matching multiple non live paths in ng incase a live match does not exist are marked NE Only

CURSOR ethernet\_evdo\_multiples\_ng IS WITH dup AS (

SELECT DISTINCT

aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier,

ne\_span\_number,

COUNT(

CASE

WHEN UPPER(ng\_path\_status) = 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier

) AS live\_count,

COUNT(

CASE

WHEN UPPER(ng\_path\_status) != 'LIVE' THEN 1

ELSE NULL

END

) OVER(

PARTITION BY aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier

) AS non\_live\_count

FROM

moto\_evdo\_ne\_vs\_ng\_audit src

WHERE

src.match\_code = 'BOTH'

AND src.termination\_type = 'Ethernet'

) SELECT

tgt.rowid

FROM

moto\_evdo\_ne\_vs\_ng\_audit tgt

JOIN dup ON dup.aems = tgt.aems

AND dup.ne\_ipbscdo\_number = tgt.ne\_ipbscdo\_number

AND dup.ne\_mccdo\_identifier = tgt.ne\_mccdo\_identifier

AND (

dup.live\_count > 1

OR (

dup.live\_count = 0

AND dup.non\_live\_count > 1

)

);

--JC

-- Update those paths that dont have a CSR port

CURSOR update\_mevdo\_wrk\_no\_csr\_port IS WITH mxa\_paths AS (

SELECT

mxa.ng\_path\_inst\_id

FROM

moto\_evdo\_ne\_vs\_ng\_audit mxa

WHERE

mxa.termination\_type = 'Ethernet'

AND mxa.ng\_bandwidth LIKE '%bps'

AND mxa.match\_code = 'BOTH'

AND UPPER(mxa.ng\_path\_status) = 'LIVE'

),paths\_w\_csr AS (

SELECT

cpi.ng\_path\_inst\_id

FROM

mxa\_paths cpi

JOIN ng\_topology.trail\_element cpe ON cpi.ng\_path\_inst\_id = cpe.trail\_id

where cpe.equipment\_type = 'CSR'

) SELECT

mxap.ng\_path\_inst\_id

FROM

mxa\_paths mxap

MINUS

SELECT

pc.ng\_path\_inst\_id

FROM

paths\_w\_csr pc;

sql\_stmt VARCHAR2(32000);

BEGIN

sql\_stmt := 'TRUNCATE TABLE moto\_evdo\_ne\_vs\_ng\_audit';

BEGIN

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(substr('Error in XngVsNortel();: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

COMMIT;

sql\_stmt := '

INSERT INTO moto\_evdo\_ne\_vs\_ng\_audit

(aems,

ne\_ipbscdo\_number,

ne\_mccdo\_identifier,

ne\_span\_number,

ne\_status,

termination\_type,

match\_code,

match\_status,

ng\_path\_name,

ng\_path\_inst\_id,

ng\_ipbscdo\_number,

xng\_mccdo\_identifier,

ng\_span\_number,

ng\_path\_type,

ng\_path\_status,

ng\_bandwidth,

extract\_date,

audit\_date)

WITH ethernet\_terminations AS

(SELECT met.vsm\_device\_name\_aems aems,

met.ipbscdo\_number ne\_ipbscdo\_number,

met.mccdo\_number ne\_mccdo\_identifier, met.extract\_date,

met.ROWID ne\_rowid

FROM moto\_evdo\_ethernet\_inv met, moto\_vsm\_name\_aems mvn

where met.vsm\_device\_name\_aems = mvn.vsm\_device\_name\_aems

AND met.ipbscdo\_number = mvn.ipbscdo\_number

AND upper(mvn.status) = ''LIVE''),

ethernet\_audit AS

(SELECT met.aems, met.ne\_ipbscdo\_number, met.ne\_mccdo\_identifier,

NULL ne\_span\_number, NULL ne\_status, ''Ethernet'' termination\_type,

CASE

WHEN cpi.ipbscdo\_number IS NULL

THEN ''NE Only''

ELSE ''BOTH''

END match\_code,

CASE

WHEN cpi.span\_number IS NOT NULL

AND UPPER (cpi.bandwidth) LIKE ''%DS1%''

THEN ''Pathname Invalid Format, Path Invalid Bandwidth''

WHEN cpi.span\_number IS NOT NULL

AND UPPER (cpi.bandwidth) NOT LIKE ''%DS1%''

THEN ''Pathname Invalid Format''

WHEN cpi.span\_number IS NULL

AND UPPER (cpi.bandwidth) LIKE ''%DS1%''

THEN ''Path Invalid Bandwidth''

ELSE parse\_status

END match\_status,

cpi.path\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

cpi.ipbscdo\_number ng\_ipbscdo\_number,

cpi.mccdo\_number xng\_mccdo\_identifier,

cpi.span\_number ng\_span\_number, cpi.ng\_path\_type,

cpi.ng\_path\_status, cpi.bandwidth ng\_bandwidth,

met.extract\_date, cpi.ROWID cpi\_rowid, met.ne\_rowid

FROM ethernet\_terminations met LEFT OUTER JOIN vzw\_moto\_evdo\_cell\_paths cpi

ON met.ne\_ipbscdo\_number = cpi.ipbscdo\_number

AND met.ne\_mccdo\_identifier = cpi.mccdo\_number

),

t1\_terminations AS

(SELECT mtt.vsm\_device\_name\_aems aems,

mtt.ipbscdo\_number ne\_ipbscdo\_number,

mtt.mccdo\_number ne\_mccdo\_identifier,

mtt.span\_number ne\_span\_number, mtt.status ne\_status,

mtt.extract\_date, mtt.ROWID ne\_rowid

FROM moto\_evdo\_t1\_inv mtt, moto\_vsm\_name\_aems mvn

WHERE mtt.vsm\_device\_name\_aems = mvn.vsm\_device\_name\_aems

AND mtt.ipbscdo\_number = mvn.ipbscdo\_number

AND upper(mvn.status) = ''LIVE''),

t1\_audit AS

(SELECT mtt.aems, mtt.ne\_ipbscdo\_number, mtt.ne\_mccdo\_identifier,

mtt.ne\_span\_number, mtt.ne\_status, ''T1'' termination\_type,

CASE

WHEN cpi.ipbscdo\_number IS NULL

THEN ''NE Only''

ELSE ''BOTH''

END match\_code,

CASE

WHEN UPPER (cpi.bandwidth) NOT LIKE ''%DS1%''

THEN ''Path Invalid Bandwidth''

ELSE parse\_status

END match\_status,

cpi.path\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

cpi.ipbscdo\_number ng\_ipbscdo\_number,

cpi.mccdo\_number xng\_mccdo\_identifier,

cpi.span\_number ng\_span\_number, cpi.ng\_path\_type,

cpi.ng\_path\_status, cpi.bandwidth ng\_bandwidth,

mtt.extract\_date, cpi.ROWID cpi\_rowid, mtt.ne\_rowid

FROM t1\_terminations mtt LEFT OUTER JOIN vzw\_moto\_evdo\_cell\_paths cpi

ON mtt.ne\_ipbscdo\_number = cpi.ipbscdo\_number

AND mtt.ne\_mccdo\_identifier = cpi.mccdo\_number

AND mtt.ne\_span\_number = cpi.span\_number

),

ne\_only\_and\_ng AS

(SELECT \* FROM ethernet\_audit ea

UNION

SELECT \* FROM t1\_audit t1a),

ng\_only\_path\_id AS

(SELECT ROWID cpi\_rowid FROM vzw\_moto\_evdo\_cell\_paths cpi

MINUS

SELECT cpi\_rowid FROM ne\_only\_and\_ng),

ng\_only AS

(SELECT mvn.vsm\_device\_name\_aems aems, NULL ne\_ipbscdo\_number, NULL ne\_mccdo\_identifier,

NULL ne\_status, NULL ne\_span\_number,

CASE

WHEN cpi.bandwidth = ''DS1''

THEN ''T1''

ELSE ''Ethernet''

END termination\_type,

''Xng Only'' match\_code, parse\_status match\_status,

cpi.path\_name ng\_path\_name,

cpi.trail\_id ng\_path\_inst\_id,

cpi.ipbscdo\_number ng\_ipbscdo\_number,

cpi.mccdo\_number xng\_mccdo\_identifier,

cpi.span\_number ng\_span\_number, cpi.ng\_path\_type,

cpi.ng\_path\_status, cpi.bandwidth ng\_bandwidth,

NULL extract\_date, cpi.ROWID cpi\_rowid, NULL ne\_rowid

FROM vzw\_moto\_evdo\_cell\_paths cpi, ng\_only\_path\_id xopi,moto\_vsm\_name\_aems mvn

WHERE cpi.ROWID = xopi.cpi\_rowid

and cpi.ipbscdo\_number=mvn.ipbscdo\_number),

full\_audit AS

(SELECT \*

FROM ne\_only\_and\_ng

UNION

SELECT \*

FROM ng\_only)

SELECT aems, ne\_ipbscdo\_number, ne\_mccdo\_identifier, ne\_span\_number,

ne\_status, termination\_type, match\_code, match\_status,ng\_path\_name,ng\_path\_inst\_id,

ng\_ipbscdo\_number, xng\_mccdo\_identifier, ng\_span\_number,

ng\_path\_type, ng\_path\_status, ng\_bandwidth, extract\_date,

TRUNC (SYSDATE) AS audit\_date

FROM full\_audit fa'

;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

dbms\_output.put\_line('Moto EVDO audit table created');

FOR cursorrec IN t1\_evdo\_multiples\_ng LOOP

UPDATE moto\_evdo\_ne\_vs\_ng\_audit

SET

match\_status = nvl2(match\_status,match\_status

|| ',Multiple spans found in Granite same tokens','Multiple spans found in Granite for same tokens'),

match\_code = 'NE Only'

WHERE

ROWID = cursorrec.rowid;

END LOOP;

COMMIT;

FOR cursorrec IN ethernet\_evdo\_multiples\_ng LOOP

UPDATE moto\_evdo\_ne\_vs\_ng\_audit

SET

match\_status = nvl2(match\_status,match\_status

|| ',Multiple spans found in Granite for same tokens','Multiple spans found in Granite for same tokens'),

match\_code = 'NE Only'

WHERE

ROWID = cursorrec.rowid

;

END LOOP;

COMMIT;

UPDATE moto\_evdo\_ne\_vs\_ng\_audit

SET

match\_code = 'NE Only'

WHERE

match\_code = 'BOTH'

AND match\_status IS NOT NULL;

COMMIT;

FOR cursorrec IN t1\_nl\_match LOOP

UPDATE moto\_evdo\_ne\_vs\_ng\_audit mnvxa

SET

match\_status = nvl2(match\_status,'Live Path Exists,'

|| match\_status,'Live Path Exists')

WHERE

mnvxa.aems = cursorrec.aems

AND mnvxa.ne\_ipbscdo\_number = cursorrec.ne\_ipbscdo\_number

AND mnvxa.ne\_mccdo\_identifier = cursorrec.ne\_mccdo\_identifier

AND mnvxa.ne\_span\_number = cursorrec.ne\_span\_number

AND mnvxa.termination\_type = 'T1'

AND upper(mnvxa.ng\_path\_status) <> 'LIVE';

END LOOP;

COMMIT;

FOR cursorrec IN ebh\_nl\_match LOOP

UPDATE moto\_evdo\_ne\_vs\_ng\_audit mnvxa

SET

match\_status = nvl2(match\_status,'Live Path Exists,'

|| match\_status,'Live Path Exists')

WHERE

mnvxa.aems = cursorrec.aems

AND mnvxa.ne\_ipbscdo\_number = cursorrec.ne\_ipbscdo\_number

AND mnvxa.ne\_mccdo\_identifier = cursorrec.ne\_mccdo\_identifier

AND mnvxa.termination\_type = 'Ethernet'

AND upper(mnvxa.ng\_path\_status) <> 'LIVE';

END LOOP;

dbms\_output.put\_line('Moto ESdit table created');

COMMIT;

--JC

FOR cursorrec IN update\_mevdo\_wrk\_no\_csr\_port LOOP

UPDATE moto\_evdo\_ne\_vs\_ng\_audit mxa

SET

match\_code = 'NE Only',

match\_status =

CASE

WHEN match\_status = 'BOTH' THEN 'No CSR port on the e-pipe'

ELSE nvl2(match\_status,'No CSR port on the e-pipe, '

|| match\_status,'No CSR port on the e-pipe')

END

WHERE

mxa.ng\_path\_inst\_id = cursorrec.ng\_path\_inst\_id;

END LOOP;

dbms\_output.put\_line('Moto EVDO audit tabVDZGFDSle created');

COMMIT;

--JC

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(substr('Error in moto\_evdo\_audit;: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END moto\_evdo\_audit;

PROCEDURE moto\_evdo\_regional\_summary AS

CURSOR updateCompliance IS

SELECT

summ.rowid

rd,

summ.\* FROM moto\_evdo\_audit\_reg\_summ summ;

sqlStmt VARCHAR2(32767);

sqlStmtT1Comp VARCHAR2(32767);

sqlStmtEbhComp VARCHAR2(32767);

sqlStmtOvlComp VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

t1Reported NUMBER;

ebhReported NUMBER;

BEGIN

dbms\_output.put\_line('here 0');

sqlStmt := 'truncate table moto\_evdo\_audit\_reg\_summ';

execute IMMEDIATE sqlStmt;

sqlStmt := 'INSERT INTO moto\_evdo\_audit\_reg\_summ

(TERRITORY, SUB\_MARKET,T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,EBH\_MATCHED\_LIVE\_CSR,EBH\_MATCHED\_NONLIVE\_CSR)

/\* Formatted on 2011/03/03 13:08 (Formatter Plus v4.8.8) \*/

WITH AEMS\_REGION\_MAP AS(

SELECT DISTINCT

cdmp.territory,

cdmp.sub\_market,

cdmp.leaf\_domain\_name,

mvn.vsm\_device\_name\_aems aems,

mvn.ipbscdo\_number ne\_ipbscdo\_number

FROM

ng\_reports.clli\_domain\_map\_v cdmp

LEFT OUTER JOIN moto\_vsm\_name\_aems mvn ON substr(mvn.vsm\_device\_name\_aems,1,6) = substr(cdmp.clli,1,6)

WHERE

cdmp.territory <> ''NNO''

),t1\_discovered AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) t1\_discovered

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND mea.ne\_status IN (

''INS'',

''OOS''

)

AND mea.match\_code IN (

''BOTH'',

''NE Only''

)

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_discovered AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_discovered

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND mea.match\_code IN (

''BOTH'',

''NE Only''

)

)

GROUP BY

ne\_ipbscdo\_number

),t1\_matched\_live AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) t1\_matched\_live

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),t1\_matched\_nonlive AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) t1\_matched\_nonlive

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND upper(mea.ng\_path\_status) <> ''LIVE''

AND mea.match\_code = ''BOTH''

MINUS

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_live AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_live

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_live\_csr AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_live\_csr

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

JOIN ng\_topology.trail\_element cpe ON mea.ng\_path\_inst\_id = cpe.trail\_id

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

AND mea.ng\_bandwidth LIKE ''%bps''

AND cpe.equipment\_type = ''CSR''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_nonlive AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_nonlive

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) <> ''LIVE''

AND mea.match\_code = ''BOTH''

MINUS

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_nonlive\_csr AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_nonlive\_csr

FROM

(

( SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

JOIN ng\_topology.trail\_element cpe ON mea.ng\_path\_inst\_id = cpe.trail\_id

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) <> ''LIVE''

AND mea.match\_code = ''BOTH''

AND mea.ng\_bandwidth LIKE ''%bps''

AND cpe.equipment\_type = ''CSR''

)

MINUS

( SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

JOIN ng\_topology.trail\_element cpe ON mea.ng\_path\_inst\_id = cpe.trail\_id

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

AND mea.ng\_bandwidth LIKE ''%bps''

AND cpe.equipment\_type = ''CSR''

)

)

GROUP BY

ne\_ipbscdo\_number

) SELECT

mrm.territory,

mrm.sub\_market,

SUM(nvl(t1\_discovered,0) ) AS t1\_discovered,

SUM(nvl(t1\_matched\_live,0) ) AS t1\_matched\_live,

SUM(nvl(t1\_matched\_nonlive,0) ) AS t1\_matched\_nonlive,

SUM(nvl(ethernet\_discovered,0) ) AS ethernet\_discovered,

SUM(nvl(ethernet\_matched\_live,0) ) AS ethernet\_matched\_live,

SUM(nvl(ethernet\_matched\_nonlive,0) ) AS ethernet\_matched\_nonlive,

SUM(nvl(ethernet\_matched\_live\_csr,0) ) AS ethernet\_matched\_live\_csr,

SUM(nvl(ethernet\_matched\_nonlive\_csr,0) ) AS ethernet\_matched\_nonlive\_csr

FROM

aems\_region\_map mrm

LEFT OUTER JOIN t1\_discovered td ON mrm.ne\_ipbscdo\_number = td.ne\_ipbscdo\_number

LEFT OUTER JOIN t1\_matched\_live tm ON mrm.ne\_ipbscdo\_number = tm.ne\_ipbscdo\_number

LEFT OUTER JOIN t1\_matched\_nonlive tum ON mrm.ne\_ipbscdo\_number = tum.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_discovered ed ON mrm.ne\_ipbscdo\_number = ed.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_live em ON mrm.ne\_ipbscdo\_number = em.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_nonlive eum ON mrm.ne\_ipbscdo\_number = eum.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_live\_csr emc ON mrm.ne\_ipbscdo\_number = emc.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_nonlive\_csr eumc ON mrm.ne\_ipbscdo\_number = eumc.ne\_ipbscdo\_number

GROUP BY

ROLLUP(mrm.territory,

mrm.sub\_market)

ORDER BY

mrm.territory,

mrm.sub\_market';

execute IMMEDIATE sqlStmt;

COMMIT;

sqlStmtT1Comp := 'update moto\_evdo\_audit\_reg\_summ set t1\_comp = :x ,t1\_reported = :y where rowid = :a';

sqlStmtEbhComp := 'update moto\_evdo\_audit\_reg\_summ set ebh\_comp = :x,ebh\_reported = :y where rowid = :a';

sqlStmtOvlComp := 'update moto\_evdo\_audit\_reg\_summ set overall\_comp = :x where rowid = :a';

for cur in updateCompliance

loop

t1Reported := nvl(cur.t1\_discovered,0) - nvl(cur.t1\_matched\_nonlive,0);

ebhReported := nvl(cur.ebh\_discovered,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

overallComp := nvl(cur.t1\_discovered,0) + nvl(cur.ebh\_discovered,0) ;

overallComp := overallComp - nvl(cur.t1\_matched\_nonlive,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

if(t1Reported >0) then

t1Comp := (nvl(cur.t1\_matched\_live,0)/t1Reported) \*100;

t1Comp := round(t1Comp,2);

execute immediate sqlStmtT1Comp using t1Comp,t1Reported,cur.rd;

--dbms\_output.put\_line('Non null '|| t1Comp);

end if;

if(ebhReported>0) then

ebhComp :=(nvl(cur.ebh\_matched\_live\_csr,0)/ebhReported) \*100;

ebhComp := round(ebhComp,2);

execute immediate sqlStmtEbhComp using ebhComp,ebhReported,cur.rd;

--dbms\_output.put\_line('good EBH '||ebhComp);

end if;

if(overallComp>0) then

overallComp := ((nvl(cur.t1\_matched\_live,0)+nvl(cur.ebh\_matched\_live\_csr,0))/overallComp) \*100;

overallComp := round(overallComp,2);

execute immediate sqlStmtOvlComp using overallComp,cur.rd;

--dbms\_output.put\_line('good EBH '||overallComp);

end if;

end loop;

COMMIT;

sqlStmt := 'update moto\_evdo\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update moto\_evdo\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is null and wk.sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update moto\_evdo\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'update moto\_evdo\_audit\_reg\_summ wk set wk.SUB\_MARKET = ''Unknown'' where territory=''Unknown''' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

dbms\_output.put\_line('moto evdo regional summary created');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in populate\_evdo\_regional\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END moto\_evdo\_regional\_summary;

PROCEDURE moto\_evdo\_device\_summary as

cursor updateCompliance is

select summ.rowid rd,summ.\* from moto\_evdo\_audit\_dev\_summ summ;

sqlStmt VARCHAR2(32767);

sqlStmtT1Comp VARCHAR2(32767);

sqlStmtEbhComp VARCHAR2(32767);

sqlStmtOvlComp VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

BEGIN

sqlStmt := 'truncate table moto\_evdo\_audit\_dev\_summ';

execute immediate sqlStmt;

sqlStmt := 'INSERT INTO moto\_evdo\_audit\_dev\_summ (

territory,

sub\_market,

leaf\_domain,

vsm\_device\_name\_aems,

ipbscdo\_number,

t1\_discovered,

t1\_matched\_live,

t1\_matched\_nonlive,

ebh\_discovered,

ebh\_matched\_live,

ebh\_matched\_nonlive,

ebh\_matched\_live\_csr,

ebh\_matched\_nonlive\_csr

)

WITH aems\_region\_map AS (

SELECT DISTINCT

cdmp.territory,

cdmp.sub\_market,

cdmp.leaf\_domain\_name,

mvn.vsm\_device\_name\_aems aems,

mvn.ipbscdo\_number ne\_ipbscdo\_number

FROM

ng\_reports.clli\_domain\_map\_v cdmp

LEFT OUTER JOIN moto\_vsm\_name\_aems mvn ON substr(mvn.vsm\_device\_name\_aems,1,6) = substr(cdmp.clli,1,6)

WHERE

cdmp.territory <> ''NNO''

AND upper(mvn.status) = ''LIVE''

),t1\_discovered AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) t1\_discovered

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND mea.ne\_status IN (

''INS'',

''OOS''

)

AND mea.match\_code IN (

''BOTH'',

''NE Only''

)

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_discovered AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_discovered

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND mea.match\_code IN (

''BOTH'',

''NE Only''

)

)

GROUP BY

ne\_ipbscdo\_number

),t1\_matched\_live AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) t1\_matched\_live

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),t1\_matched\_nonlive AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) t1\_matched\_nonlive

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND upper(mea.ng\_path\_status) <> ''LIVE''

AND mea.match\_code = ''BOTH''

MINUS

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier,

mea.ne\_span\_number

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''T1''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_live AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_live

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_live\_csr AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_live\_csr

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

JOIN ng\_topology.trail\_element cpe ON mea.ng\_path\_inst\_id = cpe.trail\_id

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

AND mea.ng\_bandwidth LIKE ''%bps''

AND cpe.equipment\_type = ''CSR''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_nonlive AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_nonlive

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) <> ''LIVE''

AND mea.match\_code = ''BOTH''

MINUS

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

)

GROUP BY

ne\_ipbscdo\_number

),ethernet\_matched\_nonlive\_csr AS (

SELECT

ne\_ipbscdo\_number,

COUNT(1) ethernet\_matched\_nonlive\_csr

FROM

(

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

JOIN ng\_topology.trail\_element cpe ON mea.ng\_path\_inst\_id = cpe.trail\_id

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) <> ''LIVE''

AND mea.match\_code = ''BOTH''

AND mea.ng\_bandwidth LIKE ''%bps''

AND cpe.equipment\_type = ''CSR''

MINUS

SELECT DISTINCT

mea.aems,

mea.ne\_ipbscdo\_number,

mea.ne\_mccdo\_identifier

FROM

ng\_reports.moto\_evdo\_ne\_vs\_ng\_audit mea

JOIN ng\_topology.trail\_element cpe ON mea.ng\_path\_inst\_id = cpe.trail\_id

WHERE

mea.termination\_type = ''Ethernet''

AND upper(mea.ng\_path\_status) = ''LIVE''

AND mea.match\_code = ''BOTH''

AND mea.ng\_bandwidth LIKE ''%bps''

AND cpe.equipment\_type = ''CSR''

)

GROUP BY

ne\_ipbscdo\_number

) SELECT

mrm.territory,

mrm.sub\_market,

mrm.leaf\_domain\_name,

mrm.aems,

mrm.ne\_ipbscdo\_number,

nvl(t1\_discovered,0) AS t1\_discovered,

nvl(t1\_matched\_live,0) AS t1\_matched\_live,

nvl(t1\_matched\_nonlive,0) AS t1\_matched\_nonlive,

nvl(ethernet\_discovered,0) AS ethernet\_discovered,

nvl(ethernet\_matched\_live,0) AS ethernet\_matched\_live,

nvl(ethernet\_matched\_nonlive,0) AS ethernet\_matched\_nonlive,

nvl(ethernet\_matched\_live\_csr,0) AS ethernet\_matched\_live\_csr,

nvl(ethernet\_matched\_nonlive\_csr,0) AS ethernet\_matched\_nonlive\_csr

FROM

aems\_region\_map mrm

LEFT OUTER JOIN t1\_discovered td ON mrm.ne\_ipbscdo\_number = td.ne\_ipbscdo\_number

LEFT OUTER JOIN t1\_matched\_live tm ON mrm.ne\_ipbscdo\_number = tm.ne\_ipbscdo\_number

LEFT OUTER JOIN t1\_matched\_nonlive tum ON mrm.ne\_ipbscdo\_number = tum.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_discovered ed ON mrm.ne\_ipbscdo\_number = ed.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_live em ON mrm.ne\_ipbscdo\_number = em.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_nonlive eum ON mrm.ne\_ipbscdo\_number = eum.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_live\_csr emc ON mrm.ne\_ipbscdo\_number = emc.ne\_ipbscdo\_number

LEFT OUTER JOIN ethernet\_matched\_nonlive\_csr eumc ON mrm.ne\_ipbscdo\_number = eumc.ne\_ipbscdo\_number

ORDER BY

mrm.ne\_ipbscdo\_number';

execute immediate sqlStmt;

commit;

sqlStmtT1Comp := 'update moto\_evdo\_audit\_dev\_summ set t1\_comp = :x where rowid = :a';

sqlStmtEbhComp := 'update moto\_evdo\_audit\_dev\_summ set ebh\_comp = :x where rowid = :a';

sqlStmtOvlComp := 'update moto\_evdo\_audit\_dev\_summ set overall\_comp = :x where rowid = :a';

for cur in updateCompliance

loop

t1Comp := nvl(cur.t1\_discovered,0) - nvl(cur.t1\_matched\_nonlive,0);

ebhComp := nvl(cur.ebh\_discovered,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

overallComp :=nvl(cur.t1\_discovered,0) + nvl(cur.ebh\_discovered,0) ;

overallComp := overallComp - nvl(cur.t1\_matched\_nonlive,0) - nvl(cur.ebh\_matched\_nonlive\_csr,0);

if(t1Comp >0) then

t1Comp :=(nvl(cur.t1\_matched\_live,0)/t1Comp) \*100;

t1Comp := round(t1Comp,2);

execute immediate sqlStmtT1Comp using t1Comp,cur.rd;

--dbms\_output.put\_line('Non null '|| t1Comp);

end if;

if(ebhComp>0) then

ebhComp :=(nvl(cur.ebh\_matched\_live\_csr,0)/ebhComp) \*100;

ebhComp := round(ebhComp,2);

execute immediate sqlStmtEbhComp using ebhComp,cur.rd;

--dbms\_output.put\_line('good EBH '||ebhComp);

end if;

if(overallComp>0) then

overallComp :=((nvl(cur.t1\_matched\_live,0)+nvl(cur.ebh\_matched\_live\_csr,0))/overallComp) \*100;

overallComp := round(overallComp,2);

execute immediate sqlStmtOvlComp using overallComp,cur.rd;

--dbms\_output.put\_line('good EBH '||overallComp);

end if;

end loop;

commit;

dbms\_output.put\_line('moto evdo device summary created');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in populate\_evdo\_device\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END moto\_evdo\_device\_summary;

PROCEDURE copy\_moto\_cdma\_wrk AS

sqlstmt VARCHAR2(32000);

BEGIN

sqlstmt := 'delete from moto\_cdma\_ne\_vs\_ng\_audit';

EXECUTE IMMEDIATE sqlstmt;

sqlstmt := 'insert into moto\_cdma\_ne\_vs\_ng\_audit select \* from MOTO\_CDMA\_NE\_VS\_NG\_AUDIT\_WRK';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done populating moto\_cdma\_ne\_vs\_ng\_audit');

sqlstmt := 'delete from moto\_cdma\_audit\_reg\_summ';

EXECUTE IMMEDIATE sqlstmt;

sqlstmt := 'insert into moto\_cdma\_audit\_reg\_summ (T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,SUB\_MARKET) select T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,SUB\_MARKET from MOTO\_CDMA\_AUDIT\_REG\_SUMM'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is null and wk.sub\_market is not null'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt := 'update moto\_cdma\_audit\_reg\_summ wk set wk.SUB\_MARKET=''Unknown'' where territory=''Unknown''';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done populating moto\_cdma\_audit\_reg\_summ');

sqlstmt := 'delete from moto\_cdma\_audit\_dev\_summ';

EXECUTE IMMEDIATE sqlstmt;

sqlstmt := 'insert into moto\_cdma\_audit\_dev\_summ select \* from MOTO\_CDMA\_AUDIT\_DEV\_SUMM';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done populating moto\_cdma\_audit\_dev\_summ');

END copy\_moto\_cdma\_wrk;

PROCEDURE copy\_moto\_evdo\_wrk

AS

BEGIN

-- TODO: Implementation required for PROCEDURE MOTOROLA\_AUDIT.copy\_moto\_evdo\_wr

NULL;

END copy\_moto\_evdo\_wrk;

PROCEDURE truncate\_moto\_cdma\_wrk AS

sqlstmt VARCHAR2(32000);

BEGIN

sqlstmt := 'delete from MOTO\_CDMA\_NE\_VS\_NG\_AUDIT';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done deleting MOTO\_CDMA\_NE\_VS\_NG\_AUDIT');

sqlstmt := 'delete from MOTO\_CDMA\_AUDIT\_REG\_SUMM';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done deleting MOTO\_CDMA\_AUDIT\_REG\_SUMM');

sqlstmt := 'delete from MOTO\_CDMA\_AUDIT\_DEV\_SUMM';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done deleting MOTO\_CDMA\_AUDIT\_DEV\_SUMM');

END truncate\_moto\_cdma\_wrk;

PROCEDURE truncate\_moto\_evdo\_wrk

AS

BEGIN

-- TODO: Implementation required for PROCEDURE MOTOROLA\_AUDIT.truncate\_moto\_evdo\_wr

NULL;

END truncate\_moto\_evdo\_wrk;

PROCEDURE restore\_moto\_cdma\_wrk AS

sqlstmt VARCHAR2(32000);

BEGIN

sqlstmt := 'delete from moto\_cdma\_ne\_vs\_ng\_audit\_wrk';

EXECUTE IMMEDIATE sqlstmt;

sqlstmt := 'insert into moto\_cdma\_ne\_vs\_ng\_audit\_wrk select \* from moto\_cdma\_ne\_vs\_ng\_audit';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done populating moto\_cdma\_ne\_vs\_ng\_audit\_wrk');

sqlstmt := 'delete from moto\_cdma\_audit\_reg\_summ\_wrk';

EXECUTE IMMEDIATE sqlstmt;

sqlstmt := 'insert into moto\_cdma\_audit\_reg\_summ\_wrk (T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,MARKET\_TERRITORY,SUB\_MARKET, TERRITORY\_MARKET\_SUB) select T1\_DISCOVERED,T1\_MATCHED\_LIVE,T1\_MATCHED\_NONLIVE,EBH\_DISCOVERED,EBH\_MATCHED\_LIVE,EBH\_MATCHED\_NONLIVE,T1\_COMP,EBH\_COMP,OVERALL\_COMP,T1\_REPORTED,EBH\_REPORTED,EBH\_MATCHED\_NONLIVE\_CSR,EBH\_MATCHED\_LIVE\_CSR,UBS\_COUNT,NON\_UBS\_COUNT,TERRITORY,MARKET\_TERRITORY,SUB\_MARKET, TERRITORY\_MARKET\_SUB from moto\_cdma\_audit\_reg\_summ'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done populating moto\_cdma\_audit\_reg\_summ\_wrk');

sqlstmt := 'delete from moto\_cdma\_audit\_dev\_summ\_wrk';

EXECUTE IMMEDIATE sqlstmt;

sqlstmt := 'insert into moto\_cdma\_audit\_dev\_summ\_wrk select \* from moto\_cdma\_audit\_dev\_summ';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

dbms\_output.put\_line('Done populating moto\_cdma\_audit\_dev\_summ\_wrk');

-- EXCEPTION

-- WHEN OTHERS THEN

-- watchdog.logerror('MOTO\_CDMA\_AUDIT',4000,substr('Error in restore\_moto\_cdma\_wrk; '

-- || TO\_CHAR(sqlcode)

-- || ': '

-- || sqlerrm,1,255),'Y');

--

-- watchdog.updateprocessend('MOTO\_CDMA\_AUDIT','STATUS\_FAILURE','N');

--null;

END restore\_moto\_cdma\_wrk;

PROCEDURE restore\_moto\_evdo\_wrk

AS

BEGIN

-- TODO: Implementation required for PROCEDURE MOTOROLA\_AUDIT.restore\_moto\_evdo\_wrk

NULL;

END restore\_moto\_evdo\_wrk;

END motorola\_audit;

/

--------------------------------------------------------

-- DDL for Package Body MSPP\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."MSPP\_AUDIT" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: MSPP\_AUDIT

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 4/20/2018 Sai Kiran Kandari 1. Created this package.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE copy\_mspp\_data\_from\_wk\_to\_prod

IS

-- v\_ready ng\_reports.all\_processes.is\_ready%TYPE;

--

-- BEGIN

-- SELECT is\_ready

-- INTO v\_ready

-- FROM ng\_reports.all\_processes

-- WHERE process\_name = 'INC\_MSPP\_AUDIT';

--

---- IF v\_ready = 'Y'

-- THEN

sql\_stmt VARCHAR2 (32000);

BEGIN

sql\_stmt := 'truncate table ' || INC\_MSPP\_XCONN\_AUDIT\_DETAILS;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INC\_MSPP\_XCONN\_AUDIT\_DETAILS

|| ' select \* from '

|| INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INC\_MSPP\_XCONN\_AUDIT\_DETAILS);

sql\_stmt := 'truncate table ' || INC\_MSPP\_XCONN\_AUDIT\_SUMMARY;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INC\_MSPP\_XCONN\_AUDIT\_SUMMARY

|| ' select \* from '

|| INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INC\_MSPP\_XCONN\_AUDIT\_SUMMARY);

sql\_stmt := 'truncate table ' || MSPP\_CIRC\_PATH\_ELEMENTS;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| MSPP\_CIRC\_PATH\_ELEMENTS

|| ' select \* from '

|| MSPP\_CIRC\_PATH\_ELEMENTS\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || MSPP\_CIRC\_PATH\_ELEMENTS);

sql\_stmt := 'truncate table ' || MSPP\_SUMMARY\_REGION\_MAP;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| MSPP\_SUMMARY\_REGION\_MAP

|| ' select \* from '

|| MSPP\_SUMMARY\_REGION\_MAP\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || MSPP\_SUMMARY\_REGION\_MAP);

sql\_stmt := 'truncate table ' || MSPP\_VSM\_MATCH;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| MSPP\_VSM\_MATCH

|| ' select \* from '

|| MSPP\_VSM\_MATCH\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || MSPP\_VSM\_MATCH);

sql\_stmt := 'truncate table ' || INC\_MSPP\_ADM\_COMP\_SUMMARY;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INC\_MSPP\_ADM\_COMP\_SUMMARY

|| ' select \* from '

|| INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INC\_MSPP\_ADM\_COMP\_SUMMARY);

sql\_stmt := 'truncate table ' || INC\_MSPP\_COMPLIANCE\_SUMMARY;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INC\_MSPP\_COMPLIANCE\_SUMMARY

|| ' select \* from '

|| INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INC\_MSPP\_COMPLIANCE\_SUMMARY);

-- END IF;

EXCEPTION

WHEN OTHERS

THEN

DBMS\_OUTPUT.put\_line (sql\_stmt);

DBMS\_OUTPUT.put\_line (SUBSTR ( 'Error in copy\_mspp\_data\_from\_wk\_to\_prod(): '

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

-- RAISE;

END copy\_mspp\_data\_from\_wk\_to\_prod;

PROCEDURE copy\_mspp\_data\_from\_prod\_to\_wk

IS

sql\_stmt VARCHAR2 (32000);

BEGIN

sql\_stmt := 'truncate table ' || INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK

|| ' select \* from '

|| INC\_MSPP\_XCONN\_AUDIT\_DETAILS;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK);

sql\_stmt := 'truncate table ' || INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK

|| ' select \* from '

|| INC\_MSPP\_XCONN\_AUDIT\_SUMMARY;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK);

sql\_stmt := 'truncate table ' || MSPP\_CIRC\_PATH\_ELEMENTS\_WK;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| MSPP\_CIRC\_PATH\_ELEMENTS\_WK

|| ' select \* from '

|| MSPP\_CIRC\_PATH\_ELEMENTS;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || MSPP\_CIRC\_PATH\_ELEMENTS\_WK);

sql\_stmt := 'truncate table ' || MSPP\_SUMMARY\_REGION\_MAP\_WK;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| MSPP\_SUMMARY\_REGION\_MAP\_WK

|| ' select \* from '

|| MSPP\_SUMMARY\_REGION\_MAP;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || MSPP\_SUMMARY\_REGION\_MAP\_WK);

sql\_stmt := 'truncate table ' || MSPP\_VSM\_MATCH\_WK;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| MSPP\_VSM\_MATCH\_WK

|| ' select \* from '

|| MSPP\_VSM\_MATCH;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || MSPP\_VSM\_MATCH\_WK);

sql\_stmt := 'truncate table ' || INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK

|| ' select \* from '

|| INC\_MSPP\_ADM\_COMP\_SUMMARY;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK);

sql\_stmt := 'truncate table ' || INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt :=

'insert into '

|| INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK

|| ' select \* from '

|| INC\_MSPP\_COMPLIANCE\_SUMMARY;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Done populating ' || INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK);

EXCEPTION

WHEN OTHERS

THEN

DBMS\_OUTPUT.put\_line (sql\_stmt);

DBMS\_OUTPUT.put\_line (SUBSTR ( 'Error in copy\_mspp\_data\_from\_prod\_to\_wk(): '

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

--RAISE;

END copy\_mspp\_data\_from\_prod\_to\_wk;

PROCEDURE truncate\_mspp\_wk

IS

sql\_stmt VARCHAR2 (32000);

BEGIN

sql\_stmt := 'delete from ' || INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Deleted from ' || INCMSPP\_XCONN\_AUDIT\_DETAILS\_WK);

sql\_stmt := 'delete from ' || INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Deleted from ' || INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK);

sql\_stmt := 'delete from ' || MSPP\_CIRC\_PATH\_ELEMENTS\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Deleted from ' || MSPP\_CIRC\_PATH\_ELEMENTS\_WK);

sql\_stmt := 'delete from ' || MSPP\_SUMMARY\_REGION\_MAP\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Deleted from ' || MSPP\_SUMMARY\_REGION\_MAP\_WK);

sql\_stmt := 'delete from ' || MSPP\_VSM\_MATCH\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Deleted from ' || MSPP\_VSM\_MATCH\_WK);

sql\_stmt := 'delete from ' || INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Deleted from ' || INC\_MSPP\_ADM\_COMP\_SUMMARY\_WK);

sql\_stmt := 'delete from ' || INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK;

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

DBMS\_OUTPUT.put\_line ('Deleted from ' || INC\_MSPP\_COMPLIANCE\_SUMMARY\_WK);

EXCEPTION

WHEN OTHERS

THEN

DBMS\_OUTPUT.put\_line (sql\_stmt);

DBMS\_OUTPUT.put\_line (SUBSTR ( 'Error in truncate\_mspp\_wk(): '

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

--RAISE;

END truncate\_mspp\_wk;

END MSPP\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package Body NCM\_CSR\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."NCM\_CSR\_AUDIT" AS

procedure NCM\_AUDIT is

sql\_stmt VARCHAR2(32000);

Begin

Begin

sql\_stmt := 'truncate table ncm\_hpov\_eh\_nt\_ci\_csr\_audit';

EXECUTE IMMEDIATE sql\_stmt;

/\*SAM/NCM GI ServOne Cilli Summary Report :\*/

INSERT INTO ncm\_hpov\_eh\_nt\_ci\_csr\_audit colums (

csr\_vendor,

csr\_device\_name,

ncm\_csr\_hostname,

ncm\_csr\_partition,

ncm\_csr\_device\_ip,

ncm\_csr\_vlan\_hostname,

ncm\_csr\_vlan\_partition,

ncm\_csr\_vlan\_device\_ip,

hp\_csr\_hostname,

hp\_csr\_ip,

eh\_csr\_hostname,

eh\_csr\_ipaddress,

nt\_device\_name,

eqp\_reference\_id,

eqp\_name,

LIVE\_in\_ng,

match\_code,

cell\_site\_name

)

WITH ncm\_csrs AS (

SELECT DISTINCT

ncm.hostname,

ncm.partition,

ncm.device\_ip ncm\_csr\_device\_ip

FROM

ng\_reports.ncm\_csr\_wk ncm

),ncm\_csvlan AS (

SELECT DISTINCT

ncm.hostname,

ncm.partition,

ncm.device\_ip ncm\_csr\_vlan\_device\_ip

FROM

ng\_reports.ncm\_csr\_vlan\_wk ncm

),hpov AS (

-- new table where all the hpov data should go

-- from here put it in csr\_devices\_wk

SELECT

hp.device\_name,

hp.ip -- hp.ne\_status, hp.ne\_source

FROM

ng\_reports.hpov\_csr\_wk hp

--where csr\_device\_name like ''%-CI-%''

),ehealth\_csr AS (

SELECT

ecl.device\_name,

ecl.device\_ip

FROM

ng\_reports.sevone\_csr ecl

WHERE

ecl.device\_name LIKE '%-CI-%'

),xng AS (

SELECT

csr.\*,

ei.inv\_status ei\_status

FROM

ng\_reports.ntls\_csr\_parsed\_wk csr --Doubt

JOIN ng\_reports.equipment ei ON ei.eqp\_reference\_id = csr.eqp\_reference\_id

WHERE

csr.csr\_device\_name LIKE '%-CI-%'

),brix\_csr AS (

SELECT

brx.cell\_site\_name

FROM

ng\_reports.brix\_csr\_discovery\_wk brx

WHERE

brx.cell\_site\_name LIKE '%-CI-%'

),ncm\_csvlan\_hpov\_full\_join AS (

SELECT

CASE

WHEN ncm.hostname IS NOT NULL THEN hostname

WHEN hp.device\_name IS NOT NULL THEN hp.device\_name

END

csr\_device\_name,

ncm.hostname ncm\_csr\_vlan\_hostname,

ncm.partition ncm\_csr\_vlan\_partition,

ncm.ncm\_csr\_vlan\_device\_ip,

hp.device\_name hp\_csr\_hostname,

hp.ip hp\_csr\_ip,

CASE

WHEN ncm.hostname IS NOT NULL

AND hp.device\_name IS NOT NULL THEN 'BOTH'

WHEN ncm.hostname IS NOT NULL

OR hp.device\_name IS NOT NULL THEN 'NE Only'

ELSE 'xxNE Only'

END

match\_code

FROM

hpov hp

FULL OUTER JOIN ncm\_csvlan ncm ON upper(hp.device\_name) = upper(ncm.hostname)

),ncm\_csrs\_csvlan\_hpov\_full\_join AS (

SELECT

CASE

WHEN nc.hostname IS NOT NULL THEN nc.hostname

WHEN ncvhfj.csr\_device\_name IS NOT NULL THEN ncvhfj.csr\_device\_name

END

csr\_device\_name,

ncvhfj.ncm\_csr\_vlan\_hostname,

ncvhfj.ncm\_csr\_vlan\_partition,

ncvhfj.ncm\_csr\_vlan\_device\_ip,

hp\_csr\_hostname,

hp\_csr\_ip,

nc.hostname ncm\_csr\_hostname,

nc.partition ncm\_csr\_partition,

nc.ncm\_csr\_device\_ip, --, ne\_status, ne\_source

CASE

WHEN nc.hostname IS NOT NULL

AND ncvhfj.csr\_device\_name IS NOT NULL

AND match\_code = 'BOTH' THEN 'BOTH'

WHEN nc.hostname IS NOT NULL THEN 'NE Only'

WHEN ncvhfj.csr\_device\_name IS NOT NULL THEN 'NE Only'

ELSE 'xxNE Only'

END

match\_code

FROM

ncm\_csvlan\_hpov\_full\_join ncvhfj

FULL OUTER JOIN ncm\_csrs nc ON upper(nc.hostname) = upper(ncvhfj.csr\_device\_name)

),ncm\_csrs\_hpov\_eh\_full\_join AS (

SELECT

CASE

WHEN ecsr.device\_name IS NOT NULL THEN ecsr.device\_name

WHEN nhfj.csr\_device\_name IS NOT NULL THEN nhfj.csr\_device\_name

END

csr\_device\_name,

nhfj.ncm\_csr\_hostname,

nhfj.ncm\_csr\_partition,

nhfj.ncm\_csr\_device\_ip,

nhfj.ncm\_csr\_vlan\_hostname,

nhfj.ncm\_csr\_vlan\_partition,

nhfj.ncm\_csr\_vlan\_device\_ip,

nhfj.hp\_csr\_hostname,

nhfj.hp\_csr\_ip,

ecsr.device\_name eh\_csr\_hostname,

ecsr.device\_ip eh\_csr\_ipaddress,

CASE

WHEN ecsr.device\_name IS NOT NULL

AND nhfj.csr\_device\_name IS NOT NULL

AND match\_code = 'BOTH' THEN 'BOTH'

WHEN ecsr.device\_name IS NOT NULL THEN 'NE Only'

WHEN nhfj.csr\_device\_name IS NOT NULL THEN 'NE Only'

ELSE 'xxNE Only'

END

match\_code

FROM

ncm\_csrs\_csvlan\_hpov\_full\_join nhfj

FULL OUTER JOIN ng\_reports.sevone\_csr ecsr ON upper(nhfj.csr\_device\_name) = upper(ecsr.device\_name)

),ncm\_csrs\_hpov\_eh\_br\_full\_join AS (

SELECT

CASE

WHEN br.cell\_site\_name IS NOT NULL THEN br.cell\_site\_name

WHEN nhfj.csr\_device\_name IS NOT NULL THEN nhfj.csr\_device\_name

END

csr\_device\_name,

nhfj.ncm\_csr\_hostname,

nhfj.ncm\_csr\_partition,

nhfj.ncm\_csr\_device\_ip,

nhfj.ncm\_csr\_vlan\_hostname,

nhfj.ncm\_csr\_vlan\_partition,

nhfj.ncm\_csr\_vlan\_device\_ip,

nhfj.hp\_csr\_hostname,

nhfj.hp\_csr\_ip,

nhfj.eh\_csr\_hostname eh\_csr\_hostname,

nhfj.eh\_csr\_ipaddress eh\_csr\_ipaddress,

CASE

WHEN nhfj.csr\_device\_name IS NOT NULL

AND br.cell\_site\_name IS NOT NULL

AND match\_code = 'BOTH' THEN 'BOTH'

WHEN nhfj.csr\_device\_name IS NOT NULL THEN 'NE Only'

WHEN br.cell\_site\_name IS NOT NULL THEN 'NE Only'

ELSE 'xxNE Only'

END

match\_code,

br.cell\_site\_name

FROM

ncm\_csrs\_hpov\_eh\_full\_join nhfj

FULL OUTER JOIN ng\_reports.brix\_csr\_discovery\_wk br ON upper(nhfj.csr\_device\_name) = upper(br.cell\_site\_name)

) SELECT DISTINCT

'CISCO' csr\_vendor,

CASE

WHEN nhefj.csr\_device\_name IS NOT NULL THEN nhefj.csr\_device\_name

WHEN xng.csr\_device\_name IS NOT NULL THEN xng.csr\_device\_name

END

csr\_device\_name,

nhefj.ncm\_csr\_hostname,

nhefj.ncm\_csr\_partition,

nhefj.ncm\_csr\_device\_ip,

nhefj.ncm\_csr\_vlan\_hostname,

nhefj.ncm\_csr\_vlan\_partition,

nhefj.ncm\_csr\_vlan\_device\_ip,

nhefj.hp\_csr\_hostname,

nhefj.hp\_csr\_ip,

nhefj.eh\_csr\_hostname,

nhefj.eh\_csr\_ipaddress,

xng.csr\_device\_name nt\_device\_name,

xng.eqp\_reference\_id,

xng.eqp\_name,

CASE

WHEN xng.ei\_status = 'LIVE' THEN 'Y'

WHEN xng.ei\_status IS NOT NULL THEN 'N'

END

LIVE\_in\_ng,

CASE

WHEN nhefj.csr\_device\_name IS NOT NULL

AND xng.csr\_device\_name IS NOT NULL

AND match\_code = 'BOTH' THEN 'BOTH'

WHEN nhefj.csr\_device\_name IS NOT NULL THEN 'NE Only'

WHEN xng.csr\_device\_name IS NOT NULL THEN 'Nautilus Only'

ELSE 'xxNE Only'

END

match\_code,

nhefj.cell\_site\_name

FROM

ncm\_csrs\_hpov\_eh\_br\_full\_join nhefj

FULL OUTER JOIN xng ON upper(xng.csr\_device\_name) = upper(nhefj.csr\_device\_name);

commit;

END;

insert\_ci\_csr\_equip\_issues();

EXECUTE IMMEDIATE sql\_stmt;

commit;

END NCM\_AUDIT;

PROCEDURE insert\_ci\_csr\_equip\_issues

IS

methodname VARCHAR2 (30) := 'insert\_CI\_csr\_equip\_issues';

MESSAGE VARCHAR2 (500);

sql\_stmt VARCHAR2 (32767);

issue\_table VARCHAR2(50) := 'csr\_device\_audit\_issues';

insertSqlStart VARCHAR2(100) := 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor ) ';

BEGIN

begin

sql\_stmt := 'delete from ' || issue\_table || ' where csr\_vendor=''CISCO''';

EXECUTE IMMEDIATE sql\_stmt;

commit;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t delete records from table: ' || issue\_table;

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

-- CSR name in NCM CSR Extract not Std Complaint --

/\* insert\_csr\_issues (processname, methodname, insertSqlStart ||

'select distinct hostname, 16, ''CISCO'' from NCM\_CSR\_WK aud where parse\_status like ''%16%''',

'Can''t insert into ' || issue\_table || ' issue 16');\*/

-- CSR name in eHealth Extract not Std Complaint --

/\* insert\_csr\_issues (processname, methodname, insertSqlStart ||

'select distinct hostname, 25, ''CISCO'' from ehealth\_csr\_list where error\_code = 25 and vendor = ''CISCO''',

'Can''t insert into ' || issue\_table || ' issue ehealth 25');\*/

-- CSR name in NCM CSR VLAN Extract not Std Complaint --

/\* insert\_csr\_issues (processname, methodname, insertSqlStart ||

'select distinct hostname, 17, ''CISCO'' from NCM\_CSR\_VLAN\_WK aud where parse\_status like ''%17%''',

'Can''t insert into ' || issue\_table || ' issue 17');\*/

/\* -- GI name not Std Complaint --

insert\_csr\_issues (processname, methodname, insertSqlStart ||

'select distinct csr\_device\_name, 7, ''CISCO'' from NTLS\_CSR\_PARSED\_WK aud where csr\_device\_name like ''%P-CI%'' and parse\_status like ''%Xng CSR Name does not match Standard%''',

'Can''t insert into ' || issue\_table || ' issue 7');

-- Invalid CSR Name --

insert\_csr\_issues (processname, methodname, insertSqlStart ||

'select distinct csr\_device\_name, 8, ''CISCO'' from NTLS\_CSR\_PARSED\_WK aud where csr\_device\_name like ''%P-CI%'' and parse\_status like ''%Invalid CSR Name%''',

'Can''t insert into ' || issue\_table || ' issue 8');

\*/

-- 12-14 issues in the previous are critical ONLY IF match\_code <> 'Nautilus Only'

-- the scheme needs to change somehow on the csr\_issues-------???????

-- device not found in ncm csr extract

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 12, ''CISCO'' from ncm\_hpov\_eh\_nt\_ci\_csr\_audit aud where ncm\_csr\_hostname is null';

EXECUTE IMMEDIATE sql\_stmt;

-- device not found in eHealth extract

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 24, ''CISCO'' from ncm\_hpov\_eh\_nt\_ci\_csr\_audit aud where eh\_csr\_hostname is null';

EXECUTE IMMEDIATE sql\_stmt;

-- device not found in ncm vlan extract

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 13, ''CISCO'' from ncm\_hpov\_eh\_nt\_ci\_csr\_audit aud where ncm\_csr\_vlan\_hostname is null';

EXECUTE IMMEDIATE sql\_stmt;

-- device not found in hpov

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 14, ''CISCO'' from ncm\_hpov\_eh\_nt\_ci\_csr\_audit aud where hp\_csr\_hostname is null';

EXECUTE IMMEDIATE sql\_stmt;

-- device not found in Brix

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 33, ''CISCO'' from ncm\_hpov\_eh\_nt\_ci\_csr\_audit aud where brix\_csr\_site is null';

EXECUTE IMMEDIATE sql\_stmt;

-- device not found in Xng

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 15, ''CISCO'' from ncm\_hpov\_eh\_nt\_ci\_csr\_audit aud where eqp\_reference\_id is null';

EXECUTE IMMEDIATE sql\_stmt;

-- DUP CSR match found in Xng

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 1, ''CISCO'' from ncm\_hpov\_eh\_nt\_ci\_csr\_audit i where match\_code <> ''Nautilus Only'' group by i.csr\_device\_name having count(1) > 1';

-- DUPs found in NE. is this possible ????

EXECUTE IMMEDIATE sql\_stmt;

-- mark 6-char match-not all- use from before

-- mark Invalid CLLI

-- We might have discovered some CLLI that is not in the list of MTSO CLLI

-- mark those as bad

-- We might have discovered some CLLI where 1st 6-char match MTSO CLLI but all 8 dont

-- mark those as bad

-- some times there are 2 legitimate cllis for the same site

-- in those cases the o/p comes as followd

-- clli\_1\_csr clli\_2\_vsm CLLI Mismatch

-- clli\_2\_csr clli\_1\_vsm CLLI Mismatch

-- so reverse the order and subtract it. That will give only the ones that are

-- truly bad

/\* want to be more specific to NCM/NCM VLAN and HPOV \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--NCM issues 18 and 20

begin

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT ei.ncm\_csr\_hostname,

SUBSTR (ei.ncm\_csr\_hostname, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.ncm\_csr\_hostname, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.ncm\_csr\_hostname, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.ncm\_csr\_hostname

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei

MINUS

SELECT ei.ncm\_csr\_hostname

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.ncm\_csr\_hostname, 1, 8)

MINUS

SELECT ei.ncm\_csr\_hostname

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.ncm\_csr\_hostname, 1, 6)

),

ISSUES AS

(

SELECT cm.ncm\_csr\_hostname, 20 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.ncm\_csr\_hostname, 18 issue\_id

FROM bad\_clli bc

)

SELECT distinct WK.CSR\_DEVICE\_NAME , i.issue\_id, ''CISCO''

FROM ISSUES i, ncm\_hpov\_eh\_nt\_ci\_csr\_audit wk

where i.ncm\_csr\_hostname = Wk.NCM\_CSR\_HOSTNAME

';

EXECUTE IMMEDIATE sql\_stmt;

end;

--eHealth issues 26 and 27

begin

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT ei.eh\_csr\_hostname,

SUBSTR (ei.eh\_csr\_hostname, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.ncm\_csr\_hostname, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.eh\_csr\_hostname, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.eh\_csr\_hostname

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei

MINUS

SELECT ei.eh\_csr\_hostname

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.eh\_csr\_hostname, 1, 8)

MINUS

SELECT ei.eh\_csr\_hostname

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.eh\_csr\_hostname, 1, 6)

),

ISSUES AS

(

SELECT cm.eh\_csr\_hostname, 27 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.eh\_csr\_hostname, 26 issue\_id

FROM bad\_clli bc

)

SELECT distinct WK.CSR\_DEVICE\_NAME , i.issue\_id, ''CISCO''

FROM ISSUES i, ncm\_hpov\_eh\_nt\_ci\_csr\_audit wk

where i.eh\_csr\_hostname = Wk.NCM\_CSR\_HOSTNAME

';

EXECUTE IMMEDIATE sql\_stmt;

end;

--NCM VLAN issues 22 and 23

begin

sql\_stmt:= 'insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT EI.NCM\_CSR\_VLAN\_HOSTNAME ,

SUBSTR (ei.NCM\_CSR\_VLAN\_HOSTNAME, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.NCM\_CSR\_VLAN\_HOSTNAME, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.NCM\_CSR\_VLAN\_HOSTNAME, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.NCM\_CSR\_VLAN\_HOSTNAME

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei

MINUS

SELECT ei.NCM\_CSR\_VLAN\_HOSTNAME

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.NCM\_CSR\_VLAN\_HOSTNAME, 1, 8)

MINUS

SELECT ei.NCM\_CSR\_VLAN\_HOSTNAME

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.NCM\_CSR\_VLAN\_HOSTNAME, 1, 6)

),

ISSUES AS

(

SELECT cm.NCM\_CSR\_VLAN\_HOSTNAME, 22 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.NCM\_CSR\_VLAN\_HOSTNAME, 23 issue\_id

FROM bad\_clli bc

)

SELECT distinct WK.CSR\_DEVICE\_NAME , i.issue\_id, ''CISCO''

FROM ISSUES i, ncm\_hpov\_eh\_nt\_ci\_csr\_audit wk

where i.NCM\_CSR\_VLAN\_HOSTNAME = Wk.NCM\_CSR\_VLAN\_HOSTNAME

';

EXECUTE IMMEDIATE sql\_stmt;

end;

--HPOV issues 19 and 21

begin

sql\_stmt:='insert into ' || issue\_table || ' columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT EI.HP\_CSR\_HOSTNAME ,

SUBSTR (ei.HP\_CSR\_HOSTNAME, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.HP\_CSR\_HOSTNAME, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.HP\_CSR\_HOSTNAME, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.HP\_CSR\_HOSTNAME

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei

MINUS

SELECT ei.HP\_CSR\_HOSTNAME

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.HP\_CSR\_HOSTNAME, 1, 8)

MINUS

SELECT ei.HP\_CSR\_HOSTNAME

FROM ncm\_hpov\_eh\_nt\_ci\_csr\_audit ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.HP\_CSR\_HOSTNAME, 1, 6)

),

ISSUES AS

(

SELECT cm.HP\_CSR\_HOSTNAME, 21 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.HP\_CSR\_HOSTNAME, 19 issue\_id

FROM bad\_clli bc

)

SELECT distinct WK.CSR\_DEVICE\_NAME , i.issue\_id, ''CISCO''

FROM ISSUES i, ncm\_hpov\_eh\_nt\_ci\_csr\_audit wk

where i.HP\_CSR\_HOSTNAME = Wk.HP\_CSR\_HOSTNAME

';

EXECUTE IMMEDIATE sql\_stmt;

end;

BEGIN

-- if some ne\_hostname matched > 1 granite ne\_inst\_id then mark it dup

update ncm\_hpov\_eh\_nt\_ci\_csr\_audit aud set match\_code = 'NE Only'

where exists (select 1

from ncm\_hpov\_eh\_nt\_ci\_csr\_audit i

where aud.csr\_device\_name = i.csr\_device\_name

group by i.csr\_device\_name

having count(1) > 1

)

and match\_code <> 'Nautilus Only'

;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t mark match\_code Duplicate in Granite';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

COMMIT;

END insert\_ci\_csr\_equip\_issues;

Procedure NCM\_CSR\_CLLI\_SUM\_LOAD IS

sqlstmt VARCHAR(4000);

updStmt VARCHAR(4000);

insStmt VARCHAR(4000);

message VARCHAR(4000);

methodName VARCHAR(4000);

l\_cnt NUMBER;

cursor CSRs\_NCM is

WITH

DOMAINS AS(

select distinct substr(clli, 1, 6) CLLI, TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, SUB\_MARKET\_LEAF as market, LEAF\_DOMAIN\_ID

from NG\_Reports.CLLI\_DOMAIN\_MAP\_V

where sub\_market <> 'NNO' and sub\_market <> 'OSS'),

NCM as(

select distinct HOSTNAME, substr(HOSTNAME, 1, 6) CLLI

from ncm\_csr\_wk ncm)

select nvl(TERRITORY, 'unknown') TERRITORY, nvl(MARKET\_TERRITORY, 'unknown') MARKET\_TERRITORY, nvl(SUB\_MARKET, 'unknown') SUB\_MARKET, nvl(market, 'unknown') market, nvl(di.LEAF\_DOMAIN\_ID, 0) LEAF\_DOMAIN\_ID , ncm.CLLI, count (1) ncm\_csrs

from DOMAINS di

right outer join NCM on ncm.CLLI = di.CLLI

group by TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, market, di.LEAF\_DOMAIN\_ID, ncm.CLLI;

cursor CSRs\_EHEALTH is

WITH

DOMAINS AS (

select distinct substr(clli, 1, 6) CLLI, TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, SUB\_MARKET\_LEAF as market, LEAF\_DOMAIN\_ID

from NG\_Reports.CLLI\_DOMAIN\_MAP\_V

where SUB\_MARKET <> 'NNO' and SUB\_MARKET <> 'OSS'), EHEALTH as (

select distinct DEVICE\_NAME, substr(DEVICE\_NAME, 1, 6) CLLI

from NG\_REPORTS.SEVONE\_CSR where DEVICE\_NAME like '%-CI-%')

-- from ehealth\_csr\_list where vendor = 'CI')

select nvl(TERRITORY, 'unknown') TERRITORY, nvl(MARKET\_TERRITORY, 'unknown') MARKET\_TERRITORY, nvl(SUB\_MARKET, 'unknown') SUB\_MARKET, nvl(market, 'unknown') market , nvl(di.LEAF\_DOMAIN\_ID, 0) LEAF\_DOMAIN\_ID , eh.CLLI, count (1) eh\_csrs

from DOMAINS di

right outer join EHEALTH eh on eh.CLLI = di.CLLI

group by TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, market, di.LEAF\_DOMAIN\_ID , eh.CLLI

;

cursor CSRs\_BRIX is

WITH

DOMAINS AS (

select distinct substr(clli, 1, 6) CLLI, TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, SUB\_MARKET\_LEAF as market, LEAF\_DOMAIN\_ID

from NG\_Reports.CLLI\_DOMAIN\_MAP\_V

where SUB\_MARKET <> 'NNO'

and SUB\_MARKET <> 'OSS' ), BRIX as(

select distinct CELL\_SITE\_NAME, substr(CELL\_SITE\_NAME, 1, 6) CLLI

from NG\_Reports.BRIX\_CSR\_DISCOVERY\_WK where CELL\_SITE\_NAME like '%-CI-%')

select nvl(TERRITORY, 'unknown') TERRITORY, nvl(MARKET\_TERRITORY, 'unknown') MARKET\_TERRITORY, nvl(SUB\_MARKET, 'unknown') SUB\_MARKET, nvl(market, 'unknown') market , nvl(di.LEAF\_DOMAIN\_ID, 0) LEAF\_DOMAIN\_ID , br.CLLI , count (1) brix\_csrs

from DOMAINS di

right outer join BRIX br on br.CLLI = di.CLLI

group by TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, market, di.LEAF\_DOMAIN\_ID , br.CLLI;

cursor CSRs\_VLAN\_NCM is

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market, leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> 'NNO'

and sub\_market <> 'OSS'

)

,

NCM as

(

select distinct hostname, substr(hostname, 1, 6) clli\_6

from ncm\_csr\_vlan\_wk ncm

)

select nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, nvl(market, 'unknown') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, ncm.clli\_6

, count (1) ncm\_csrs

from DOMAINS di

right outer join NCM

on ncm.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, ncm.clli\_6

;

CURSOR BRIX\_nt\_MATCH\_CSRs is

WITH

BRIX\_nt\_MATCH as

(

select csr\_device\_name --, match\_status, live\_in\_ng

, case when live\_in\_ng = 'Y' then 1 else 0 end brix\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end brix\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

where aud.BRIX\_CSR\_SITE is not null

and aud.nt\_DEVICE\_NAME is not null

minus

-- DUPS are NOT good

select aud.csr\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end ncm\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end ncm\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

join csr\_device\_audit\_issues cdai

on upper(cdai.csr\_device\_name) = upper(aud.csr\_device\_name)

where csr\_issue\_id = 1

and match\_code <> 'Nautilus Only'

)

select case when br.csr\_device\_name is not null then substr(br.csr\_device\_name, 1, 6)

end clli\_6

, count(br.csr\_device\_name) brix\_nt\_match

, sum(br.brix\_nt\_l\_match) brix\_nt\_l\_match

, sum(br.brix\_nt\_nl\_match) brix\_nt\_nl\_match

from BRIX\_nt\_MATCH br

group by case when br.csr\_device\_name is not null then substr(br.csr\_device\_name, 1, 6)

end

;

cursor NCM\_CSRs\_W\_ISSUES is

with

domains as

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market,

leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> 'NNO'

and sub\_market <> 'OSS'

),

ncm\_csr\_w\_issues as (

select distinct cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id in (20, 22, 23, 12, 13, 16, 17, 18)

and CAI.CSR\_VENDOR='CISCO'

minus

(

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =12

and CAI.CSR\_VENDOR='CISCO'

intersect

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =13

and CAI.CSR\_VENDOR='CISCO'

)

),

ncm as

(

select distinct csr\_device\_name, substr(csr\_device\_name, 1, 6) clli\_6 from ncm\_csr\_w\_issues ncm

)

select nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, nvl(market, 'unknown') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, ncm.clli\_6

, count (1) ncm\_csrs\_w\_issues

from DOMAINS di

right outer join ncm

on ncm.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, ncm.clli\_6 order by territory, market\_territory, sub\_market

;

cursor NCM\_HPOV\_ISSUES is

with

domains as

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market,

leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> 'NNO'

and sub\_market <> 'OSS'

),

ncm\_hpov\_issues as (

select cai.csr\_device\_name

from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id in (13, 14, 17, 19, 21, 22, 23) and CAI.CSR\_VENDOR='CISCO'

minus

(

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =14 and CAI.CSR\_VENDOR='CISCO'

intersect

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =13 and CAI.CSR\_VENDOR='CISCO'

)

),

ncm\_hpov as

(

select distinct csr\_device\_name, substr(csr\_device\_name, 1, 6) clli\_6

from ncm\_hpov\_issues ncm

)

select nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, nvl(market, 'unknown') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, ncm.clli\_6

, count (1) ncm\_hpov

from DOMAINS di

right outer join ncm\_hpov ncm

on ncm.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, ncm.clli\_6 order by territory, market\_territory, sub\_market

;

cursor NCM\_nt\_ISSUES is

with

domains as

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market,

leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> 'NNO'

and sub\_market <> 'OSS'

),

ncm\_nt\_issues as (

select cai.csr\_device\_name

from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id in (17,1,7,8,13,15,22,23) and CAI.CSR\_VENDOR='CISCO'

minus

(

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =13 and CAI.CSR\_VENDOR='CISCO'

intersect

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =15 and CAI.CSR\_VENDOR='CISCO'

)

),ncm\_gi as

(

select distinct csr\_device\_name, substr(csr\_device\_name, 1, 6) clli\_6

from ncm\_nt\_issues ncm

)

select nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, nvl(market, 'unknown') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, ncm.clli\_6

, count (1) ncm\_gi

from DOMAINS di

right outer join ncm\_gi ncm

on ncm.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, ncm.clli\_6 order by territory, market\_territory, sub\_market

;

cursor EH\_nt\_ISSUES is

with

domains as

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market,

leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> 'NNO'

and sub\_market <> 'OSS'

),

eh\_nt\_issues as (

select cai.csr\_device\_name

from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id in (1,27,8,26,24,25,15) and CAI.CSR\_VENDOR='CISCO'

minus

(

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =24 and CAI.CSR\_VENDOR='CISCO'

intersect

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id =15 and CAI.CSR\_VENDOR='CISCO'

)

),eh\_gi as

(

select distinct csr\_device\_name, substr(csr\_device\_name, 1, 6) clli\_6

from eh\_nt\_issues eh

)

select nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, nvl(market, 'unknown') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, eh.clli\_6

, count (1) eh\_gi

from DOMAINS di

right outer join eh\_gi eh

on eh.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, eh.clli\_6 order by territory, market\_territory, sub\_market

;

cursor BRIX\_nt\_ISSUES is

with

domains as

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market,

leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> 'NNO'

and sub\_market <> 'OSS'

),

brix\_nt\_issues as (

select cai.csr\_device\_name

from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id in (1, 8, 33, 15) and CAI.CSR\_VENDOR='CISCO'

minus

(

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id = 33 and CAI.CSR\_VENDOR='CISCO'

intersect

select cai.csr\_device\_name from csr\_device\_audit\_issues cai

where cai.csr\_issue\_id = 15 and CAI.CSR\_VENDOR='CISCO'

)

),

brix\_gi as

(

select distinct csr\_device\_name, substr(csr\_device\_name, 1, 6) clli\_6

from brix\_nt\_issues

)

select nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, nvl(market, 'unknown') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, nvl(brixgi.clli\_6, 'un') clli\_6

, count (1) brix\_gi

from DOMAINS di

right outer join brix\_gi brixgi

on brixgi.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, brixgi.clli\_6 order by territory, market\_territory, sub\_market

;

cursor CSRs\_HPOV is

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market, leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> 'NNO'

and sub\_market <> 'OSS'

)

,

HPOV as

(

select distinct substr(device\_name, 1, 6) clli\_6, device\_name csr\_device\_name

from HPOV\_CSR\_WK hp

where device\_name like '%-CI-%'

)

select nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, nvl(market, 'unknown') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, hp.clli\_6

, count (1) hpov\_csrs

from DOMAINS di

right outer join HPOV hp

on hp.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, hp.clli\_6

;

CURSOR NCM\_HPOV\_MATCH\_CSRs is

WITH

NCM\_HPOV\_MATCH as

(

select csr\_device\_name

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

where aud.NCM\_CSR\_VLAN\_HOSTNAME is not null

and aud.HP\_CSR\_HOSTNAME is not null

)

select case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end clli\_6

, count(ma.csr\_device\_name) ncm\_hpov\_match

from NCM\_HPOV\_MATCH ma

group by case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end

;

CURSOR NCM\_nt\_MATCH\_CSRs is

WITH

NCM\_nt\_MATCH as

(

select csr\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end ncm\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end ncm\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

where aud.NCM\_CSR\_VLAN\_HOSTNAME is not null

and aud.nt\_DEVICE\_NAME is not null

minus

-- DUPS are NOT good

select aud.csr\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end ncm\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end ncm\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

join csr\_device\_audit\_issues cdai

on upper(cdai.csr\_device\_name) = upper(aud.csr\_device\_name)

where csr\_issue\_id = 1

and match\_code <> 'Nautilus Only'

)

select

case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end clli\_6

, count(ma.csr\_device\_name) ncm\_nt\_match

, sum(ma.ncm\_nt\_l\_match) ncm\_nt\_l\_match

, sum(ma.ncm\_nt\_nl\_match) ncm\_nt\_nl\_match

from NCM\_nt\_MATCH ma

group by

case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end

;

CURSOR EH\_nt\_MATCH\_CSRs is

WITH

EH\_nt\_MATCH as

(

select csr\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end eh\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end eh\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

where aud.eh\_csr\_hostname is not null

and aud.nt\_DEVICE\_NAME is not null

minus

-- DUPS are NOT good

select aud.csr\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end eh\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end eh\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

join csr\_device\_audit\_issues cdai

on upper(cdai.csr\_device\_name) = upper(aud.csr\_device\_name)

where csr\_issue\_id = 1

and match\_code <> 'Nautilus Only'

)

select

case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end clli\_6

, count(ma.csr\_device\_name) eh\_nt\_match

, sum(ma.eh\_nt\_l\_match) eh\_nt\_l\_match

, sum(ma.eh\_nt\_nl\_match) eh\_nt\_nl\_match

from eh\_nt\_MATCH ma

group by

case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end

;

CURSOR HPOV\_nt\_MATCH\_CSRs is

WITH

HPOV\_nt\_MATCH as

(

select csr\_device\_name --, match\_status, live\_in\_ng

, case when live\_in\_ng = 'Y' then 1 else 0 end hpov\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end hpov\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

where aud.HP\_CSR\_HOSTNAME is not null

and aud.nt\_DEVICE\_NAME is not null

minus

-- DUPS are NOT good

select aud.csr\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end ncm\_nt\_l\_match

, case when live\_in\_ng <> 'Y' then 1 else 0 end ncm\_nt\_nl\_match

from NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

join csr\_device\_audit\_issues cdai

on upper(cdai.csr\_device\_name) = upper(aud.csr\_device\_name)

where csr\_issue\_id = 1

and match\_code <> 'Nautilus Only'

)

select case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end clli\_6

, count(ma.csr\_device\_name) hpov\_nt\_match

, sum(ma.hpov\_nt\_l\_match) hpov\_nt\_l\_match

, sum(ma.hpov\_nt\_nl\_match) hpov\_nt\_nl\_match

from HPOV\_nt\_MATCH ma

group by case when ma.csr\_device\_name is not null then substr(ma.csr\_device\_name, 1, 6)

end

;

BEGIN

methodName := 'NCM\_CSR\_CLLI\_SUM\_LOAD';

EXECUTE IMMEDIATE 'truncate table NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

sqlStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, NT\_csrs, NT\_l\_csrs, NT\_nl\_csrs)

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf as market, leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> ''NNO''

and sub\_market <> ''OSS''

)

,

Xng as

(

select distinct csr\_device\_name, substr(csr\_device\_name, 1, 6) clli\_6

, nvl(case when inv\_status = ''LIVE'' then 1 else 0 end, 0) l

, nvl(case when inv\_status <> ''LIVE'' then 1 else 0 end, 0) nl

from NTLS\_CSR\_PARSED\_WK csr

join ng\_reports.equipment\_domain\_map edm

on edm.EQP\_REFERENCE\_ID = csr.EQP\_REFERENCE\_ID

join ng\_reports.equipment ei

on ei.EQP\_REFERENCE\_ID = csr.EQP\_REFERENCE\_ID

where csr\_device\_name like ''%-CI-%'' )

select nvl(territory, ''unknown'') territory, nvl(market\_territory, ''unknown'') market\_territory, nvl(sub\_market, ''unknown'') sub\_market, nvl(market, ''unknown'') market

, nvl(di.leaf\_domain\_id, 0) leaf\_domain\_id

, xng.clli\_6

, count (xng.csr\_device\_name) NT\_csrs

, sum (l) NT\_l\_csr

, sum (nl) NT\_nl\_csr

from DOMAINS di

right outer join Xng

on xng.clli\_6 = di.clli\_6

group by territory, market\_territory, sub\_market, market, di.leaf\_domain\_id

, xng.clli\_6

';

execute immediate sqlStmt;

commit;

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM set ncm\_csrs = :ncm\_csrs where CLLI\_6 = :CLLI';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, market, LEAF\_DOMAIN\_ID, CLLI\_6, ncm\_csrs)

values (:TERRITORY, :MARKET\_TERRITORY, :SUB\_MARKET, :market, :LEAF\_DOMAIN\_ID, :CLLI, :ncm\_csrs)';

for rec in CSRs\_NCM

loop

BEGIN

execute immediate updStmt using rec.ncm\_csrs, rec.CLLI;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for NCM CLLI'); dbms\_output.put\_line('FYI: should insert in NCM' || rec.TERRITORY ||' '|| rec.MARKET\_TERRITORY||' '|| rec.sub\_market||' '|| rec.market||' ' || rec.LEAF\_DOMAIN\_ID||' '|| rec.CLLI||' '|| rec.ncm\_csrs);

BEGIN

execute immediate insStmt using rec.TERRITORY, rec.MARKET\_TERRITORY, rec.SUB\_MARKET, rec.market, rec.LEAF\_DOMAIN\_ID, rec.CLLI, rec.ncm\_csrs;

--rollback;

--RAISE;

END;

end if;

EXCEPTION WHEN OTHERS then message := 'Error: (): Can''t update '|| rec.CLLI||' for NCM-CSR-extract in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message ||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set eh\_csrs = :eh\_csrs

where CLLI\_6 = :CLLI';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, market, LEAF\_DOMAIN\_ID, CLLI\_6, eh\_csrs)

values (:TERRITORY, :MARKET\_TERRITORY, :SUB\_MARKET, :market, :LEAF\_DOMAIN\_ID, :CLLI, :eh\_csrs)';

for rec in CSRs\_EHEALTH

loop

BEGIN

execute immediate updStmt using rec.eh\_csrs, rec.CLLI;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for ehealth CLLI');

dbms\_output.put\_line('FYI: should insert in EH' || rec.TERRITORY ||' '|| rec.MARKET\_TERRITORY||' '|| rec.sub\_market||' '|| rec.market||' ' || rec.LEAF\_DOMAIN\_ID||' '|| rec.CLLI||' '|| rec.eh\_csrs);

BEGIN

execute immediate insStmt using rec.TERRITORY, rec.MARKET\_TERRITORY, rec.SUB\_MARKET, rec.market , rec.LEAF\_DOMAIN\_ID, rec.CLLI, rec.eh\_csrs;

EXCEPTION

WHEN OTHERS then

message := 'Error: (): Can''t insert '|| rec.CLLI||' EH only CSR extract CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

EXCEPTION WHEN OTHERS then

message := 'Error: (): Can''t update '|| rec.CLLI||' forEH-CSR-extract in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set brix\_csrs = :brix\_csrs

where CLLI\_6 = :CLLI';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, market, LEAF\_DOMAIN\_ID, CLLI\_6, eh\_csrs)

values (:TERRITORY, :MARKET\_TERRITORY, :SUB\_MARKET, :market, :LEAF\_DOMAIN\_ID, :CLLI, :brix\_csrs)';

for rec in CSRs\_BRIX

loop

BEGIN

execute immediate updStmt using rec.brix\_csrs, rec.CLLI;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for ehealth CLLI');

dbms\_output.put\_line('FYI: should insert in EH' || rec.TERRITORY

||' '|| rec.MARKET\_TERRITORY||' '|| rec.sub\_market||' '||rec.market||' '

|| rec.LEAF\_DOMAIN\_ID||' '|| rec.CLLI||' '|| rec.brix\_csrs);

BEGIN

execute immediate insStmt using rec.TERRITORY, rec.MARKET\_TERRITORY, rec.SUB\_MARKET, rec.market

, rec.LEAF\_DOMAIN\_ID, rec.CLLI, rec.brix\_csrs;

EXCEPTION

WHEN OTHERS then

message := 'Error: (): Can''t insert '|| rec.CLLI||' BRIX only CSR extract CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

EXCEPTION WHEN OTHERS then

message := 'Error:(): Can''t update '|| rec.CLLI||' for BRIX-CSR-extract in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message ||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

/\*==============================================================

Loading/updating CSRs\_VLAN\_NCM

===============================================================

\*/

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set ncm\_csrs\_w\_vlan = :ncm\_csrs\_w\_vlan

where clli\_6 = :clli\_6

';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, ncm\_csrs\_w\_vlan)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :ncm\_csrs\_w\_vlan)

';

for rec in CSRs\_VLAN\_NCM

loop

BEGIN

execute immediate updStmt using rec.ncm\_csrs, rec.clli\_6;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for NCM CLLI');

dbms\_output.put\_line('FYI: should insert in NCM' || rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '||rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.clli\_6||' '|| rec.ncm\_csrs);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.ncm\_csrs;

EXCEPTION

WHEN OTHERS then

message := 'Error:(): Can''t insert '|| rec.clli\_6||' NCM-CSR-VLAN-extract CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: (): Can''t update '|| rec.clli\_6||' for NCM-CSR-VLAN-extract in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

/\*==============================================================

Loading/updating NCM\_CSRs\_W\_ISSUES

===============================================================\*/

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set ncm\_csrs\_w\_issues = :ncm\_csrs\_w\_issues

where clli\_6 = :clli\_6

';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, ncm\_csrs\_w\_issues)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :ncm\_csrs\_w\_issues)

';

for rec in NCM\_CSRs\_W\_ISSUES

loop

BEGIN

execute immediate updStmt using rec.ncm\_csrs\_w\_issues, rec.clli\_6;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for NCM CLLI');

dbms\_output.put\_line('FYI: should insert in NCM' || rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '||rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.clli\_6||' '|| rec.ncm\_csrs\_w\_issues);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.ncm\_csrs\_w\_issues;

EXCEPTION

WHEN OTHERS then

message := 'Error:(): Can''t insert '|| rec.clli\_6||' NCM-CSRs-W-ISSUES CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: Can''t update '|| rec.clli\_6||' for NCM-CSR-VLAN-extract in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

/\*============================================

Loading/updating NCM\_HPOV\_ISSUES

==============================================

\*/

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set ncm\_hpov\_issues = :ncm\_hpov\_issues

where clli\_6 = :clli\_6

';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, ncm\_hpov\_issues)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :ncm\_hpov\_issues)

';

for rec in NCM\_HPOV\_ISSUES

loop

BEGIN

execute immediate updStmt using rec.ncm\_hpov, rec.clli\_6;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for NCM CLLI');

dbms\_output.put\_line('FYI: should insert in NCM' || rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '||rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.clli\_6||' '|| rec.ncm\_hpov);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.ncm\_hpov;

EXCEPTION

WHEN OTHERS then

message := 'Error: (): Can''t insert '|| rec.clli\_6||' NCM\_HPOV\_ISSUES CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: : Can''t update '|| rec.clli\_6||' for NCM\_HPOV\_ISSUES in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

/\*

Loading/ updating NCM\_nt\_ISSUES

==============================================

\*/

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set ncm\_nt\_issues = :ncm\_nt\_issues

where clli\_6 = :clli\_6

';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, ncm\_nt\_issues)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :ncm\_nt\_issues)

';

for rec in NCM\_nt\_ISSUES

loop

BEGIN

execute immediate updStmt using rec.ncm\_gi, rec.clli\_6;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for NCM CLLI');

dbms\_output.put\_line('FYI: should insert in NCM' || rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '|| rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.clli\_6||' '|| rec.ncm\_gi);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.ncm\_gi;

EXCEPTION

WHEN OTHERS then

message := 'Error:: Can''t insert '|| rec.clli\_6||' NCM-NT-ISSUES CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: Can''t update '|| rec.clli\_6||' for NCM-NT-ISSUES in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

--

-- Loading/updating NCM\_nt\_ISSUES

--==============================================

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set ncm\_nt\_issues = :ncm\_nt\_issues

where clli\_6 = :clli\_6

';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, ncm\_nt\_issues)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :ncm\_nt\_issues)

';

for rec in NCM\_nt\_ISSUES

loop

BEGIN

execute immediate updStmt using rec.ncm\_gi, rec.clli\_6;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for NCM CLLI');

dbms\_output.put\_line('FYI: should insert in NCM' || rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '|| rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.clli\_6||' '|| rec.ncm\_gi);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.ncm\_gi;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t insert '|| rec.clli\_6||' NCM-NT-ISSUES CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '|| rec.clli\_6||' for NCM-NT-ISSUES in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

-- Loading/updating EH\_nt\_ISSUES

--==============================================

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM set eh\_nt\_issues = :eh\_nt\_issues where clli\_6 = :clli\_6';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, eh\_nt\_issues)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :eh\_nt\_issues)

';

for rec in EH\_nt\_ISSUES

loop

BEGIN

execute immediate updStmt using rec.eh\_gi, rec.clli\_6;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for NCM CLLI');

dbms\_output.put\_line('FYI: should insert in NCM' || rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '|| rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.clli\_6||' '|| rec.eh\_gi);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.eh\_gi;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t insert '|| rec.clli\_6||' EH-GI-ISSUES CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '|| rec.clli\_6||' for EH-GI-ISSUES in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

--============================================

-- Loading/updating BRIX\_nt\_ISSUES

--==============================================

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM set brix\_nt\_issues = :brix\_nt\_issues where clli\_6 = :clli\_6';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, brix\_nt\_issues)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :brix\_nt\_issues)

';

for rec in BRIX\_nt\_ISSUES

loop

BEGIN

execute immediate updStmt using rec.brix\_gi, rec.clli\_6;

l\_cnt := sql%rowcount;

if (l\_cnt = 0) then

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for BRIX CLLI');

dbms\_output.put\_line('FYI: should insert in NCM' ||rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '||rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.clli\_6||' '|| rec.brix\_gi);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.brix\_gi;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t insert '|| rec.clli\_6||'BRIX-GI-ISSUES CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

--RAISE;

END;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '|| rec.clli\_6||' for BRIX-GI-ISSUES in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

-- Loading/updating CSRs\_HPOV

--==============================================

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set hpov\_csrs = :hpov\_csrs

where clli\_6 = :clli\_6

';

insStmt := 'insert into NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id, clli\_6, hpov\_csrs)

values (:territory, :market\_territory, :sub\_market, :market, :leaf\_domain\_id, :clli\_6, :hpov\_csrs)

';

for rec in CSRs\_HPOV

loop

BEGIN

execute immediate updStmt using rec.hpov\_csrs, rec.clli\_6;

l\_cnt := sql%rowcount;

dbms\_output.put\_line('FYI: NCM updated Rows' || l\_cnt);

if (l\_cnt = 0) then

dbms\_output.put\_line('FYI: '||l\_cnt||' matching rows for HPOV CLLI');

dbms\_output.put\_line('FYI: should insert in HPOV' || rec.territory

||' '|| rec.market\_territory||' '|| rec.sub\_market||' '||rec.market||' '

|| rec.leaf\_domain\_id||' '|| rec.hpov\_csrs);

BEGIN

execute immediate insStmt using rec.territory, rec.market\_territory, rec.sub\_market, rec.market, rec.leaf\_domain\_id, rec.clli\_6, rec.hpov\_csrs;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t insert '|| rec.clli\_6||' HPOV only CLLI in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

end if;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '|| rec.clli\_6||' for HPOV in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

commit;

--============================================

-- updating NCM\_HPOV\_MATCH\_CSRs

--==============================================

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set ncm\_hpov\_match = :ncm\_hpov\_match

where clli\_6 = :clli\_6

';

for rec in NCM\_HPOV\_MATCH\_CSRs

loop

BEGIN

execute immediate updStmt using rec.ncm\_hpov\_match, rec.clli\_6;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update NCM-HPOV match in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM ';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

END LOOP;

-- updating NCM\_nt\_MATCH\_CSRs

--==============================================

-- calc matches / mismatches metric between ncm and gi

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set ncm\_nt\_match = :ncm\_nt\_match

, ncm\_nt\_l\_match = :ncm\_nt\_l\_match

, ncm\_nt\_nl\_match = :ncm\_nt\_nl\_match

where clli\_6 = :clli\_6

';

for rec in NCM\_nt\_MATCH\_CSRs

loop

BEGIN

execute immediate updStmt using rec.ncm\_nt\_match, rec.ncm\_nt\_l\_match, rec.ncm\_nt\_nl\_match, rec.clli\_6;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update NCM-GI mismatch in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

END LOOP;

commit;

--============================================

-- updating EH\_nt\_MATCH\_CSRs

--==============================================

-- calc matches / mismatches metric between eh and gi

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set eh\_nt\_match = :eh\_nt\_match

, eh\_nt\_l\_match = :eh\_nt\_l\_match

, eh\_nt\_nl\_match = :eh\_nt\_nl\_match

where clli\_6 = :clli\_6

';

for rec in EH\_nt\_MATCH\_CSRs

loop

BEGIN

execute immediate updStmt using rec.eh\_nt\_match, rec.eh\_nt\_l\_match, rec.eh\_nt\_nl\_match, rec.clli\_6;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update EH-GI mismatch in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

END LOOP;

commit;

--============================================

-- updating HPOV\_nt\_MATCH\_CSRs

--==============================================

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set hpov\_nt\_match = :hpov\_nt\_match

, hpov\_nt\_l\_match = :hpov\_nt\_l\_match

, hpov\_nt\_nl\_match = :hpov\_nt\_nl\_match

where clli\_6 = :clli\_6

';

for rec in HPOV\_nt\_MATCH\_CSRs

loop

BEGIN

execute immediate updStmt using rec.hpov\_nt\_match, rec.hpov\_nt\_l\_match, rec.hpov\_nt\_nl\_match, rec.clli\_6;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update HPOV match in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

END LOOP;

--============================================

-- updating BRIX\_nt\_MATCH\_CSRs

--==============================================

-- calc matches / mismatches metric between hpov and gi

updStmt := 'update NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

set brix\_nt\_match = :brix\_nt\_match

, brix\_nt\_l\_match = :brix\_nt\_l\_match

, brix\_nt\_nl\_match = :brix\_nt\_nl\_match

where clli\_6 = :clli\_6

';

for rec in BRIX\_nt\_MATCH\_CSRs

loop

BEGIN

execute immediate updStmt using rec.brix\_nt\_match, rec.brix\_nt\_l\_match, rec.brix\_nt\_nl\_match, rec.clli\_6;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update BRIX match in table: NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

END LOOP;

NCM\_CSR\_REGN\_SUM\_LOAD();

end NCM\_CSR\_CLLI\_SUM\_LOAD;

PROCEDURE NCM\_CSR\_REGN\_SUM\_LOAD is

sqlstmt VARCHAR(4000);

BEGIN

EXECUTE IMMEDIATE 'truncate table ncm\_hpov\_eh\_NT\_csr\_regn\_sum';

sqlstmt := 'insert into ncm\_hpov\_eh\_NT\_csr\_regn\_sum columns

(territory, market\_territory, sub\_market, ncm\_csrs, ncm\_csrs\_w\_vlan, hpov\_csrs,NT\_csrs, NT\_l\_csrs, NT\_nl\_csrs

, ncm\_csrs\_w\_issues

, ncm\_hpov\_match, ncm\_hpov\_issues

, ncm\_NT\_match, ncm\_NT\_l\_match, ncm\_NT\_nl\_match, ncm\_NT\_issues

, hpov\_NT\_match, hpov\_NT\_l\_match, hpov\_NT\_nl\_match, hpov\_NT\_issues

, eh\_csrs,eh\_NT\_match,eh\_NT\_l\_match,eh\_NT\_nl\_match,eh\_NT\_issues

, brix\_csrs, brix\_NT\_match, brix\_NT\_l\_match, brix\_NT\_nl\_match, brix\_NT\_issues

)select \* from(

select dlr.territory, dlr.market\_territory, dlr.sub\_market

, sum(ncm\_csrs) ncm\_csrs, sum(ncm\_csrs\_w\_vlan) ncm\_csrs\_w\_vlan

, sum(hpov\_csrs) hpov\_csrs

, sum(NT\_csrs) NT\_csrs, sum(NT\_l\_csrs) NT\_l\_csrs, sum(NT\_nl\_csrs) i\_nl\_csrs

, sum(ncm\_csrs\_w\_issues) ncm\_csrs\_w\_issues

, sum(ncm\_hpov\_match) ncm\_hpov\_match

, sum(ncm\_hpov\_issues) ncm\_hpov\_issues

, sum(ncm\_NT\_match) ncm\_NT\_match

, sum(ncm\_NT\_l\_match) ncm\_NT\_l\_match

, sum(ncm\_NT\_nl\_match) ncm\_NT\_nl\_match

, sum(ncm\_NT\_issues) ncm\_NT\_issues

, sum(hpov\_NT\_match) hpov\_NT\_match

, sum(hpov\_NT\_l\_match) hpov\_NT\_l\_match

, sum(hpov\_NT\_nl\_match) hpov\_NT\_nl\_match

, sum(hpov\_NT\_issues) hpov\_NT\_issues

, sum(eh\_csrs) eh\_csrs

, sum(eh\_NT\_match) eh\_NT\_match

, sum(eh\_NT\_l\_match) eh\_NT\_l\_match

, sum(eh\_NT\_nl\_match) eh\_NT\_nl\_match

, sum(eh\_NT\_issues) eh\_NT\_issues

, sum(brix\_csrs) brix\_csrs

, sum(brix\_NT\_match) brix\_NT\_match

, sum(brix\_NT\_l\_match) brix\_NT\_l\_match

, sum(brix\_NT\_nl\_match) brix\_NT\_nl\_match

, sum(brix\_NT\_issues) brix\_NT\_issues

from ng\_reports.domains\_regional\_reporting dlr

left outer join NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM metric

on metric.sub\_market = dlr.sub\_market

where dlr.sub\_market not in (''NNO'', ''OSS'')

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

order by dlr.territory, dlr.market\_territory, dlr.sub\_market )

where market\_territory != ''unknown'' or market\_territory is null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update ncm\_hpov\_eh\_NT\_csr\_regn\_sum wk set market\_territory =''unknown'',sub\_market=''unknown'' where territory=''unknown''' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

/\* sqlstmt:= 'update ncm\_hpov\_eh\_NT\_csr\_regn\_sum wk set TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update ncm\_hpov\_eh\_NT\_csr\_regn\_sum wk set TERRITORY\_MARKET\_SUB = market\_territory where sub\_market is null and market\_territory is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update ncm\_hpov\_eh\_NT\_csr\_regn\_sum wk set TERRITORY\_MARKET\_SUB = territory where sub\_market is null and market\_territory is null' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

\*/

END NCM\_CSR\_REGN\_SUM\_LOAD;

END NCM\_CSR\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package Body NCM\_CSR\_VLAN\_AUDIT\_PKG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."NCM\_CSR\_VLAN\_AUDIT\_PKG" AS

PROCEDURE load\_csr\_vlan\_paths is

sqlstmt VARCHAR2 (32767);

begin

EXECUTE IMMEDIATE ('truncate table ntls\_CSR\_vlan\_paths');

COMMIT;

sqlstmt := 'insert /\*+ parallel(6) \*/ into ntls\_CSR\_vlan\_paths columns ( eqp\_reference\_id, --ne\_inst\_id, removing EQUIP\_INST\_ID

card\_reference\_id,

trail\_id,

port\_bandwidth,

path\_type,

vlan\_inst\_id,

vlan\_name,

vlan\_number,

vlan\_status,

extract\_date

)

with

csr\_gige\_paths as (

select distinct XNE.EQP\_REFERENCE\_ID, p.card\_reference\_id --, p.port\_inst\_id, pli.leg\_inst\_id,(pli.sequence), PLI.LEG\_NAME

,cpi.trail\_id gige\_inst\_id, cpi.bandwidth, cpi.type

from ng\_reports.NTLS\_CSR\_PARSED\_WK xne,

ng\_reports.PORT p,

ng\_topology.trail cpi, --ng\_topology.trail cpi,

ng\_topology.trail\_component pli, --ng\_topology.trail\_component pli

ng\_topology.trail\_component\_element plm, --vzwnet.path\_leg\_member plm

ng\_topology.trail\_element cpe --vzwnet.circ\_path\_element cpe

WHERE

p.EQP\_REFERENCE\_ID = xne.EQP\_REFERENCE\_ID

and cpi.trail\_id in (

p.ckt\_path\_reference\_id, --p.trail\_id

p.next\_path\_reference\_Id --p.next\_path\_inst\_id

)

and cpi.BANDWIDTH like ''%Gbps''

and pli.TRAIL\_ID = cpi.trail\_id

and pli.component\_id = plm.component\_id --pli.leg\_inst\_id = plm.leg\_inst\_id

-- and plm.sequence = cpe.sequence

and plm.element\_id = cpe.element\_id

and cpe.element\_ref\_id =p.port\_reference\_id AND cpe.element\_type=''E''

and pli.sequence = (select min(sequence) sequence from ng\_topology.trail\_component li where LI.TRAIL\_ID=cpi.trail\_id)

),

gige\_vlan\_paths as (

select distinct gige.trail\_id gige\_inst\_id , VLAN.trail\_id vlan\_inst\_id

,vlan.trail\_name vlan\_hum\_id, regexp\_substr(regexp\_substr(vlan.trail\_name, ''VLAN(-|[[:space:]]|)\*[[:digit:]]+''), ''[[:digit:]]+'') vlan\_number

, vlan.status

from ng\_topology.trail gige, --ng\_topology.trail gige,

ng\_topology.trail vlan, --ng\_topology.trail vlan,

ng\_topology.trail\_component pli, --vzwnet.path\_leg\_inst pli,

ng\_topology.trail\_component\_element plm, --vzwnet.path\_leg\_member plm,

ng\_topology.trail\_element cpe --vzwnet.circ\_path\_element cpe

where gige.BANDWIDTH like ''%Gbps''

and vlan.BANDWIDTH = ''VLAN'' and vlan.TYPE = ''EBH''

and VLAN.trail\_id = CPE.trail\_id

and GIGE.trail\_id = CPE.parent\_trail\_id and cpe.element\_type = ''P''

and VLAN.trail\_id = pli.TRAIL\_ID

AND pli.component\_id = plm.component\_id

-- and plm.sequence = cpe.sequence

and plm.element\_id = cpe.element\_id

and pli.sequence=(select min(sequence) from ng\_topology.trail\_component pl where PL.trail\_id = vlan.trail\_id)

)select ngp.EQP\_REFERENCE\_ID, ngp.card\_reference\_id

,ngp.gige\_inst\_id trail\_id, ngp.bandwidth, ngp.type

, gvp.vlan\_inst\_id, gvp.vlan\_hum\_id vlan\_name, gvp.vlan\_number, gvp.status vlan\_status, sysdate

from csr\_gige\_paths ngp, gige\_vlan\_paths gvp

where ngp.gige\_inst\_id = gvp.gige\_inst\_id

';

execute immediate sqlStmt;

commit;

end load\_csr\_vlan\_paths;

PROCEDURE audit\_ci\_csr\_vlans

IS

methodname VARCHAR2 (30) := 'audit\_CI\_csr\_vlans ';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

BEGIN

EXECUTE IMMEDIATE ('delete ncm\_csr\_vlan\_audits where CSR\_vendor = ''CISCO''');

COMMIT;

sqlstmt :=

'insert /\*+ parallel(6) \*/ into ncm\_csr\_vlan\_audits columns (CSR\_Device\_Name, CSR\_vendor

, ne\_hostName, displayed\_name, device\_ip, ne\_vlan\_number

, match\_code, match\_status

, ng\_vlan\_number

, vlan\_inst\_id

, vlan\_status

, description

)

WITH

ci\_vlans AS

(SELECT xnvp.\*, xne.CSR\_DEVICE\_NAME

FROM ntls\_CSR\_vlan\_paths xnvp , NTLS\_CSR\_PARSED\_WK xne

where xne.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID and

xne.CSR\_DEVICE\_NAME like ''%-CI-%'' AND xnvp.vlan\_number IS NOT NULL and length(xnvp.vlan\_number) > 3),

matches AS

(SELECT aud.csr\_device\_name, ncm.VLAN\_NUMBER ne\_vlan\_number,

ncm.SUB\_INTERFACE\_NAME, ncm.DEVICE\_IP, ''BOTH'' match\_code,

NULL match\_status,

xnvp.vlan\_number ng\_vlan\_number, xnvp.vlan\_inst\_id,

xnvp.vlan\_status, aud.CSR\_VENDOR, ncm.description

FROM NCM\_CSR\_VLAN\_WK ncm JOIN NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

ON upper(aud.CSR\_DEVICE\_NAME) = upper(ncm.hostname)

JOIN ci\_vlans xnvp

ON aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID

WHERE ncm.port\_type = ''GigEthernet''

and REGEXP\_LIKE (ncm.port\_ip, ''^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$'')

AND ncm.VLAN\_NUMBER = xnvp.vlan\_number and

ncm.DEVICE\_IP IS NOT NULL and length(ncm.VLAN\_NUMBER) > 3 and ncm.DEVICE\_IP <> ''0.0.0.0'' and ncm.port\_ip <> ''0.0.0.0'' ),

ne\_only AS

(SELECT aud.CSR\_DEVICE\_NAME, ncm.SUB\_INTERFACE\_NAME, ncm.DEVICE\_IP,

ncm.VLAN\_NUMBER ne\_vlan\_number, ''NE Only'' match\_code,

''Missing in Granite'' match\_status, null,

null ng\_vlan\_number, NULL vlan\_inst\_id, NULL vlan\_status,

aud.CSR\_VENDOR, ncm.description

FROM NCM\_CSR\_VLAN\_WK ncm JOIN NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

ON aud.CSR\_DEVICE\_NAME = ncm.hostname

WHERE ncm.VLAN\_NUMBER IS NOT NULL

AND ncm.port\_type = ''GigEthernet''

and REGEXP\_LIKE (ncm.port\_ip, ''^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$'')

AND ncm.DEVICE\_IP IS NOT NULL and length(ncm.VLAN\_NUMBER) > 3 and ncm.DEVICE\_IP <> ''0.0.0.0'' and ncm.port\_ip <> ''0.0.0.0''

AND NOT EXISTS (

SELECT 1

FROM ci\_vlans xnvp

WHERE aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID

AND upper(xnvp.csr\_device\_name) = upper(aud.csr\_device\_name)

AND ncm.vlan\_number = xnvp.vlan\_number)),

xng\_only AS

(SELECT aud.csr\_device\_name, NULL SUB\_INTERFACE\_NAME, NULL DEVICE\_IP,

null ne\_vlan\_number, ''Nautilus Only'' match\_code,

''Missing in NE'' match\_status,

xnvp.vlan\_number ng\_vlan\_number, xnvp.vlan\_inst\_id, xnvp.vlan\_status,

aud.CSR\_VENDOR, Null description

FROM ci\_vlans xnvp JOIN NCM\_HPOV\_EH\_NT\_CI\_CSR\_AUDIT aud

ON aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID

AND AUD.CSR\_DEVICE\_NAME = xnvp.CSR\_DEVICE\_NAME

and not exists(select 1 from NCM\_CSR\_VLAN\_WK ncm

where upper(aud.csr\_device\_name) = upper(ncm.hostname)

AND ncm.VLAN\_NUMBER = xnvp.vlan\_number

and ncm.VLAN\_NUMBER IS NOT NULL and ncm.port\_type = ''GigEthernet'' and length(ncm.VLAN\_NUMBER) > 3 and ncm.DEVICE\_IP <> ''0.0.0.0'' and ncm.port\_ip <> ''0.0.0.0'' and REGEXP\_LIKE (ncm.port\_ip, ''^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$'')))

SELECT CSR\_DEVICE\_NAME, CSR\_vendor, CSR\_device\_name, SUB\_INTERFACE\_NAME,

DEVICE\_IP, ne\_vlan\_number, match\_code, match\_status,

ng\_vlan\_number, vlan\_inst\_id, vlan\_status, description

FROM matches

UNION

SELECT CSR\_DEVICE\_NAME, CSR\_vendor, CSR\_device\_name, SUB\_INTERFACE\_NAME,

DEVICE\_IP, ne\_vlan\_number, match\_code, match\_status,

ng\_vlan\_number, vlan\_inst\_id, vlan\_status, description

FROM ne\_only

UNION

SELECT CSR\_DEVICE\_NAME, CSR\_vendor, NULL, SUB\_INTERFACE\_NAME, DEVICE\_IP,

ne\_vlan\_number, match\_code, match\_status, ng\_vlan\_number,

vlan\_inst\_id, vlan\_status, description

FROM xng\_only

';

EXECUTE IMMEDIATE (sqlstmt);

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE := 'Error: ' || methodname || '(): Cant audit CI VLANs';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

/\*watchdog.logerror (methodName,

8000,

SUBSTR (message

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

),

'Y'

); \*/

RAISE;

END;

insert\_ci\_csr\_vlan\_issues ();

assign\_area\_region\_to\_csr\_vlan();

END audit\_ci\_csr\_vlans;

PROCEDURE insert\_ci\_csr\_vlan\_issues

IS

methodname VARCHAR2 (30) := 'insert\_CI\_CSR\_vlan\_issues';

MESSAGE VARCHAR2 (300);

sqlstmt VARCHAR2 (32767);

CURSOR dup\_l\_vlan

IS

-- if the same DEVICE\_NAME, nxa.VLAN\_number has > 1 LIVE status then

-- mark it as dup

WITH vlan\_status AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status,

match\_code, nxa.vlan\_inst\_id

FROM ncm\_csr\_vlan\_audits nxa JOIN ng\_topology.trail cpi

ON cpi.trail\_id = nxa.vlan\_inst\_id

),

dup\_l\_vlans AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number,

COUNT (vlan\_inst\_id)

FROM vlan\_status nxa

WHERE status = 'LIVE' AND match\_code <> 'Nautilus Only'

GROUP BY nxa.csr\_device\_name, nxa.ng\_vlan\_number, status

HAVING COUNT (vlan\_inst\_id) > 1)

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status,

match\_code, nxa.vlan\_inst\_id

FROM dup\_l\_vlans dup JOIN vlan\_status nxa

ON upper(nxa.csr\_device\_name) = upper(dup.csr\_device\_name)

AND nxa.ng\_vlan\_number = dup.ng\_vlan\_number

ORDER BY nxa.csr\_device\_name, nxa.ng\_vlan\_number;

-- if the same csr\_device\_name, nxa.VLAN\_number has > 1 status then

-- and 1 is LIVE and other is not then mark NL as ''NE Only'' & ''LIVE path exists''

CURSOR LIVE\_vlan\_exists

IS

WITH vlan\_status AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status,

match\_code, nxa.vlan\_inst\_id

FROM ncm\_csr\_vlan\_audits nxa JOIN ng\_topology.trail cpi

ON cpi.trail\_id = nxa.vlan\_inst\_id

),

l\_and\_nl\_vlans AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status = 'LIVE'

INTERSECT

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status <> 'LIVE')

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, nxa.vlan\_inst\_id

FROM l\_and\_nl\_vlans dup JOIN vlan\_status nxa

ON upper(nxa.csr\_device\_name) = upper(dup.csr\_device\_name)

AND nxa.ng\_vlan\_number = dup.ng\_vlan\_number

WHERE nxa.status <> 'LIVE'

ORDER BY nxa.csr\_device\_name, nxa.ng\_vlan\_number;

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

BEGIN

EXECUTE IMMEDIATE ('delete CSR\_VLAN\_AUDIT\_ISSUES where csr\_vendor = ''CISCO''');

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant delete rows in table: CSR\_VLAN\_AUDIT\_ISSUES';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

-- mark duplicate in Granite as NE Only, or LIVE Path Exists

-- if there is a path revision then the same VLAN will be present twice.

-- BUT that is NOT a duplicate

-- if NOT a revision path and VLAN# is present multiple times THEN it is dup

-- mark duplicate in Granite as NE Only, or LIVE Path Exists

-- else 2 DUP vlans on a csr then Duplicate

sqlstmt :=

'update ncm\_csr\_vlan\_audits

set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

, match\_status = nvl2(match\_status, match\_status|| '', Dup in NT'', ''Dup in NT'')

where vlan\_inst\_id = :p\_vlan\_inst\_id

and ng\_vlan\_number = :p\_xng\_vlan\_number

and upper(CSR\_DEVICE\_NAME) = upper(:p\_csr\_device\_name)

';

FOR rec IN dup\_l\_vlan

LOOP

BEGIN

EXECUTE IMMEDIATE sqlstmt

USING rec.vlan\_inst\_id,

rec.ng\_vlan\_number,

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant update ''Dup in NT'' in table: ncm\_csr\_vlan\_audits';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

COMMIT;

BEGIN

sqlstmt :=

'insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

select distinct W.CSR\_DEVICE\_NAME

, case when ne\_vlan\_number is null then ng\_vlan\_number

else ne\_vlan\_number

end vlan\_number,1,''CISCO''

from ncm\_csr\_vlan\_audits w

where W.MATCH\_STATUS like ''%Dup in NT%''';

EXECUTE IMMEDIATE sqlstmt;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant insert into CSR\_VLAN\_AUDIT\_ISSUES issue Duplicate in NT ';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

COMMIT;

-- BEGIN

-- -- check the ne\_hostname matches MTCE std, else update match\_code = 'NE Only'

-- -- and match\_status = 'ne\_hostname not MTCE Stds compliant'

-- sqlstmt :=

-- ' update ncm\_csr\_vlan\_audits aud set match\_code = case when match\_code <> 'Nautilus Only' then 'NE Only' else 'Nautilus Only' end

-- where exists (select 1

-- from NCM\_CSR\_VLAN\_WK i

-- where upper(AUD.CSR\_DEVICE\_NAME) = upper(I.HOSTNAME)

-- and port\_type = 'GigEthernet'

-- and length(i.VLAN\_NUMBER) > 3

-- and i.DEVICE\_IP <> '0.0.0.0' and i.port\_ip <> '0.0.0.0'

-- AND REGEXP\_LIKE (i.port\_ip, '^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$')

-- and parse\_status like '%11%')';

--

-- EXECUTE IMMEDIATE sqlstmt;

--

-- COMMIT;

-- sqlstmt :=

-- 'insert into CSR\_VLAN\_AUDIT\_ISSUES

-- columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

-- select distinct W.CSR\_DEVICE\_NAME

-- , case when ne\_vlan\_number is null then ng\_vlan\_number

-- else ne\_vlan\_number

-- end vlan\_number,17,'CISCO'

-- from ncm\_csr\_vlan\_audits w, NCM\_CSR\_VLAN\_WK n

-- where

-- W.CSR\_VENDOR = 'CISCO'

-- and upper(W.CSR\_DEVICE\_NAME) = upper(N.HOSTNAME)

-- and n.port\_type = 'GigEthernet'

-- AND REGEXP\_LIKE (n.port\_ip, '^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$')

-- and length(n.VLAN\_NUMBER) > 3

-- and n.DEVICE\_IP <> '0.0.0.0' and n.port\_ip <> '0.0.0.0'

-- and n.parse\_status like '%11%'';

--

-- --and regexp\_substr(parse\_status, '[[:digit:]]+') = '11'

-- EXECUTE IMMEDIATE sqlstmt;

--

-- COMMIT;

-- EXCEPTION

-- WHEN OTHERS

-- THEN

-- MESSAGE :=

-- 'Error: '

-- || methodname

-- || '(): Can't update name does not match MTCE Std';

-- DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

-- || TO\_CHAR (SQLCODE)

-- || ': '

-- || SQLERRM,

-- 1,

-- 255

-- )

-- );

-- RAISE;

-- END;

BEGIN

-- check if ne\_hostnames' don't match first 6-char match some MTSO CLLI

-- then match\_code = 'NE Only' and match\_status = '1st 6-chars of don''t match any MTSO CLLIs'

sqlstmt :=

'insert into CSR\_VLAN\_AUDIT\_ISSUES columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT ei.CSR\_device\_name,

SUBSTR (ei.csr\_device\_name, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM ncm\_csr\_vlan\_audits ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.csr\_device\_name, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.csr\_device\_name, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.csr\_device\_name

FROM ncm\_csr\_vlan\_audits ei

MINUS

SELECT ei.csr\_device\_name

FROM ncm\_csr\_vlan\_audits ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.csr\_device\_name, 1, 8)

MINUS

SELECT ei.csr\_device\_name

FROM ncm\_csr\_vlan\_audits ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.csr\_device\_name, 1, 6)

),

ISSUES AS

(

SELECT cm.csr\_device\_name, 22 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.csr\_device\_name, 2 issue\_id

FROM bad\_clli bc

)

SELECT distinct i.csr\_device\_name, W.NE\_VLAN\_NUMBER, i.issue\_id, ''CISCO''

FROM ISSUES i, ncm\_csr\_vlan\_audits w

where upper(i.csr\_device\_name) = upper(W.CSR\_DEVICE\_NAME)

and W.NE\_VLAN\_NUMBER is not null

';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND

THEN

NULL;

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant mark 1st 6-char dont match MTSO CLLIs';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- device not found in sam vlan extract

sqlstmt :=

' insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

select distinct aud.CSR\_DEVICE\_NAME, ng\_vlan\_number, 13, ''CISCO''

from ncm\_csr\_vlan\_audits aud

where aud.ne\_hostname is null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant insert issue Not in NCM VLAN Extract';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- device not found in NT

sqlstmt :=

' insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

select distinct aud.CSR\_DEVICE\_NAME, ne\_vlan\_number, 15, ''CISCO''

from ncm\_csr\_vlan\_audits aud

where aud.vlan\_inst\_id is null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname

|| '(): Cant insert issue Not in NT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- duplicate in NE

sqlstmt :=

' insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

SELECT csr\_device\_name, ne\_vlan\_number, 10, ''CISCO''

FROM ncm\_csr\_vlan\_audits

WHERE ne\_vlan\_number > 0

GROUP BY csr\_device\_name, ne\_vlan\_number

HAVING COUNT (\*) > 1';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant insert issue Duplicate in NE';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

-- if there is a path revision then the same VLAN will be present twice.

-- BUT that is NOT a duplicate

-- if NOT a revision path and VLAN# is present multiple times THEN it is dup

-- if 1 == LIVE and the other <> LIVE then mark <> LIVE as 'LIVE in Xng'

sqlstmt :=

'update ncm\_csr\_vlan\_audits

set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

, match\_status = nvl2(match\_status, ''LIVE Path Exists,'' ||match\_status , ''LIVE Path Exists'')

where vlan\_inst\_id = :p\_vlan\_inst\_id

and ng\_vlan\_number = :p\_xng\_vlan\_number

and upper(CSR\_DEVICE\_NAME) = upper(:p\_csr\_device\_name)

--and csr\_vendor = :p\_csr\_vendor

';

FOR rec IN LIVE\_vlan\_exists

LOOP

BEGIN

EXECUTE IMMEDIATE sqlstmt

USING rec.vlan\_inst\_id,

rec.ng\_vlan\_number,

rec.csr\_device\_name; --, p\_csr\_vendor;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant update LIVE Path Exists in table: ncm\_csr\_vlan\_audits';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

COMMIT;

BEGIN

-- IF BOTH and has issues then mark it NE

sqlstmt :=

' update ncm\_csr\_vlan\_audits aud

set match\_code = ''NE Only''

where exists (select 1

from

CSR\_VLAN\_AUDIT\_ISSUES vi,

CSR\_issue n

where aud.MATCH\_CODE = ''BOTH''

and aud.csr\_vendor = ''CISCO''

and VI.CSR\_VENDOR=''CISCO''

and upper(AUD.CSR\_DEVICE\_NAME) = upper(VI.CSR\_DEVICE\_NAME)

and AUD.NE\_VLAN\_NUMBER = VI.VLAN\_NUMBER

and N.IS\_CRITICAL=''Y''

and N.CSR\_ISSUE\_ID = vi.CSR\_ISSUE\_ID

)';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant update match\_status to NE Only';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- device not found in NCM vlan extract

sqlstmt :=

' insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

WITH vlan\_status AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status, match\_code,

nxa.vlan\_inst\_id

FROM ncm\_csr\_vlan\_audits nxa JOIN ng\_topology.trail cpi

ON cpi.trail\_id = nxa.vlan\_inst\_id

),

l\_and\_nl\_vlans AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status = ''LIVE''

INTERSECT

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status <> ''LIVE''),

LIVE\_vlan\_exists AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, nxa.vlan\_inst\_id

FROM l\_and\_nl\_vlans dup JOIN vlan\_status nxa

ON upper(nxa.csr\_device\_name) = upper(dup.csr\_device\_name)

AND nxa.ng\_vlan\_number = dup.ng\_vlan\_number

WHERE nxa.status <> ''LIVE'')

SELECT DISTINCT aud.csr\_device\_name, ng\_vlan\_number, 31, ''CISCO''

FROM ncm\_csr\_vlan\_audits aud

WHERE vlan\_inst\_id IS NOT NULL

AND match\_code = ''NE Only''

AND NOT EXISTS (

SELECT 1

FROM CSR\_VLAN\_AUDIT\_ISSUES nvai

WHERE nvai.vlan\_number = aud.ng\_vlan\_number

AND upper(aud.csr\_device\_name) = upper(nvai.csr\_device\_name)

AND nvai.csr\_vendor = ''CISCO'')

AND EXISTS (

SELECT 1

FROM LIVE\_vlan\_exists lle

WHERE upper(lle.csr\_device\_name) = upper(aud.csr\_device\_name)

AND aud.ng\_vlan\_number = lle.ng\_vlan\_number)';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant insert issue LIVE VLAN Exists';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

END insert\_ci\_csr\_vlan\_issues;

PROCEDURE assign\_area\_region\_to\_csr\_vlan

IS

methodname VARCHAR2 (30) := 'assign\_area\_region\_to\_csr\_vlan';

MESSAGE VARCHAR2 (300);

updstmt VARCHAR2 (32767);

processname VARCHAR2 (30) := 'NCM\_GI\_CSR\_VLAN\_AUDIT';

CURSOR area\_region\_dev

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory,

sub\_market, sub\_market\_leaf, LEAF\_DOMAIN\_ID

FROM clli\_domain\_map\_v

WHERE market\_territory <> 'NNO' AND market\_territory <> 'OSS')

SELECT di.territory, di.market\_territory, di.sub\_market, di.sub\_market\_leaf, di.LEAF\_DOMAIN\_ID,

csr\_device\_name, clli\_6

FROM ncm\_csr\_vlan\_audits aud JOIN domains di

ON di.clli\_6 = SUBSTR (aud.csr\_device\_name, 1, 6)

;

BEGIN

updstmt :=

'update ncm\_csr\_vlan\_audits set territory = :territory

, market\_territory = :market\_territory

, sub\_market = :sub\_market

, market = :sub\_market\_leaf

, LEAF\_DOMAIN\_ID = :LEAF\_DOMAIN\_ID

, clli = :clli\_6

where csr\_device\_name = :csr\_device\_name';

FOR rec IN area\_region\_dev

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.LEAF\_DOMAIN\_ID,

rec.clli\_6,

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant update clli for '

|| rec.csr\_device\_name

|| ' in table: ncm\_csr\_vlan\_audits';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

BEGIN

-- if there still are some devices which cant be mapped to any domain then mark those as unknown

updstmt :=

' update ncm\_csr\_vlan\_audits

set territory = ''Unknown''

, market\_territory = ''Unknown''

, sub\_market = ''Unknown''

, market = ''Unknown''

, LEAF\_DOMAIN\_ID = 0

, clli = SUBSTR (csr\_device\_name, 1, 6)

where territory is null ';

EXECUTE IMMEDIATE updstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant update area/region from to unknown in table: ncm\_csr\_vlan\_audits';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END assign\_area\_region\_to\_csr\_vlan;

PROCEDURE load\_vlan\_csr\_clli\_summary

IS

methodname VARCHAR2 (30) := 'load\_vlan\_csr\_clli\_summary';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

EXECUTE IMMEDIATE 'truncate table NCM\_CSR\_VLAN\_CLLI\_SUMM';

-- generate device summary

sqlstmt :=

' insert into NCM\_CSR\_VLAN\_CLLI\_SUMM

columns (territory, market\_territory, sub\_market, market

, clli

, ne\_total\_csr\_vlan

, matched\_l\_csr\_vlan

, matched\_nl\_csr\_vlan

, mismatched\_csr\_vlan

, nt\_nl\_2\_total\_vlan\_pct

, vlan\_COMPLIANCE)

WITH

csr as

(

select distinct territory, market\_territory

, sub\_market, market, LEAF\_DOMAIN\_ID, clli

, csr\_device\_name, csr\_vendor

from ncm\_csr\_vlan\_audits ng

where ng.CSR\_DEVICE\_NAME is not null

),

csr\_VLAN as

(

select distinct hostname CSR\_DEVICE\_NAME, VLAN\_NUMBER

from NCM\_CSR\_VLAN\_WK ng

where ng.VLAN\_NUMBER > 0 and port\_type = ''GigEthernet''

and REGEXP\_LIKE (ng.port\_ip, ''^([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})$'')

and length(ng.VLAN\_NUMBER) > 3

and ng.DEVICE\_IP <> ''0.0.0.0'' and ng.port\_ip <> ''0.0.0.0''

)

,

CNT\_csr\_VLANs as

(

select distinct ng.CSR\_DEVICE\_NAME, count(vlan\_number) total\_ne\_csr\_vlan

from csr\_VLAN nv

join csr ng

on upper(ng.CSR\_DEVICE\_NAME) = upper(nv.CSR\_DEVICE\_NAME)

group by ng.CSR\_DEVICE\_NAME

),

LIVE\_match as

(

select distinct aud.CSR\_DEVICE\_NAME, aud.ne\_vlan\_number, vlan\_status

from ncm\_csr\_vlan\_audits aud

where match\_code = ''BOTH''

and aud.csr\_vendor = ''CISCO''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status = ''LIVE''

)

,nonLIVE\_match as

(

select distinct aud.csr\_DEVICE\_NAME, aud.ne\_vlan\_number

from ncm\_csr\_vlan\_audits aud

where match\_code = ''BOTH''

and aud.csr\_vendor = ''CISCO''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status <> ''LIVE''

minus

select distinct aud.csr\_DEVICE\_NAME, aud.ne\_vlan\_number

from ncm\_csr\_vlan\_audits aud

where match\_code = ''BOTH''

and aud.csr\_vendor = ''CISCO''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status = ''LIVE''

) ,

CNT\_BOTH\_LIVE\_VLANs as

( -- good matches

select csr\_DEVICE\_NAME, nvl (count(csr\_DEVICE\_NAME), 0) matched\_l\_csr\_vlan

from LIVE\_match aud

group by csr\_DEVICE\_NAME

),

CNT\_BOTH\_NONLIVE\_VLANs as

( -- good matches

select csr\_DEVICE\_NAME, nvl( count(csr\_DEVICE\_NAME), 0) matched\_nl\_csr\_vlan

-- , nvl(sum(case when vlan\_status = ''LIVE'' then 1 else 0 end), 0) matched\_l\_csr\_vlan

-- , nvl(sum(case when vlan\_status <> ''LIVE'' then 1 else 0 end), 0) matched\_nl\_csr\_vlan

from nonLIVE\_match aud

group by aud.csr\_DEVICE\_NAME

) ,

MISMATCH\_VLANs as

( -- bad matches

select distinct aud.csr\_DEVICE\_NAME, aud.ne\_vlan\_number, aud.ng\_vlan\_number, vlan\_status

from ncm\_csr\_vlan\_audits aud

where match\_code <> ''BOTH''

and aud.csr\_vendor = ''CISCO''

and (aud.ne\_vlan\_number > 0 or aud.ng\_vlan\_number > 0)

AND NOT EXISTS (

SELECT 1

FROM CSR\_VLAN\_AUDIT\_ISSUES nvai

WHERE nvai.vlan\_number = aud.ng\_vlan\_number

AND upper(aud.csr\_device\_name) = upper(nvai.csr\_device\_name)

AND nvai.csr\_vendor = ''CISCO'' and nvai.CSR\_ISSUE\_ID = 31)

)

,

CNT\_MISMATCH\_VLANs as

( -- bad match cnts

select aud.CSR\_DEVICE\_NAME

, count(1) mismatch\_cnt

from MISMATCH\_VLANs aud

group by aud.csr\_DEVICE\_NAME

)

select territory, market\_territory, sub\_market, market,clli

, sum(nvl(nv.total\_ne\_csr\_vlan, 0)) ne\_total\_csr\_vlan

, sum(nvl(matched\_l\_csr\_vlan, 0)) matched\_l\_csr\_vlan

, sum(nvl(matched\_nl\_csr\_vlan, 0)) matched\_nl\_csr\_vlan

, sum(nvl(mismatch\_cnt, 0)) mismatched\_csr\_vlan

, round(sum(NVL(matched\_nl\_csr\_vlan, 0)) / decode(sum(NVL(matched\_l\_csr\_vlan, 0)), 0, 1, sum(nvl(matched\_l\_csr\_vlan, 0)))

\* 100, 2) nt\_nl\_2\_total\_vlan\_pct

, round (sum(nvl(matched\_l\_csr\_vlan,0)) / decode (sum(nvl( nv.total\_ne\_csr\_vlan, 0)) - sum(nvl(matched\_nl\_csr\_vlan, 0)), 0 , 1

, sum(nvl(nv.total\_ne\_csr\_vlan,0)) - sum(nvl(matched\_nl\_csr\_vlan,0)))

\* 100 , 2) vlan\_COMPLIANCE

from csr ng

left outer join CNT\_csr\_VLANs nv

on upper(nv.csr\_DEVICE\_NAME) = upper(ng.csr\_DEVICE\_NAME)

left outer join CNT\_BOTH\_LIVE\_VLANs lmv

on upper(lmv.csr\_DEVICE\_NAME) = upper(ng.csr\_DEVICE\_NAME)

left outer join CNT\_BOTH\_NONLIVE\_VLANs nlmv

on upper(nlmv.csr\_DEVICE\_NAME) = upper(ng.csr\_DEVICE\_NAME)

left outer join CNT\_MISMATCH\_VLANs mv

on upper(mv.csr\_DEVICE\_NAME) = upper(ng.csr\_DEVICE\_NAME)

group by ng.territory, ng.market\_territory, ng.sub\_market, ng.market, csr\_Vendor, clli';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname || '(): Cant generate clli summary';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

load\_csr\_vlan\_regional\_summ();

END load\_vlan\_csr\_clli\_summary;

PROCEDURE load\_csr\_vlan\_regional\_summ

IS

methodname VARCHAR2 (30) := 'load\_vlan\_regional\_summary';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

EXECUTE IMMEDIATE 'truncate table NCM\_CSR\_VLAN\_REGION\_SUMM';

-- generate device summary

sqlstmt :=

' insert into NCM\_CSR\_VLAN\_REGION\_SUMM

columns (territory, market\_territory, sub\_market

, ne\_total\_csr\_vlan

, matched\_l\_csr\_vlan

, matched\_nl\_csr\_vlan

, mismatched\_csr\_vlan

, nt\_nl\_2\_total\_vlan\_pct

, vlan\_compliance)

select dlr.territory, dlr.market\_territory, dlr.sub\_market

, sum(ne\_total\_csr\_vlan) ne\_total\_csr\_vlan

, sum(matched\_l\_csr\_vlan) matched\_l\_csr\_vlan

, sum(matched\_nl\_csr\_vlan) matched\_nl\_csr\_vlan

, sum(mismatched\_csr\_vlan) mismatched\_csr\_vlan

, round(avg(nt\_nl\_2\_total\_vlan\_pct), 2) nt\_nl\_2\_total\_vlan\_pct

, round(avg(vlan\_COMPLIANCE), 2) vlan\_compliance

from domains\_leaf\_reporting dlr

left outer join NCM\_CSR\_VLAN\_CLLI\_SUMM ng

on dlr.sub\_market\_leaf = ng.market

where dlr.territory not in (''NNO'', ''OSS'')

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

ORDER BY territory, market\_territory, sub\_market ';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt:= 'update NCM\_CSR\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update NCM\_CSR\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = market\_territory where sub\_market is null and market\_territory is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update NCM\_CSR\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = territory where sub\_market is null and market\_territory is null' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

-- sqlstmt:= 'update NCM\_CSR\_VLAN\_REGION\_SUMM\_WK wk set market\_territory =''Unknown'',sub\_market=''Unknown'' where territory=''Unknown'' or territory=''unknown''' ;

-- EXECUTE IMMEDIATE sqlstmt;

commit;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname || '(): Cant generate regional summary';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END load\_csr\_vlan\_regional\_summ;

END NCM\_CSR\_VLAN\_AUDIT\_PKG;

/

--------------------------------------------------------

-- DDL for Package Body NETSMART\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."NETSMART\_AUDIT" AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NAME: netsmart\_audit

PURPOSE:

REVISIONS:

Ver Date Author Description

--------- ---------- --------------- ------------------------------------

1.0 5/2/2012 NSP Created this package.

1.1 5/11/2012 SME Realized detail query and

added report table functions

1.2 3/9/2015 SME Split audit into MSPP and ADM

methods.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure GetNautilusEquip \*

\* \*

\* Purpose: Finds all the top level Fujitsu FlashWave MSPP Equipment in Nautilus Inventory \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE GETNAUTILUSEQUIP IS

BEGIN

EXECUTE IMMEDIATE 'TRUNCATE TABLE NG\_FW\_EQUIP';

INSERT INTO NG\_FW\_EQUIP ( NE\_INST\_ID, EQP\_REFERENCE\_ID, SITE\_REFERENCE\_ID,

PARENT\_EQP\_REFERENCE\_ID, EQ\_CLASS\_TYPE, EQP\_NAME, EQP\_MODEL, EQP\_TYPE, INV\_STATUS, EQP\_VENDOR, CONTAINER

)

SELECT TO\_NUMBER

(TRIM

(BOTH '/' FROM SYS\_CONNECT\_BY\_PATH

(DECODE (LEVEL,

1, EI.EQP\_REFERENCE\_ID,

''

),

'/'

)

)

) NE\_INST\_ID,

EI.EQP\_REFERENCE\_ID,

EI.SITE\_REFERENCE\_ID,

EI.PARENT\_EQP\_REFERENCE\_ID,

DECODE (LEVEL,

1, 'NE',

DECODE (EI.CONTAINER, 'SHELF', 'SHELF', 'N/A') --------------CONTAINER ---???

) EQ\_CLASS\_EQP\_TYPE,

EI.EQP\_NAME ,

EI.EQP\_MODEL ,

EI.EQP\_TYPE,

EI.INV\_STATUS,

EI.EQP\_VENDOR ,

EI.CONTAINER

FROM NG\_REPORTS.EQUIPMENT EI

WHERE (LEVEL = 1 OR EI.CONTAINER= 'SHELF')

START WITH EI.EQP\_VENDOR = 'FUJITSU'

AND REGEXP\_LIKE (EQP\_MODEL, 'FLASHWAVE|FW')

AND REGEXP\_LIKE (EQP\_NAME, '([[:alnum:]]{1,4}\s+)?[[:alnum:]]{8}MS\d-(R|N)FW\d{5}')

CONNECT BY PRIOR EI.EQP\_REFERENCE\_ID = EI.PARENT\_EQP\_REFERENCE\_ID;

COMMIT;

-- since we dont know top-equip from where to start looking for all children equip

-- in the previous start-with-connect-by the same equip might come is as both NE and SHELF.

-- if it does then delete those rows where it is a NE.

DELETE NG\_FW\_EQUIP XFE

WHERE XFE.CONTAINER = 'NE'

AND EXISTS (

WITH

TO\_BE\_DELETED AS

(

SELECT EQP\_REFERENCE\_ID

FROM NG\_FW\_EQUIP

WHERE CONTAINER = 'SHELF'

INTERSECT

SELECT EQP\_REFERENCE\_ID

FROM NG\_FW\_EQUIP

WHERE CONTAINER = 'NE'

)

SELECT 1

FROM TO\_BE\_DELETED TBD

WHERE TBD.EQP\_REFERENCE\_ID = XFE.EQP\_REFERENCE\_ID

);

COMMIT;

-- delete dups if some EQP\_REFERENCE\_ID comes in > 1 as SHELF

/\*select EQP\_REFERENCE\_ID

from NG\_fw\_equip

where eq\_class\_EQP\_TYPE = ''SHELF''

group by EQP\_REFERENCE\_ID

having count(1) > 1

;

commit;

\*/

-- update the NG NE rows that dont match Nautilus naming format

UPDATE NG\_FW\_EQUIP SET PARSE\_STATUS = 'Name not Nautilus Stds compliant'

WHERE NOT REGEXP\_LIKE(EQP\_NAME, '^([[:alnum:]]{1,4}\s+)?[[:alnum:]]{8}MS\d-(R|N)FW\d{5}(\s\*\[[^][]+\])?$') -- Nautilus std name

AND EQ\_CLASS\_TYPE = 'NE';

COMMIT;

END GETNAUTILUSEQUIP;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure GetADMNautilusEquip \*

\* \*

\* Purpose: Finds all the top level Fujitsu FlashWave ADM Equipment in Nautilus Inventory \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE GETADMNAUTILUSEQUIP IS

BEGIN

-- from Nautilus get all equip that have EQP\_VENDOR = FUJITSU and EQP\_MODEL like 'FLASHWAVE% or 'FW%' and is not a Flashwave MSPP device

EXECUTE IMMEDIATE 'TRUNCATE TABLE NG\_ADM\_FW\_EQUIP';

INSERT INTO NG\_ADM\_FW\_EQUIP ( NE\_INST\_ID, EQP\_REFERENCE\_ID, SITE\_REFERENCE\_ID,

PARENT\_EQP\_REFERENCE\_ID, EQ\_CLASS\_TYPE, EQP\_NAME, EQP\_MODEL, EQP\_TYPE,INV\_STATUS, EQP\_VENDOR, CONTAINER

)

SELECT TO\_NUMBER

(TRIM

(BOTH '/' FROM SYS\_CONNECT\_BY\_PATH

(DECODE (LEVEL,

1, EI.EQP\_REFERENCE\_ID,

''

),

'/'

)

)

) NE\_INST\_ID,

EI.EQP\_REFERENCE\_ID,

EI.SITE\_REFERENCE\_ID,

EI.PARENT\_EQP\_REFERENCE\_ID,

DECODE (LEVEL,

1, 'NE',

DECODE (EI.CONTAINER,'SHELF', 'SHELF', 'N/A')

) EQ\_CLASS\_TYPE,

EI.EQP\_NAME,

EI.EQP\_VENDOR,

EI.EQP\_TYPE,

EI.INV\_STATUS,

EI.EQP\_VENDOR ,

EI.CONTAINER

FROM NG\_REPORTS.EQUIPMENT EI

WHERE (LEVEL = 1 OR EI.CONTAINER= 'SHELF')

START WITH EI.EQP\_VENDOR = 'FUJITSU'

AND REGEXP\_LIKE (EQP\_MODEL, 'FLASHWAVE|FW')

AND NOT REGEXP\_LIKE (EQP\_NAME, '([[:alnum:]]{1,4}\s+)?[[:alnum:]]{8}MS\d-(R|N)FW\d{5}')

CONNECT BY PRIOR EI.EQP\_REFERENCE\_ID = EI.PARENT\_EQP\_REFERENCE\_ID;

COMMIT;

-- since we dont know top-equip from where to start looking for all children equip

-- in the previous start-with-connect-by the same equip might come is as both NE and SHELF.

-- if it does then delete those rows where it is a NE.

DELETE NG\_ADM\_FW\_EQUIP XFE

WHERE XFE.EQ\_CLASS\_TYPE = 'NE'

AND EXISTS (

WITH

TO\_BE\_DELETED AS

(

SELECT EQP\_REFERENCE\_ID

FROM NG\_ADM\_FW\_EQUIP

WHERE EQ\_CLASS\_TYPE = 'SHELF'

INTERSECT

SELECT EQP\_REFERENCE\_ID

FROM NG\_ADM\_FW\_EQUIP

WHERE EQ\_CLASS\_TYPE = 'NE'

)

SELECT 1

FROM TO\_BE\_DELETED TBD

WHERE TBD.EQP\_REFERENCE\_ID = XFE.EQP\_REFERENCE\_ID

);

COMMIT;

-- delete dups if some EQP\_REFERENCE\_ID comes in > 1 as SHELF

/\*select EQP\_REFERENCE\_ID

from NG\_fw\_equip

where EQ\_CLASS\_TYPE = ''SHELF''

group by EQP\_REFERENCE\_ID

having count(1) > 1

;

commit;

\*/

-- update the NG NE rows that dont match Nautilus naming format

UPDATE NG\_ADM\_FW\_EQUIP SET PARSE\_STATUS = 'Name not Nautilus Stds compliant'

WHERE NOT REGEXP\_LIKE(EQP\_NAME, '^([[:alnum:]]{1,4}\s+)?[[:alnum:]]{8}OS\d-(R|N)FW\d{5}(\s\*\[[^][]+\])?$') -- Nautilus std name

AND EQ\_CLASS\_TYPE = 'NE';

COMMIT;

END GETADMNAUTILUSEQUIP;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure AuditFWEquip \*

\* \*

\* Purpose: Audit Fujitsu FlashWave MSPP Equipment against Nautilus Inventory \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE AUDITFWEQUIP IS

BEGIN

-- find matches in netsmart and NG (look only at EQ\_CLASS\_TYPE = NE)

--DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP';

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP (LEAF\_DOMAIN\_NAME, TID, TID\_TYPE, LOCATION, CONTACTS, IP\_ADDR,

MATCH\_CODE, MATCH\_STATUS, NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, MATCH8)

SELECT DISTINCT M.LEAF\_DOMAIN\_NAME, NET.TID, NET.TID\_TYPE, NET.LOCATION, NET.CONTACTS, NET.IP\_ADDR

, CASE WHEN XFE.PARSE\_STATUS IS NULL THEN 'BOTH' ELSE 'NE Only' END MATCH\_CODE

, XFE.PARSE\_STATUS

, XFE.NE\_INST\_ID, XFE.EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, NULL

FROM NETSMART\_SNCNE NET

JOIN NG\_FW\_EQUIP XFE

ON XFE.EQP\_NAME LIKE '%'||NET.TID ||'%' -- need to make these match

LEFT OUTER JOIN CLLI\_DOMAIN\_MAP\_V M

ON SUBSTR(NET.TID,1,6) = SUBSTR(M.CLLI,1,6)

WHERE XFE.EQ\_CLASS\_TYPE = 'NE'

AND REGEXP\_LIKE(NET.TID,'^[[:alnum:]]{8}MS');

COMMIT;

-- if some TID matched > 1 Nautilus ne\_inst\_id then mark it dup

UPDATE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD SET MATCH\_CODE = 'NE Only'

, MATCH\_STATUS = NVL2(MATCH\_STATUS, 'Duplicate in Nautilus, '||RTRIM(MATCH\_STATUS), 'Duplicate in Nautilus')

WHERE EXISTS (SELECT 1

FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP I

WHERE AUD.TID = I.TID

GROUP BY I.TID

HAVING COUNT(1) > 1

);

COMMIT;

-- insert NE Only

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP (TID, TID\_TYPE, LOCATION, CONTACTS, IP\_ADDR, MATCH\_CODE, MATCH\_STATUS)

SELECT TID, TID\_TYPE, LOCATION, CONTACTS, IP\_ADDR, 'NE Only', 'No match in Nautilus'

FROM (

SELECT NET.TID, NET.TID\_TYPE, NET.LOCATION, NET.CONTACTS, IP\_ADDR

FROM NETSMART\_SNCNE NET

WHERE REGEXP\_LIKE(NET.TID,'^[[:alnum:]]{8}MS')

MINUS

SELECT AUD.TID, AUD.TID\_TYPE, AUD.LOCATION, AUD.CONTACTS, IP\_ADDR

FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD

);

COMMIT;

-- check the TID matches MTCE std, else update match\_code = ''NE Only'' and match\_status = ''TID not MTCE Stds compliant''

UPDATE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS = NVL2(MATCH\_STATUS, 'TID not MTCE Stds compliant, '||RTRIM(MATCH\_STATUS), 'TID not MTCE Stds compliant')

WHERE NOT REGEXP\_LIKE(AUD.TID, '[[:alnum:]]{8}MS[[:digit:]]-(R|N)FW[[:digit:]]{5}');

COMMIT;

-- check if TID''s don't match first 6-char match some MTSO CLLI then match\_code = ''NE Only'' and match\_status = ''First six chars of Target Id do not match any MTSO CLLIs''

UPDATE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS = NVL2(MATCH\_STATUS, RTRIM(MATCH\_STATUS)||', First six chars of Target Id do not match any MTSO CLLIs', 'First six chars of Target Id do not match any MTSO CLLIs')

WHERE NOT EXISTS (

WITH

CLLI\_DOMAINS AS

(

SELECT DISTINCT SUBSTR(CLLI, 1,6) CLLI\_6

FROM CLLI\_DOMAIN\_MAP\_V

WHERE UPPER(SUB\_MARKET) != 'NNO'

)

SELECT 1

FROM CLLI\_DOMAINS C

WHERE C.CLLI\_6 = SUBSTR(AUD.TID, 1,6)

);

COMMIT;

-- check if TID''s match first 8-char match some MTSO CLLI then set match\_code = ''BOTH'' temporarily

UPDATE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD SET AUD.MATCH8 = '1'

WHERE EXISTS (

WITH

CLLI\_DOMAINS AS

(

SELECT DISTINCT CLLI, SUBSTR(CLLI, 1, 6) CLLI\_6

FROM CLLI\_DOMAIN\_MAP\_V

WHERE UPPER(SUB\_MARKET) != 'NNO'

)

SELECT 1

FROM CLLI\_DOMAINS C

WHERE C.CLLI = SUBSTR(AUD.TID, 1,8)

);

COMMIT;

-- check if TID''s match first 6-char match some MTSO CLLI then match\_code = ''NE Only'' and match\_status = ''Only 1st 6-char of CLLI match MTSO''

UPDATE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS = NVL2(MATCH\_STATUS, RTRIM(MATCH\_STATUS)||', Only first six chars of Target Id match a MTSO CLLI', 'Only first six chars of Target Id match a MTSO CLLI')

WHERE EXISTS (

WITH

CLLI\_DOMAINS AS

(

SELECT DISTINCT CLLI, SUBSTR(CLLI, 1, 6) CLLI\_6

FROM CLLI\_DOMAIN\_MAP\_V

WHERE UPPER(SUB\_MARKET) != 'NNO'

)

SELECT 1

FROM CLLI\_DOMAINS C

WHERE C.CLLI\_6 = SUBSTR(AUD.TID, 1,6) AND AUD.MATCH8 IS NULL

);

COMMIT;

-- Set Domain name for NNO Equipment

UPDATE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD SET LEAF\_DOMAIN\_NAME = 'NNO\_DOMAIN'

WHERE REGEXP\_LIKE(AUD.TID, '[[:alnum:]]{8}MS[[:digit:]]-NFW[[:digit:]]{5}');

-- insert NG Only. These will only be the top-level equip NE\_inst\_id ONLY

-- these will be shown in details, but wont affect the summary

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP (NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, MATCH\_CODE, MATCH\_STATUS)

SELECT NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, 'Nautilus Only', 'No match in NE'

FROM (

SELECT NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR

FROM NG\_FW\_EQUIP XFE

MINUS

SELECT NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR

FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD

);

COMMIT;

UPDATE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP AUD SET MATCH\_STATUS = RTRIM(MATCH\_STATUS);

COMMIT;

--DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_NE\_VS\_NG\_AUDIT';

-- Move records into WK table where we already know the domain name.

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT

SELECT \* FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP WHERE LEAF\_DOMAIN\_NAME IS NOT NULL;

DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP WHERE LEAF\_DOMAIN\_NAME IS NOT NULL;

COMMIT;

-- Move NG\_Only records with Leaf Domain name.

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID)

WHERE

NE.TID IS NULL;

DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP WHERE NE\_INST\_ID IN (SELECT DISTINCT

NE.NE\_INST\_ID

FROM

NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID)

WHERE

NE.TID IS NULL);

COMMIT;

-- Insert NE\_Only records with Leaf Domain name.

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID)

;

DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP WHERE NE\_INST\_ID IN (SELECT DISTINCT

NE.NE\_INST\_ID

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID));

COMMIT;

-- The only items left should be NE Only Items

-- Insert NE\_Only records with only a match in the CLLI\_DOMAIN\_MAP (8 Character).

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.CLLI = SUBSTR(NE.TID,1,8));

DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP WHERE TID IN (SELECT DISTINCT

NE.TID

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.CLLI = SUBSTR(NE.TID,1,8)));

COMMIT;

-- Insert NE\_Only records with only a match in the CLLI\_DOMAIN\_MAP (6 Character).

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (SUBSTR(CDM.CLLI,1,6) = SUBSTR(NE.TID,1,6));

DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP WHERE TID IN (SELECT DISTINCT

NE.TID

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (SUBSTR(CDM.CLLI,1,6) = SUBSTR(NE.TID,1,6)));

COMMIT;

-- The only records left are those TIDs that do not match the CLLI DOMAIN MAP and have no match in Nautilus.

INSERT INTO NETSMART\_NE\_VS\_NG\_AUDIT

SELECT NE.\* FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP NE;

--DELETE FROM NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP';

COMMIT;

-- dont implement yet. even though Shirley says it is a good idea, she needs approvals from other area leads.

-- EQP\_MODEL # in TID\_TYPE match EQP\_MODEL# in NG\_equip\_EQP\_TYPE

-- EQP\_TYPE of the NE\_inst\_id should be MSPP or FIBER MUX for 9500 and FIBRE MUX for all others

END AUDITFWEQUIP;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure AuditADMFWEquip \*

\* \*

\* Purpose: Audit Fujitsu FlashWave ADM Equipment against Nautilus Inventory \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE AUDITADMFWEQUIP IS

BEGIN

-- find matches in netsmart and NG (look only at EQ\_CLASS\_TYPE = NE)

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP';

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP (LEAF\_DOMAIN\_NAME, TID, TID\_TYPE, LOCATION, CONTACTS, IP\_ADDR,

MATCH\_CODE, MATCH\_STATUS, NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, MATCH8)

SELECT DISTINCT M.LEAF\_DOMAIN\_NAME, NET.TID, NET.TID\_TYPE, NET.LOCATION, NET.CONTACTS, NET.IP\_ADDR

, CASE WHEN XFE.PARSE\_STATUS IS NULL THEN 'BOTH' ELSE 'NE Only' END MATCH\_CODE

, XFE.PARSE\_STATUS

, XFE.NE\_INST\_ID, XFE.EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, NULL

FROM NETSMART\_SNCNE NET

JOIN NG\_ADM\_FW\_EQUIP XFE

ON XFE.EQP\_NAME LIKE '%'||NET.TID ||'%' -- need to make these match

LEFT OUTER JOIN CLLI\_DOMAIN\_MAP\_V M

ON SUBSTR(NET.TID,1,6) = SUBSTR(M.CLLI,1,6)

WHERE XFE.EQ\_CLASS\_TYPE = 'NE'

AND NOT REGEXP\_LIKE(NET.TID,'^[[:alnum:]]{8}MS');

COMMIT;

-- if some TID matched > 1 Nautilus ne\_inst\_id then mark it dup

UPDATE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD SET MATCH\_CODE = 'NE Only'

, MATCH\_STATUS = NVL2(MATCH\_STATUS, 'Duplicate in Nautilus, '||RTRIM(MATCH\_STATUS), 'Duplicate in Nautilus')

WHERE EXISTS (SELECT 1

FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP I

WHERE AUD.TID = I.TID

GROUP BY I.TID

HAVING COUNT(1) > 1

);

COMMIT;

-- insert NE Only

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP (TID, TID\_TYPE, LOCATION, CONTACTS, IP\_ADDR, MATCH\_CODE, MATCH\_STATUS)

SELECT TID, TID\_TYPE, LOCATION, CONTACTS, IP\_ADDR, 'NE Only', 'No match in Nautilus'

FROM (

SELECT NET.TID, NET.TID\_TYPE, NET.LOCATION, NET.CONTACTS, IP\_ADDR

FROM NETSMART\_SNCNE NET

WHERE NOT REGEXP\_LIKE(NET.TID,'^[[:alnum:]]{8}MS')

MINUS

SELECT AUD.TID, AUD.TID\_TYPE, AUD.LOCATION, AUD.CONTACTS, IP\_ADDR

FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD

);

COMMIT;

-- check the TID matches MTCE std, else update match\_code = ''NE Only'' and match\_status = ''TID not MTCE Stds compliant''

UPDATE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS = NVL2(MATCH\_STATUS, 'TID not MTCE Stds compliant, '||RTRIM(MATCH\_STATUS), 'TID not MTCE Stds compliant')

WHERE NOT REGEXP\_LIKE(AUD.TID, '[[:alnum:]]{8}OS[[:digit:]]-(R|N)FW[[:digit:]]{5}');

COMMIT;

-- check if TID''s don't match first 6-char match some MTSO CLLI then match\_code = ''NE Only'' and match\_status = ''First six chars of Target Id do not match any MTSO CLLIs''

UPDATE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS = NVL2(MATCH\_STATUS, RTRIM(MATCH\_STATUS)||', First six chars of Target Id do not match any MTSO CLLIs', 'First six chars of Target Id do not match any MTSO CLLIs')

WHERE NOT EXISTS (

WITH

CLLI\_DOMAINS AS

(

SELECT DISTINCT SUBSTR(CLLI, 1,6) CLLI\_6

FROM CLLI\_DOMAIN\_MAP\_V

WHERE UPPER(SUB\_MARKET) != 'NNO'

)

SELECT 1

FROM CLLI\_DOMAINS C

WHERE C.CLLI\_6 = SUBSTR(AUD.TID, 1,6)

);

COMMIT;

-- check if TID''s match first 8-char match some MTSO CLLI then set match\_code = ''BOTH'' temporarily

UPDATE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD SET AUD.MATCH8 = '1'

WHERE EXISTS (

WITH

CLLI\_DOMAINS AS

(

SELECT DISTINCT CLLI, SUBSTR(CLLI, 1, 6) CLLI\_6

FROM CLLI\_DOMAIN\_MAP\_V

WHERE UPPER(SUB\_MARKET) != 'NNO'

)

SELECT 1

FROM CLLI\_DOMAINS C

WHERE C.CLLI = SUBSTR(AUD.TID, 1,8)

);

COMMIT;

-- check if TID''s match first 6-char match some MTSO CLLI then match\_code = ''NE Only'' and match\_status = ''Only 1st 6-char of CLLI match MTSO''

UPDATE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS = NVL2(MATCH\_STATUS, RTRIM(MATCH\_STATUS)||', Only first six chars of Target Id match a MTSO CLLI', 'Only first six chars of Target Id match a MTSO CLLI')

WHERE EXISTS (

WITH

CLLI\_DOMAINS AS

(

SELECT DISTINCT CLLI, SUBSTR(CLLI, 1, 6) CLLI\_6

FROM CLLI\_DOMAIN\_MAP\_V

WHERE UPPER(SUB\_MARKET) != 'NNO'

)

SELECT 1

FROM CLLI\_DOMAINS C

WHERE C.CLLI\_6 = SUBSTR(AUD.TID, 1,6) AND AUD.MATCH8 IS NULL

);

COMMIT;

-- Set Domain name for NNO Equipment

UPDATE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD SET LEAF\_DOMAIN\_NAME = 'NNO\_DOMAIN'

WHERE REGEXP\_LIKE(AUD.TID, '[[:alnum:]]{8}OS[[:digit:]]-NFW[[:digit:]]{5}');

-- insert NG Only. These will only be the top-level equip NE\_inst\_id ONLY

-- these will be shown in details, but wont affect the summary

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP (NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, MATCH\_CODE, MATCH\_STATUS)

SELECT NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR, 'Nautilus Only', 'No match in NE'

FROM (

SELECT NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR

FROM NG\_ADM\_FW\_EQUIP XFE

MINUS

SELECT NE\_INST\_ID, EQP\_NAME, EQP\_MODEL, EQP\_VENDOR

FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD

);

COMMIT;

UPDATE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP AUD SET MATCH\_STATUS = RTRIM(MATCH\_STATUS);

COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_NE\_VS\_NG\_AUDIT';

-- Move records into WK table where we already know the domain name.

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

SELECT \* FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP WHERE LEAF\_DOMAIN\_NAME IS NOT NULL;

DELETE FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP WHERE LEAF\_DOMAIN\_NAME IS NOT NULL;

COMMIT;

-- Move NG\_Only records with Leaf Domain name.

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID)

WHERE

NE.TID IS NULL;

DELETE FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP WHERE NE\_INST\_ID IN (SELECT DISTINCT

NE.NE\_INST\_ID

FROM

NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID)

WHERE

NE.TID IS NULL);

COMMIT;

-- Insert NE\_Only records with Leaf Domain name.

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID)

;

DELETE FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP WHERE NE\_INST\_ID IN (SELECT DISTINCT

NE.NE\_INST\_ID

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

JOIN

NG\_REPORTS.EQUIPMENT E ON E.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID

JOIN

NG\_REPORTS.EQUIPMENT\_DOMAIN\_MAP EDM ON EDM.EQP\_REFERENCE\_ID = E.EQP\_REFERENCE\_ID

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.LEAF\_DOMAIN\_ID = EDM.DOMAIN\_ID));

COMMIT;

-- Insert NE\_Only records with only a match in the CLLI\_DOMAIN\_MAP (8 Character).

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.CLLI = SUBSTR(NE.TID,1,8));

DELETE FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP WHERE TID IN (SELECT DISTINCT

NE.TID

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (CDM.CLLI = SUBSTR(NE.TID,1,8)));

COMMIT;

-- Insert NE\_Only records with only a match in the CLLI\_DOMAIN\_MAP (6 Character).

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

SELECT DISTINCT

CDM.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

NE.EQP\_NAME,

NE.EQP\_MODEL,

NE.EQP\_VENDOR,

NE.MATCH8

FROM

NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (SUBSTR(CDM.CLLI,1,6) = SUBSTR(NE.TID,1,6));

DELETE FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP WHERE TID IN (SELECT DISTINCT

TID

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

JOIN

CLLI\_DOMAIN\_MAP\_V CDM ON (SUBSTR(CDM.CLLI,1,6) = SUBSTR(NE.TID,1,6)));

COMMIT;

-- The only records left are those TIDs that do not match the CLLI DOMAIN MAP and have no match in Nautilus.

INSERT INTO NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

SELECT NE.\* FROM NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP NE;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP';

COMMIT;

-- dont implement yet. even though Shirley says it is a good idea, she needs approvals from other area leads.

-- EQP\_MODEL # in TID\_TYPE match EQP\_MODEL# in NG\_equip\_EQP\_TYPE

-- EQP\_TYPE of the NE\_inst\_id should be MSPP or FIBER MUX for 9500 and FIBRE MUX for all others

END AUDITADMFWEQUIP;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure GenSummary \*

\* \*

\* Purpose: Realize summary report \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE GENSUMMARY IS

BEGIN

--DELETE FROM NETSMART\_EQ\_SUMM\_REGION\_WK;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_EQ\_SUMM\_TMSM';

-- generate summary

INSERT INTO NETSMART\_EQ\_SUMM\_TMSM (TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, TOTAL\_NE, DUPLICATE\_NE, MATCHED\_NE, PERCENT\_MATCHED, PASSED\_AUDIT, NAMING\_COMPLIANCE, DISCOVERED\_XCONNECTS, LIVE\_XCONNECTS, OTHER\_XCONNECTS, TOTAL\_COMPLIANCE)

WITH DEVICEMETRICS AS

(

SELECT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

(CASE (COUNT(TID) - 1) WHEN 0 THEN 0 ELSE 1 END) HAS\_DUPLICATES,

MAX(DECODE(NE\_INST\_ID, NULL, 0, 1)) NE\_MATCHED,

MAX(DECODE(UPPER(TRIM(MATCH\_CODE)), 'BOTH', 1, 0)) PASSED\_AUDIT,

MAX(NUM\_XCONNECT) NUM\_XCONNECT,

MAX(LIVE\_XCONNECTS) LIVE\_XCONNECTS,

MAX(OTHER\_XCONNECTS) OTHER\_XCONNECTS

FROM

TABLE(NG\_REPORTS.NETSMART\_AUDIT.NS\_DETAIL\_TBL\_WRK(1, 10000))

WHERE

TID IS NOT NULL

GROUP BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID

)

,

DOMAINS AS

(

SELECT TERRITORY, MARKET\_TERRITORY, SUB\_MARKET FROM DOMAINS\_REGIONAL\_REPORTING DRR WHERE DRR.SUB\_MARKET <> 'OSS'

UNION

SELECT 'unknown' TERRITORY, 'unknown' MARKET\_TERRITORY, 'unknown' SUB\_MARKET FROM DUAL

)

SELECT

DRR.TERRITORY

, DRR.MARKET\_TERRITORY

, DRR.SUB\_MARKET

, NVL(COUNT(DM.TID),0) TOTAL\_NE

, NVL(SUM(DM.HAS\_DUPLICATES), 0) DUPLICATE\_NE

, NVL(SUM(DM.NE\_MATCHED), 0) MATCHED\_NE

, (CASE SUM(DECODE(NVL(DM.NE\_MATCHED, 0),0,0,1)) WHEN 0 THEN ' 0.00' ELSE (TO\_CHAR(ROUND((SUM(DECODE(NVL(DM.NE\_MATCHED, 0),0,0,1)))/COUNT(DM.TID)\*100,2),'990.00')) END) PERCENT\_MATCHED

, NVL(SUM(DM.PASSED\_AUDIT), 0) PASSED\_AUDIT

, (CASE NVL(COUNT(DM.TID), 0) WHEN 0 THEN ' 0.00' ELSE (TO\_CHAR(ROUND((SUM(DM.PASSED\_AUDIT)/COUNT(DM.TID)\*100),2),'990.00')) END) NAMING\_COMPLIANCE

, NVL(SUM(DM.NUM\_XCONNECT), 0) DISCOVERED\_XCONNECTS

, NVL(SUM(DM.LIVE\_XCONNECTS), 0) LIVE\_XCONNECTS

, NVL(SUM(DM.OTHER\_XCONNECTS), 0) OTHER\_XCONNECTS

, CASE

WHEN NVL(SUM(DM.NUM\_XCONNECT), 0) > 0

THEN

--TO\_CHAR(ROUND(((NVL(SUM(DM.PASSED\_AUDIT), 0) + NVL(SUM(DM.LIVE\_XCONNECTS), 0) + NVL(SUM(DM.OTHER\_XCONNECTS), 0))/(NVL(COUNT(DM.TID),0) + NVL(SUM(DM.NUM\_XCONNECT), 0)))\*100,2),'990.00')

TO\_CHAR(ROUND(((NVL(SUM(DM.LIVE\_XCONNECTS), 0) + NVL(SUM(DM.OTHER\_XCONNECTS), 0))/(NVL(SUM(DM.NUM\_XCONNECT), 0)))\*100,2),'990.00')

ELSE

' 0.00'

END TOTAL\_COMPLIANCE

FROM

DOMAINS DRR

LEFT OUTER JOIN

DEVICEMETRICS DM ON DM.SUB\_MARKET = DRR.SUB\_MARKET

GROUP BY

ROLLUP(DRR.TERRITORY, DRR.MARKET\_TERRITORY, DRR.SUB\_MARKET)

ORDER BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET;

COMMIT;

END GENSUMMARY;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure GenADMSummary \*

\* \*

\* Purpose: Realize ADM Summary report \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE GENADMSUMMARY IS

BEGIN

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_EQ\_SUMM\_TMSM';

-- generate summary

INSERT INTO NETSMART\_ADM\_EQ\_SUMM\_TMSM (TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, TOTAL\_NE, DUPLICATE\_NE, MATCHED\_NE, PERCENT\_MATCHED, PASSED\_AUDIT, NAMING\_COMPLIANCE, DISCOVERED\_XCONNECTS, LIVE\_XCONNECTS, OTHER\_XCONNECTS, TOTAL\_COMPLIANCE)

WITH DEVICEMETRICS AS

(

SELECT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

(CASE (COUNT(TID) - 1) WHEN 0 THEN 0 ELSE 1 END) HAS\_DUPLICATES,

MAX(DECODE(NE\_INST\_ID, NULL, 0, 1)) NE\_MATCHED,

MAX(DECODE(UPPER(TRIM(MATCH\_CODE)), 'BOTH', 1, 0)) PASSED\_AUDIT,

MAX(NUM\_XCONNECT) NUM\_XCONNECT,

MAX(LIVE\_XCONNECTS) LIVE\_XCONNECTS,

MAX(OTHER\_XCONNECTS) OTHER\_XCONNECTS

FROM

TABLE(NG\_REPORTS.NETSMART\_AUDIT.NS\_ADM\_DETAIL\_TBL\_WRK(1, 10000))

WHERE

TID IS NOT NULL

GROUP BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID

)

,

DOMAINS AS

(

SELECT DISTINCT TERRITORY, MARKET\_TERRITORY, SUB\_MARKET FROM NG\_REPORTS.CLLI\_DOMAIN\_MAP\_V DRR

UNION

SELECT 'unknown' TERRITORY, 'unknown' MARKET\_TERRITORY, 'unknown' SUB\_MARKET FROM DUAL

)

SELECT

DRR.TERRITORY

, DRR.MARKET\_TERRITORY

, DRR.SUB\_MARKET

, NVL(COUNT(DM.TID),0) TOTAL\_NE

, NVL(SUM(DM.HAS\_DUPLICATES), 0) DUPLICATE\_NE

, NVL(SUM(DM.NE\_MATCHED), 0) MATCHED\_NE

, (CASE SUM(DECODE(NVL(DM.NE\_MATCHED, 0),0,0,1)) WHEN 0 THEN ' 0.00' ELSE (TO\_CHAR(ROUND((SUM(DECODE(NVL(DM.NE\_MATCHED, 0),0,0,1)))/COUNT(DM.TID)\*100,2),'990.00')) END) PERCENT\_MATCHED

, NVL(SUM(DM.PASSED\_AUDIT), 0) PASSED\_AUDIT

, (CASE NVL(COUNT(DM.TID), 0) WHEN 0 THEN ' 0.00' ELSE (TO\_CHAR(ROUND((SUM(DM.PASSED\_AUDIT)/COUNT(DM.TID)\*100),2),'990.00')) END) NAMING\_COMPLIANCE

, NVL(SUM(DM.NUM\_XCONNECT), 0) DISCOVERED\_XCONNECTS

, NVL(SUM(DM.LIVE\_XCONNECTS), 0) LIVE\_XCONNECTS

, NVL(SUM(DM.OTHER\_XCONNECTS), 0) OTHER\_XCONNECTS

, CASE

WHEN NVL(SUM(DM.NUM\_XCONNECT), 0) > 0

THEN

--TO\_CHAR(ROUND(((NVL(SUM(DM.PASSED\_AUDIT), 0) + NVL(SUM(DM.LIVE\_XCONNECTS), 0) + NVL(SUM(DM.OTHER\_XCONNECTS), 0))/(NVL(COUNT(DM.TID),0) + NVL(SUM(DM.NUM\_XCONNECT), 0)))\*100,2),'990.00')

TO\_CHAR(ROUND(((NVL(SUM(DM.LIVE\_XCONNECTS), 0) + NVL(SUM(DM.OTHER\_XCONNECTS), 0))/(NVL(SUM(DM.NUM\_XCONNECT), 0)))\*100,2),'990.00')

ELSE

' 0.00'

END TOTAL\_COMPLIANCE

FROM

DOMAINS DRR

LEFT OUTER JOIN

DEVICEMETRICS DM ON DM.SUB\_MARKET = DRR.SUB\_MARKET

GROUP BY

ROLLUP(DRR.TERRITORY, DRR.MARKET\_TERRITORY, DRR.SUB\_MARKET)

ORDER BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET;

COMMIT;

END GENADMSUMMARY;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure Audit\_XConnects \*

\* \*

\* Purpose: Audit FlashWave cross connects against NG \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE AUDIT\_XCONNECTS IS

BEGIN

--DELETE FROM NETSMART\_FUJITSU\_EQUIPMENT;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_FUJITSU\_EQUIPMENT';

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_XCONNECT\_EQUIPMENT';

INSERT INTO NETSMART\_XCONNECT\_EQUIPMENT

SELECT

LEVEL NE\_LEVEL,

TO\_NUMBER

(TRIM

(BOTH '/' FROM SYS\_CONNECT\_BY\_PATH

(DECODE (LEVEL,

1, EI.EQP\_REFERENCE\_ID,

''

),

'/'

)

)

) NE\_INST\_ID,

EI.EQP\_REFERENCE\_ID,

EI.SITE\_REFERENCE\_ID,

EI.PARENT\_EQP\_REFERENCE\_ID,

DECODE (LEVEL,

1, 'NE',

DECODE (EI.CONTAINER, 'SHELF', 'SHELF', 'N/A') --------------CONTAINER---???

) EQ\_CLASS\_TYPE,

EI.EQP\_NAME ,

EI.EQP\_MODEL ,

EI.EQP\_TYPE,

EI.INV\_STATUS,

EI.EQP\_VENDOR,

EI.CONTAINER

FROM NG\_REPORTS.EQUIPMENT EI

WHERE

(LEVEL = 1 OR EI.CONTAINER= 'SHELF')

START WITH

EI.EQP\_VENDOR = 'FUJITSU'

AND REGEXP\_LIKE (EQP\_MODEL, 'FLASHWAVE|FW')

CONNECT BY

PRIOR EI.EQP\_REFERENCE\_ID = EI.PARENT\_EQP\_REFERENCE\_ID;

COMMIT;

INSERT INTO NETSMART\_FUJITSU\_EQUIPMENT

SELECT

NE.TID,

NE.TID\_TYPE,

NXE.NE\_LEVEL,

NXE.NE\_INST\_ID,

NXE.EQP\_REFERENCE\_ID,

NXE.SITE\_REFERENCE\_ID,

NXE.PARENT\_EQP\_REFERENCE\_ID,

NXE.EQ\_CLASS\_TYPE,

NXE.EQP\_NAME,

NXE.EQP\_MODEL,

NXE.EQP\_TYPE,

NXE.INV\_STATUS,

NXE.EQP\_VENDOR,

NXE.CONTAINER

FROM

NETSMART\_XCONNECT\_EQUIPMENT NXE

JOIN

NETSMART\_NE\_VS\_NG\_AUDIT NE ON (NE.NE\_INST\_ID = NXE.NE\_INST\_ID)

WHERE

NE.TID IS NOT NULL;

COMMIT;

UPDATE NETSMART\_FUJITSU\_EQUIPMENT

SET PARENT\_EQP\_REFERENCE\_ID = NULL

WHERE EQP\_REFERENCE\_ID IN

(SELECT

NE2.EQP\_REFERENCE\_ID

FROM

NETSMART\_FUJITSU\_EQUIPMENT NE2

WHERE

NE2.PARENT\_EQP\_REFERENCE\_ID IS NOT NULL

AND NOT EXISTS (SELECT

NE.EQP\_REFERENCE\_ID

FROM

NETSMART\_FUJITSU\_EQUIPMENT NE

WHERE

NE.EQP\_REFERENCE\_ID = NE2.PARENT\_EQP\_REFERENCE\_ID

)

);

COMMIT;

--DELETE FROM NETSMART\_NG\_PORTS;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_NG\_PORTS';

INSERT INTO NETSMART\_NG\_PORTS

with legs as (select TRAIL\_ID, COMPONENT\_ID, NAME, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%WORKING%' and upper(NAME) not like '%PROTECT%'

--and TRAIL\_ID = 3369828

union

select TRAIL\_ID, COMPONENT\_ID, NAME, 2 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

union

select a.TRAIL\_ID, a.COMPONENT\_ID, a.NAME, 3 as match\_ord

from (select TRAIL\_ID, COMPONENT\_ID, NAME, SEQUENCE, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

) a,

(select distinct TRAIL\_ID, min(SEQUENCE) min\_ro

from NG\_TOPOLOGY.TRAIL\_COMPONENT LI

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

group by TRAIL\_ID

) b

where a.TRAIL\_ID = b.TRAIL\_ID

and a.SEQUENCE = b.min\_ro),

max\_leg as (select TRAIL\_ID, min(match\_ord) as match\_ord

from legs

group by TRAIL\_ID),

legs\_unq as (select l.TRAIL\_ID, l.COMPONENT\_ID, l.NAME

from legs l, max\_leg ml

where l.TRAIL\_ID = ml.TRAIL\_ID

and l.match\_ord = ml.match\_ord)

SELECT DISTINCT

NE.TID,

NE.TID\_TYPE,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(CRD.SLOT\_NAME, '-0+', '-'), '^0+', ''), '^-', '0-') SLOT\_NAME,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(EP.PORT\_NAME, ':', '-'), '-0+', '-'), '^0+', '') PORT\_NAME,

EP.AID,

CPE.TRAIL\_ID,

LI.COMPONENT\_ID,

CPE.CABLE\_MEMBER,

CPI.A\_SITE\_ID,

S1.SITE\_NAME A\_SITE\_NAME,

CPI.Z\_SITE\_ID,

S2.SITE\_NAME Z\_SITE\_NAME,

CPI.TRAIL\_NAME,

CPI.BANDWIDTH,

CPI.STATUS PATH\_STATUS,

CPI.NEXT\_REVISION\_TRAIL\_ID,

CPI.PREVIOUS\_REVISION\_TRAIL\_ID,

NE.NE\_INST\_ID,

NE.EQP\_REFERENCE\_ID,

NE.PARENT\_EQP\_REFERENCE\_ID,

NE.SITE\_REFERENCE\_ID,

NE.EQP\_VENDOR,

NE.EQP\_MODEL,

NE.EQP\_NAME,

EP.PORT\_REFERENCE\_ID,

EP.CARD\_REFERENCE\_ID

FROM

NETSMART\_FUJITSU\_EQUIPMENT NE

JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.EQP\_REFERENCE\_ID)

JOIN

NG\_REPORTS.PORT EP ON (EP.EQP\_REFERENCE\_ID = EQ.EQP\_REFERENCE\_ID)

JOIN

NG\_TOPOLOGY.TRAIL\_ELEMENT CPE ON CPE.ELEMENT\_REF\_ID = EP.PORT\_REFERENCE\_ID and CPE.ELEMENT\_TYPE='E'

JOIN

legs\_unq LI ON LI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_TOPOLOGY.TRAIL\_COMPONENT\_ELEMENT PLM ON PLM.COMPONENT\_ID = LI.COMPONENT\_ID AND PLM.ELEMENT\_ID = CPE.ELEMENT\_ID

JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_REPORTS.SITE S1 ON S1.SITE\_REFERENCE\_ID = CPI.A\_SITE\_ID

JOIN

NG\_REPORTS.SITE S2 ON S2.SITE\_REFERENCE\_ID = CPI.Z\_SITE\_ID

JOIN

NG\_REPORTS.CARD CRD ON (CRD.CARD\_REFERENCE\_ID = EP.CARD\_REFERENCE\_ID)

WHERE

NE.TID IS NOT NULL

AND REGEXP\_LIKE(EP.PORT\_NAME, '-')

AND UPPER(CPI.STATUS) = 'LIVE'

AND NE.CONTAINER= 'SHELF'

ORDER BY

TID,

PORT\_NAME;

COMMIT;

---------------------------------------------------------------

-- Insert ports where a "live" port does not already exist --

---------------------------------------------------------------

INSERT INTO NETSMART\_NG\_PORTS

with legs as (select TRAIL\_ID, COMPONENT\_ID, NAME, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%WORKING%' and upper(NAME) not like '%PROTECT%'

--and TRAIL\_ID = 3369828

union

select TRAIL\_ID, COMPONENT\_ID, NAME, 2 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

union

select a.TRAIL\_ID, a.COMPONENT\_ID, a.NAME, 3 as match\_ord

from (select TRAIL\_ID, COMPONENT\_ID, NAME, SEQUENCE, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

) a,

(select distinct TRAIL\_ID, min(SEQUENCE) min\_ro

from NG\_TOPOLOGY.TRAIL\_COMPONENT LI

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

group by TRAIL\_ID

) b

where a.TRAIL\_ID = b.TRAIL\_ID

and a.SEQUENCE = b.min\_ro),

max\_leg as (select TRAIL\_ID, min(match\_ord) as match\_ord

from legs

group by TRAIL\_ID),

legs\_unq as (select l.TRAIL\_ID, l.COMPONENT\_ID, l.NAME

from legs l, max\_leg ml

where l.TRAIL\_ID = ml.TRAIL\_ID

and l.match\_ord = ml.match\_ord)

SELECT DISTINCT

NE.TID,

NE.TID\_TYPE,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(CRD.SLOT\_NAME, '-0+', '-'), '^0+', ''), '^-', '0-') SLOT\_NAME,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(EP.PORT\_NAME, ':', '-'), '-0+', '-'), '^0+', '') PORT\_NAME,

EP.AID,

CPE.TRAIL\_ID,

LI.COMPONENT\_ID,

CPE.CABLE\_MEMBER,

CPI.A\_SITE\_ID,

S1.SITE\_NAME A\_SITE\_NAME,

CPI.Z\_SITE\_ID,

S2.SITE\_NAME Z\_SITE\_NAME,

CPI.TRAIL\_NAME,

CPI.BANDWIDTH,

CPI.STATUS PATH\_STATUS,

CPI.NEXT\_REVISION\_TRAIL\_ID,

CPI.PREVIOUS\_REVISION\_TRAIL\_ID,

NE.NE\_INST\_ID,

NE.EQP\_REFERENCE\_ID,

NE.PARENT\_EQP\_REFERENCE\_ID,

NE.SITE\_REFERENCE\_ID,

NE.EQP\_VENDOR,

NE.EQP\_MODEL,

NE.EQP\_NAME,

EP.PORT\_REFERENCE\_ID,

EP.CARD\_REFERENCE\_ID

FROM

NETSMART\_FUJITSU\_EQUIPMENT NE

JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.EQP\_REFERENCE\_ID OR EQ.PARENT\_EQP\_REFERENCE\_ID = NE.EQP\_REFERENCE\_ID)

JOIN

NG\_REPORTS.PORT EP ON (EP.EQP\_REFERENCE\_ID = EQ.EQP\_REFERENCE\_ID)

JOIN

NG\_TOPOLOGY.TRAIL\_ELEMENT CPE ON CPE.ELEMENT\_REF\_ID = EP.PORT\_REFERENCE\_ID and CPE.ELEMENT\_TYPE='E'

JOIN

legs\_unq LI ON LI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_TOPOLOGY.TRAIL\_COMPONENT\_ELEMENT PLM ON PLM.COMPONENT\_ID = LI.COMPONENT\_ID AND PLM.ELEMENT\_ID = CPE.ELEMENT\_ID

JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_REPORTS.SITE S1 ON S1.SITE\_REFERENCE\_ID = CPI.A\_SITE\_ID

JOIN

NG\_REPORTS.SITE S2 ON S2.SITE\_REFERENCE\_ID = CPI.Z\_SITE\_ID

JOIN

NG\_REPORTS.CARD CRD ON (CRD.CARD\_REFERENCE\_ID = EP.CARD\_REFERENCE\_ID)

WHERE

NE.TID IS NOT NULL

AND REGEXP\_LIKE(EP.PORT\_NAME, '-')

and upper(cpi.status) != 'LIVE'

AND NE.CONTAINER= 'SHELF'

AND NOT EXISTS (SELECT 1

FROM

NG\_REPORTS.NETSMART\_NG\_PORTS NXP

WHERE

NXP.TID = NE.TID

--AND NXP.SLOT\_NAME = REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(CRD.SLOT\_NAME, '-0+', '-'), '^0+', ''), '^-', '0-')

AND NXP.PORT\_NAME = REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(EP.PORT\_NAME, ':', '-'), '-0+', '-'), '^0+', '')

)

ORDER BY

TID,

PORT\_NAME;

COMMIT;

--DELETE FROM netsmart\_xconnect\_audit;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE netsmart\_xconnect\_audit';

INSERT INTO netsmart\_xconnect\_audit

SELECT

NE.TID,

NE.TID\_TYPE,

NULL MATCH\_CODE,

NULL MATCH\_STATUS,

XC.XCONNECT\_ASIDE,

XP1.NE\_INST\_ID A\_EQUIPMENT\_ID,

XP1.TRAIL\_ID A\_SIDE\_PATH\_INST\_ID,

XP1.COMPONENT\_ID A\_COMPONENT\_ID,

XP1.CABLE\_MEMBER A\_CABLE\_MEMBER,

XP1.PORT\_NAME A\_PORT\_HUM\_ID,

XP1.AID a\_port\_aid,

XC.XCONNECT\_ZSIDE,

XP2.NE\_INST\_ID Z\_EQUIPMENT\_ID,

XP2.TRAIL\_ID Z\_SIDE\_PATH\_INST\_ID,

XP2.COMPONENT\_ID A\_COMPONENT\_ID,

XP2.CABLE\_MEMBER Z\_CABLE\_MEMBER,

XP2.PORT\_NAME Z\_PORT\_HUM\_ID,

XP2.AID Z\_PORT\_ACCESS\_ID,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.TRAIL\_ID

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.TRAIL\_ID

ELSE NULL

END TRAIL\_ID,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.TRAIL\_NAME

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.TRAIL\_NAME

ELSE NULL

END TRAIL\_NAME,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.PATH\_STATUS

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.PATH\_STATUS

ELSE NULL

END PATH\_STATUS,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.BANDWIDTH

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.BANDWIDTH

ELSE NULL

END BANDWIDTH

FROM

NETSMART\_SNCNE NE

JOIN

NETSMART\_XCONNECT XC ON XC.SNCNE\_ID = NE.INST\_ID

LEFT OUTER JOIN

NETSMART\_NG\_PORTS XP1 ON XP1.TID = NE.TID AND (((XP1.PORT\_NAME = XC.XCONNECT\_ASIDE) OR (XP1.SLOT\_NAME || '-' || XP1.PORT\_NAME = XC.XCONNECT\_ASIDE)) OR

(XP1.AID = XC.XCONNECT\_ASIDE))

LEFT OUTER JOIN

NETSMART\_NG\_PORTS XP2 ON XP2.TID = NE.TID AND (((XP2.PORT\_NAME = XC.XCONNECT\_ZSIDE) OR (XP2.SLOT\_NAME || '-' || XP2.PORT\_NAME = XC.XCONNECT\_ZSIDE)) OR

(XP2.AID = XC.XCONNECT\_ZSIDE))

WHERE

REGEXP\_LIKE(NE.TID,'^[[:alnum:]]{8}MS')

-- WHERE

-- XP1.A\_SITE\_ID = XP2.Z\_SITE\_ID

-- REGEXP\_LIKE(NE.TID\_TYPE, '9500')

ORDER BY

NE.TID,

XCONNECT\_ASIDE,

XCONNECT\_ZSIDE;

COMMIT;

-- Delete the xconnect rows where xconnect a\_side = xconnect z\_side (loopback)

DELETE FROM

NG\_REPORTS.netsmart\_xconnect\_audit

WHERE

XCONNECT\_ASIDE = XCONNECT\_ZSIDE AND

((A\_EQUIPMENT\_ID = Z\_EQUIPMENT\_ID) OR (A\_EQUIPMENT\_ID IS NULL AND Z\_EQUIPMENT\_ID IS NULL));

COMMIT;

-- Mark the audit rows where the there is a duplicate in the port access id values (A Side)

update netsmart\_xconnect\_audit xc set match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', Port Access ID ''' || a\_port\_aid || ''' exists on more than one port in this TID. This can cause false discrepencies.',

'Port Access ID ''' || a\_port\_aid || ''' exists on more than one port in this TID. This can cause false discrepencies.')

where exists (

with port\_counts as

(

select \* from

(

select

tid,

AID,

count(1) as port\_count

from

netsmart\_NG\_ports

where

AID is not null

group by

tid,

AID

)

where port\_count > 1

)

select

1

from

port\_counts pc

where

pc.tid = xc.tid

and pc.AID = xc.a\_port\_aid

);

-- Mark the audit rows where the there is a duplicate in the port access id values (Z Side)

update netsmart\_xconnect\_audit xc set match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', Port Access ID ''' || z\_port\_access\_id || ''' exists on more than one port in this TID. This can cause false discrepencies.',

'Port Access ID ''' || z\_port\_access\_id || ''' exists on more than one port in this TID. This can cause false discrepencies.')

where exists (

with port\_counts as

(

select \* from

(

select

tid,

AID,

count(1) as port\_count

from

netsmart\_NG\_ports

where

AID is not null

group by

tid,

AID

)

where port\_count > 1

)

select

1

from

port\_counts pc

where

pc.tid = xc.tid

and pc.AID = xc.z\_port\_access\_id

);

commit;

-- Mark the audit rows where neither side was found

UPDATE

NG\_REPORTS.netsmart\_xconnect\_audit

SET

match\_code = 'DISCREPANCY',

--match\_status = 'Neither of the two cross connected ports have an associated path in Nautilus.'

match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', Neither of the two cross connected ports have an associated path in Nautilus.',

'Neither of the two cross connected ports have an associated path in Nautilus.')

WHERE

MATCH\_STATUS IS NULL AND

(A\_COMPONENT\_ID IS NULL AND Z\_COMPONENT\_ID IS NULL);

COMMIT;

-- Mark the audit rows where A side was found but the Z side wasn't

UPDATE

NG\_REPORTS.netsmart\_xconnect\_audit XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ZSIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ZSIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.',

'One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ZSIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.')

WHERE

XC.MATCH\_STATUS IS NULL AND

A\_COMPONENT\_ID IS NOT NULL AND Z\_COMPONENT\_ID IS NULL;

COMMIT;

-- Mark the audit rows where Z side was found but the A side wasn't

UPDATE

NG\_REPORTS.netsmart\_xconnect\_audit XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ASIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', One of the cross connected ports is missing from the path in Nautilus. ' || xc.xconnect\_aside || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.',

'One of the cross connected ports is missing from the path in Nautilus. ' || xc.xconnect\_aside || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.')

WHERE

XC.MATCH\_STATUS IS NULL AND

A\_COMPONENT\_ID IS NULL AND Z\_COMPONENT\_ID IS NOT NULL;

COMMIT;

-- Mark the rows where the path\_inst ids do not match

UPDATE

NG\_REPORTS.netsmart\_xconnect\_audit XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'The A side port and Z side port of the cross connect do not belong to the same Circuit Path.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', The A side port and Z side port of the cross connect do not belong to the same Circuit Path.',

'The A side port and Z side port of the cross connect do not belong to the same Circuit Path.')

WHERE

XC.MATCH\_CODE IS NULL

AND A\_SIDE\_PATH\_INST\_ID != Z\_SIDE\_PATH\_INST\_ID;

COMMIT;

-- Mark the rows where the leg\_inst ids do not match

UPDATE

NG\_REPORTS.netsmart\_xconnect\_audit XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'The A side port and Z side port are not endpoints of a single circuit leg.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', The A side port and Z side port are not endpoints of a single circuit leg.',

'The A side port and Z side port are not endpoints of a single circuit leg.')

WHERE

XC.MATCH\_CODE IS NULL

AND A\_COMPONENT\_ID != Z\_COMPONENT\_ID;

COMMIT;

-- Mark the rows where the leg\_inst match but the member numbers do not match

UPDATE

NG\_REPORTS.netsmart\_xconnect\_audit XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'The member number of the A side port does not match the member number of the Z side port.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', The member number of the A side port does not match the member number of the Z side port.',

'The member number of the A side port does not match the member number of the Z side port.')

WHERE

A\_COMPONENT\_ID = Z\_COMPONENT\_ID

AND ((A\_CABLE\_MEMBER != Z\_CABLE\_MEMBER

OR (A\_CABLE\_MEMBER IS NULL AND Z\_CABLE\_MEMBER IS NOT NULL)

OR (A\_CABLE\_MEMBER IS NOT NULL AND Z\_CABLE\_MEMBER IS NULL)));

COMMIT;

-- Mark the rest of the Audit records as passed audit

UPDATE

NG\_REPORTS.netsmart\_xconnect\_audit XC

SET

xc.match\_code = 'MATCH',

--XC.MATCH\_STATUS = ''

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status),

'')

WHERE

A\_COMPONENT\_ID = Z\_COMPONENT\_ID

and (A\_CABLE\_MEMBER = Z\_CABLE\_MEMBER or (A\_CABLE\_MEMBER is null and Z\_CABLE\_MEMBER is null));

COMMIT;

END AUDIT\_XCONNECTS;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure Audit\_ADM\_XConnects \*

\* \*

\* Purpose: Audit FlashWave cross connects against NG \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE AUDIT\_ADM\_XCONNECTS IS

BEGIN

--DELETE FROM NETSMART\_FUJITSU\_EQUIPMENT;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_FUJITSU\_EQUIPMENT';

INSERT INTO NETSMART\_ADM\_FUJITSU\_EQUIPMENT

SELECT

NE.TID,

NE.TID\_TYPE,

NXE.NE\_LEVEL,

NXE.NE\_INST\_ID,

NXE.EQP\_REFERENCE\_ID,

NXE.SITE\_REFERENCE\_ID,

NXE.PARENT\_EQP\_REFERENCE\_ID,

NXE.EQ\_CLASS\_TYPE,

NXE.EQP\_NAME,

NXE.EQP\_MODEL,

NXE.EQP\_TYPE,

NXE.INV\_STATUS,

NXE.EQP\_VENDOR,

NXE.CONTAINER

FROM

NETSMART\_XCONNECT\_EQUIPMENT NXE

JOIN

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE ON (NE.NE\_INST\_ID = NXE.NE\_INST\_ID)

WHERE

NE.TID IS NOT NULL;

COMMIT;

UPDATE NETSMART\_ADM\_FUJITSU\_EQUIPMENT

SET PARENT\_EQP\_REFERENCE\_ID = NULL

WHERE EQP\_REFERENCE\_ID IN

(SELECT

NE2.EQP\_REFERENCE\_ID

FROM

NETSMART\_ADM\_FUJITSU\_EQUIPMENT NE2

WHERE

NE2.PARENT\_EQP\_REFERENCE\_ID IS NOT NULL

AND NOT EXISTS (SELECT

NE.EQP\_REFERENCE\_ID

FROM

NETSMART\_ADM\_FUJITSU\_EQUIPMENT NE

WHERE

NE.EQP\_REFERENCE\_ID = NE2.PARENT\_EQP\_REFERENCE\_ID

)

);

COMMIT;

--DELETE FROM NETSMART\_NG\_PORTS;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_NG\_PORTS';

INSERT INTO NETSMART\_ADM\_NG\_PORTS

with legs as (select TRAIL\_ID, COMPONENT\_ID, NAME, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%WORKING%' and upper(NAME) not like '%PROTECT%'

--and TRAIL\_ID = 3369828

union

select TRAIL\_ID, COMPONENT\_ID, NAME, 2 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

union

select a.TRAIL\_ID, a.COMPONENT\_ID, a.NAME, 3 as match\_ord

from (select TRAIL\_ID, COMPONENT\_ID, NAME, SEQUENCE, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

) a,

(select distinct TRAIL\_ID, min(SEQUENCE) min\_ro

from NG\_TOPOLOGY.TRAIL\_COMPONENT LI

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

group by TRAIL\_ID

) b

where a.TRAIL\_ID = b.TRAIL\_ID

and a.SEQUENCE = b.min\_ro),

max\_leg as (select TRAIL\_ID, min(match\_ord) as match\_ord

from legs

group by TRAIL\_ID),

legs\_unq as (select l.TRAIL\_ID, l.COMPONENT\_ID, l.NAME

from legs l, max\_leg ml

where l.TRAIL\_ID = ml.TRAIL\_ID

and l.match\_ord = ml.match\_ord)

SELECT DISTINCT

NE.TID,

NE.TID\_TYPE,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(CRD.SLOT\_NAME, '-0+', '-'), '^0+', ''), '^-', '0-') SLOT\_NAME,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(EP.PORT\_NAME, ':', '-'), '-0+', '-'), '^0+', '') PORT\_NAME,

EP.AID,

CPE.TRAIL\_ID,

LI.COMPONENT\_ID,

CPE.CABLE\_MEMBER,

CPI.A\_SITE\_ID,

S1.SITE\_NAME A\_SITE\_NAME,

CPI.Z\_SITE\_ID,

S2.SITE\_NAME Z\_SITE\_NAME,

CPI.TRAIL\_NAME,

CPI.BANDWIDTH,

CPI.STATUS PATH\_STATUS,

CPI.NEXT\_REVISION\_TRAIL\_ID,

CPI.PREVIOUS\_REVISION\_TRAIL\_ID,

NE.NE\_INST\_ID,

NE.EQP\_REFERENCE\_ID,

NE.PARENT\_EQP\_REFERENCE\_ID,

NE.SITE\_REFERENCE\_ID,

NE.EQP\_VENDOR,

NE.EQP\_MODEL,

NE.EQP\_NAME,

EP.PORT\_REFERENCE\_ID,

EP.CARD\_REFERENCE\_ID

FROM

NETSMART\_ADM\_FUJITSU\_EQUIPMENT NE

JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.EQP\_REFERENCE\_ID)

JOIN

NG\_REPORTS.PORT EP ON (EP.EQP\_REFERENCE\_ID = EQ.EQP\_REFERENCE\_ID)

JOIN

NG\_TOPOLOGY.TRAIL\_ELEMENT CPE ON CPE.ELEMENT\_REF\_ID = EP.PORT\_REFERENCE\_ID and CPE.ELEMENT\_TYPE='E'

JOIN

legs\_unq LI ON LI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_TOPOLOGY.TRAIL\_COMPONENT\_ELEMENT PLM ON PLM.COMPONENT\_ID = LI.COMPONENT\_ID AND PLM.ELEMENT\_ID = CPE.ELEMENT\_ID

JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_REPORTS.SITE S1 ON S1.SITE\_REFERENCE\_ID = CPI.A\_SITE\_ID

JOIN

NG\_REPORTS.SITE S2 ON S2.SITE\_REFERENCE\_ID = CPI.Z\_SITE\_ID

JOIN

NG\_REPORTS.CARD CRD ON (CRD.CARD\_REFERENCE\_ID = EP.CARD\_REFERENCE\_ID)

WHERE

NE.TID IS NOT NULL

AND REGEXP\_LIKE(EP.PORT\_NAME, '-')

and upper(cpi.status) = 'LIVE'

AND NE.CONTAINER= 'SHELF'

ORDER BY

TID,

PORT\_NAME;

COMMIT;

---------------------------------------------------------------

-- Insert ports where a "live" port does not already exist --

---------------------------------------------------------------

INSERT INTO NETSMART\_ADM\_NG\_PORTS

with legs as (select TRAIL\_ID, COMPONENT\_ID, NAME, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%WORKING%' and upper(NAME) not like '%PROTECT%'

--and TRAIL\_ID = 3369828

union

select TRAIL\_ID, COMPONENT\_ID, NAME, 2 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

union

select a.TRAIL\_ID, a.COMPONENT\_ID, a.NAME, 3 as match\_ord

from (select TRAIL\_ID, COMPONENT\_ID, NAME, SEQUENCE, 1 as match\_ord

from NG\_TOPOLOGY.TRAIL\_COMPONENT

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

) a,

(select distinct TRAIL\_ID, min(SEQUENCE) min\_ro

from NG\_TOPOLOGY.TRAIL\_COMPONENT LI

where upper(NAME) not like '%PROTECT%' and upper(NAME) not like '%WORKING%'

--and TRAIL\_ID = 3369828

group by TRAIL\_ID

) b

where a.TRAIL\_ID = b.TRAIL\_ID

and a.SEQUENCE = b.min\_ro),

max\_leg as (select TRAIL\_ID, min(match\_ord) as match\_ord

from legs

group by TRAIL\_ID),

legs\_unq as (select l.TRAIL\_ID, l.COMPONENT\_ID, l.NAME

from legs l, max\_leg ml

where l.TRAIL\_ID = ml.TRAIL\_ID

and l.match\_ord = ml.match\_ord)

SELECT DISTINCT

NE.TID,

NE.TID\_TYPE,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(CRD.SLOT\_NAME, '-0+', '-'), '^0+', ''), '^-', '0-') SLOT\_NAME,

REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(EP.PORT\_NAME, ':', '-'), '-0+', '-'), '^0+', '') PORT\_NAME,

EP.AID,

CPE.TRAIL\_ID,

LI.COMPONENT\_ID,

CPE.CABLE\_MEMBER,

CPI.A\_SITE\_ID,

S1.SITE\_NAME A\_SITE\_NAME,

CPI.Z\_SITE\_ID,

S2.SITE\_NAME Z\_SITE\_NAME,

CPI.TRAIL\_NAME,

CPI.BANDWIDTH,

CPI.STATUS PATH\_STATUS,

CPI.NEXT\_REVISION\_TRAIL\_ID,

CPI.PREVIOUS\_REVISION\_TRAIL\_ID,

NE.NE\_INST\_ID,

NE.EQP\_REFERENCE\_ID,

NE.PARENT\_EQP\_REFERENCE\_ID,

NE.SITE\_REFERENCE\_ID,

NE.EQP\_VENDOR,

NE.EQP\_MODEL,

NE.EQP\_NAME,

EP.PORT\_REFERENCE\_ID,

EP.CARD\_REFERENCE\_ID

FROM

NETSMART\_ADM\_FUJITSU\_EQUIPMENT NE

JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.EQP\_REFERENCE\_ID OR EQ.PARENT\_EQP\_REFERENCE\_ID = NE.EQP\_REFERENCE\_ID)

JOIN

NG\_REPORTS.PORT EP ON (EP.EQP\_REFERENCE\_ID = EQ.EQP\_REFERENCE\_ID)

JOIN

NG\_TOPOLOGY.TRAIL\_ELEMENT CPE ON CPE.ELEMENT\_REF\_ID = EP.PORT\_REFERENCE\_ID and CPE.ELEMENT\_TYPE='E'

JOIN

legs\_unq LI ON LI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_TOPOLOGY.TRAIL\_COMPONENT\_ELEMENT PLM ON PLM.COMPONENT\_ID = LI.COMPONENT\_ID AND PLM.ELEMENT\_ID = CPE.ELEMENT\_ID

JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = CPE.TRAIL\_ID

JOIN

NG\_REPORTS.SITE S1 ON S1.SITE\_REFERENCE\_ID = CPI.A\_SITE\_ID

JOIN

NG\_REPORTS.SITE S2 ON S2.SITE\_REFERENCE\_ID = CPI.Z\_SITE\_ID

JOIN

NG\_REPORTS.CARD CRD ON (CRD.CARD\_REFERENCE\_ID = EP.CARD\_REFERENCE\_ID)

WHERE

NE.TID IS NOT NULL

AND REGEXP\_LIKE(EP.PORT\_NAME, '-')

and upper(cpi.status) != 'LIVE'

AND NE.CONTAINER= 'SHELF'

AND NOT EXISTS (SELECT 1

FROM

NG\_REPORTS.NETSMART\_ADM\_NG\_PORTS NXP

WHERE

NXP.TID = NE.TID

--AND NXP.SLOT\_NAME = REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(CRD.SLOT\_NAME, '-0+', '-'), '^0+', ''), '^-', '0-')

AND NXP.PORT\_NAME = REGEXP\_REPLACE(REGEXP\_REPLACE(REGEXP\_REPLACE(EP.PORT\_NAME, ':', '-'), '-0+', '-'), '^0+', '')

)

ORDER BY

TID,

PORT\_NAME;

COMMIT;

--DELETE FROM netsmart\_xconnect\_audit;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_XCONNECT\_AUDIT';

INSERT INTO NETSMART\_ADM\_XCONNECT\_AUDIT

SELECT

NE.TID,

NE.TID\_TYPE,

NULL MATCH\_CODE,

NULL MATCH\_STATUS,

XC.XCONNECT\_ASIDE,

XP1.NE\_INST\_ID A\_EQUIPMENT\_ID,

XP1.TRAIL\_ID A\_SIDE\_PATH\_INST\_ID,

XP1.COMPONENT\_ID A\_COMPONENT\_ID,

XP1.CABLE\_MEMBER A\_CABLE\_MEMBER,

XP1.PORT\_NAME A\_PORT\_HUM\_ID,

XP1.AID a\_port\_aid,

XC.XCONNECT\_ZSIDE,

XP2.NE\_INST\_ID Z\_EQUIPMENT\_ID,

XP2.TRAIL\_ID Z\_SIDE\_PATH\_INST\_ID,

XP2.COMPONENT\_ID A\_COMPONENT\_ID,

XP2.CABLE\_MEMBER Z\_CABLE\_MEMBER,

XP2.PORT\_NAME Z\_PORT\_HUM\_ID,

XP2.AID Z\_PORT\_ACCESS\_ID,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.TRAIL\_ID

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.TRAIL\_ID

ELSE NULL

END TRAIL\_ID,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.TRAIL\_NAME

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.TRAIL\_NAME

ELSE NULL

END TRAIL\_NAME,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.PATH\_STATUS

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.PATH\_STATUS

ELSE NULL

END PATH\_STATUS,

CASE

WHEN XP1.TRAIL\_ID IS NOT NULL

THEN XP1.BANDWIDTH

WHEN XP2.TRAIL\_ID IS NOT NULL

THEN XP2.BANDWIDTH

ELSE NULL

END BANDWIDTH

FROM

NETSMART\_SNCNE NE

JOIN

NETSMART\_XCONNECT XC ON XC.SNCNE\_ID = NE.INST\_ID

LEFT OUTER JOIN

NETSMART\_ADM\_NG\_PORTS XP1 ON XP1.TID = NE.TID AND (((XP1.PORT\_NAME = XC.XCONNECT\_ASIDE) OR (XP1.SLOT\_NAME || '-' || XP1.PORT\_NAME = XC.XCONNECT\_ASIDE)) OR

(XP1.AID = XC.XCONNECT\_ASIDE))

LEFT OUTER JOIN

NETSMART\_ADM\_NG\_PORTS XP2 ON XP2.TID = NE.TID AND (((XP2.PORT\_NAME = XC.XCONNECT\_ZSIDE) OR (XP2.SLOT\_NAME || '-' || XP2.PORT\_NAME = XC.XCONNECT\_ZSIDE)) OR

(XP2.AID = XC.XCONNECT\_ZSIDE))

WHERE

NOT REGEXP\_LIKE(NE.TID,'^[[:alnum:]]{8}MS')

ORDER BY

NE.TID,

XCONNECT\_ASIDE,

XCONNECT\_ZSIDE;

COMMIT;

-- Delete the audit rows where xconnect a size = xconnect z\_side (loopback)

DELETE FROM

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT

WHERE

XCONNECT\_ASIDE = XCONNECT\_ZSIDE AND

((A\_EQUIPMENT\_ID = Z\_EQUIPMENT\_ID) OR (A\_EQUIPMENT\_ID IS NULL AND Z\_EQUIPMENT\_ID IS NULL));

COMMIT;

-- Mark the audit rows where the there is a duplicate in the port access id values (A Side)

update NETSMART\_ADM\_XCONNECT\_AUDIT xc set match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', Port Access ID ''' || a\_port\_aid || ''' exists on more than one port in this TID. This can cause false discrepencies.',

'Port Access ID ''' || a\_port\_aid || ''' exists on more than one port in this TID. This can cause false discrepencies.')

where exists (

with port\_counts as

(

select \* from

(

select

tid,

AID,

count(1) as port\_count

from

netsmart\_adm\_NG\_ports

where

AID is not null

group by

tid,

AID

)

where port\_count > 1

)

select

1

from

port\_counts pc

where

pc.tid = xc.tid

and pc.AID = xc.a\_port\_aid

);

-- Mark the audit rows where the there is a duplicate in the port access id values (Z Side)

update NETSMART\_ADM\_XCONNECT\_AUDIT xc set match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', Port Access ID ''' || z\_port\_access\_id || ''' exists on more than one port in this TID. This can cause false discrepencies.',

'Port Access ID ''' || z\_port\_access\_id || ''' exists on more than one port in this TID. This can cause false discrepencies.')

where exists (

with port\_counts as

(

select \* from

(

select

tid,

AID,

count(1) as port\_count

from

netsmart\_adm\_NG\_ports

where

AID is not null

group by

tid,

AID

)

where port\_count > 1

)

select

1

from

port\_counts pc

where

pc.tid = xc.tid

and pc.AID = xc.z\_port\_access\_id

);

commit;

-- Mark the audit rows where neither side was found

UPDATE

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT

SET

match\_code = 'DISCREPANCY',

--match\_status = 'Neither of the two cross connected ports have an associated path in Nautilus.'

match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', Neither of the two cross connected ports have an associated path in Nautilus.',

'Neither of the two cross connected ports have an associated path in Nautilus.')

WHERE

MATCH\_STATUS IS NULL AND

(A\_COMPONENT\_ID IS NULL AND Z\_COMPONENT\_ID IS NULL);

COMMIT;

-- Mark the audit rows where A side was found but the Z side wasn't

UPDATE

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ZSIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ZSIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.',

'One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ZSIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.')

WHERE

XC.MATCH\_STATUS IS NULL AND

A\_COMPONENT\_ID IS NOT NULL AND Z\_COMPONENT\_ID IS NULL;

COMMIT;

-- Mark the audit rows where Z side was found but the A side wasn't

UPDATE

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'One of the cross connected ports is missing from the path in Nautilus. ' || XC.XCONNECT\_ASIDE || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', One of the cross connected ports is missing from the path in Nautilus. ' || xc.xconnect\_aside || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.',

'One of the cross connected ports is missing from the path in Nautilus. ' || xc.xconnect\_aside || ' not found in the circuit path. Remove cross connect from the MSPP or correct circuit path elements in Nautilus.')

WHERE

XC.MATCH\_STATUS IS NULL AND

A\_COMPONENT\_ID IS NULL AND Z\_COMPONENT\_ID IS NOT NULL;

COMMIT;

-- Mark the rows where the path\_inst ids do not match

UPDATE

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'The A side port and Z side port of the cross connect do not belong to the same Circuit Path.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', The A side port and Z side port of the cross connect do not belong to the same Circuit Path.',

'The A side port and Z side port of the cross connect do not belong to the same Circuit Path.')

WHERE

XC.MATCH\_CODE IS NULL

AND A\_SIDE\_PATH\_INST\_ID != Z\_SIDE\_PATH\_INST\_ID;

COMMIT;

-- Mark the rows where the leg\_inst ids do not match

UPDATE

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'The A side port and Z side port are not endpoints of a single circuit leg.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', The A side port and Z side port are not endpoints of a single circuit leg.',

'The A side port and Z side port are not endpoints of a single circuit leg.')

WHERE

XC.MATCH\_CODE IS NULL

AND A\_COMPONENT\_ID != Z\_COMPONENT\_ID;

COMMIT;

-- Mark the rows where the leg\_inst match but the member numbers do not match

UPDATE

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

SET

xc.match\_code = 'DISCREPANCY',

--XC.MATCH\_STATUS = 'The member number of the A side port does not match the member number of the Z side port.'

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status)||', The member number of the A side port does not match the member number of the Z side port.',

'The member number of the A side port does not match the member number of the Z side port.')

WHERE

A\_COMPONENT\_ID = Z\_COMPONENT\_ID

AND ((A\_CABLE\_MEMBER != Z\_CABLE\_MEMBER

OR (A\_CABLE\_MEMBER IS NULL AND Z\_CABLE\_MEMBER IS NOT NULL)

OR (A\_CABLE\_MEMBER IS NOT NULL AND Z\_CABLE\_MEMBER IS NULL)));

COMMIT;

-- Mark the rest of the Audit records as passed audit

UPDATE

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

SET

xc.match\_code = 'MATCH',

--XC.MATCH\_STATUS = ''

xc.match\_status = nvl2(MATCH\_STATUS,

rtrim(match\_status),

'')

WHERE

A\_COMPONENT\_ID = Z\_COMPONENT\_ID

AND (A\_CABLE\_MEMBER = Z\_CABLE\_MEMBER OR (A\_CABLE\_MEMBER IS NULL AND Z\_CABLE\_MEMBER IS NULL));

COMMIT;

END AUDIT\_ADM\_XCONNECTS;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Function NS\_Summary\_Tbl \*

\* \*

\* Purpose: Outputs NetSmart MSPP Summary Report Table \*

\* \*

\* Inputs: None \*

\* Outputs: NetSmart\_Summary\_Tbl \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_SUMMARY\_TBL RETURN NETSMART\_SUMMARY\_TBL PIPELINED IS

OUT\_REC NETSMART\_SUMMARY\_REC := NETSMART\_SUMMARY\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR NETSMART\_CUR IS

SELECT

CASE

WHEN SUB\_MARKET IS NOT NULL

THEN SUB\_MARKET

WHEN MARKET\_TERRITORY IS NOT NULL

THEN MARKET\_TERRITORY

WHEN TERRITORY IS NOT NULL

THEN TERRITORY

ELSE NULL

END TERRITORY\_MARKET\_SUB\_MARKET,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

TOTAL\_NE,

DUPLICATE\_NE,

MATCHED\_NE,

PERCENT\_MATCHED,

PASSED\_AUDIT,

NAMING\_COMPLIANCE,

DISCOVERED\_XCONNECTS,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

TOTAL\_COMPLIANCE

FROM

NETSMART\_EQ\_SUMM\_TMSM;

BEGIN

OPEN NETSMART\_CUR;

LOOP

FETCH NETSMART\_CUR INTO OUT\_REC.TERRITORY\_MARKET\_SUB\_MARKET, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.TOTAL\_NE, OUT\_REC.DUPLICATE\_NE, OUT\_REC.MATCHED\_NE,

OUT\_REC.PERCENT\_MATCHED, OUT\_REC.PASSED\_AUDIT, OUT\_REC.NAMING\_COMPLIANCE, OUT\_REC.DISCOVERED\_XCONNECTS, OUT\_REC.LIVE\_XCONNECTS, OUT\_REC.OTHER\_XCONNECTS,

OUT\_REC.TOTAL\_COMPLIANCE;

EXIT WHEN NETSMART\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE NETSMART\_CUR;

RETURN;

END NS\_SUMMARY\_TBL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Function NS\_Summary\_Tbl \*

\* \*

\* Purpose: Outputs NetSmart ADM Summary Report Table \*

\* \*

\* Inputs: None \*

\* Outputs: NetSmart\_Summary\_Tbl \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_ADM\_SUMMARY\_TBL RETURN NETSMART\_SUMMARY\_TBL PIPELINED IS

OUT\_REC NETSMART\_SUMMARY\_REC := NETSMART\_SUMMARY\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR NETSMART\_CUR IS

SELECT

CASE

WHEN SUB\_MARKET IS NOT NULL

THEN SUB\_MARKET

WHEN MARKET\_TERRITORY IS NOT NULL

THEN MARKET\_TERRITORY

WHEN TERRITORY IS NOT NULL

THEN TERRITORY

ELSE NULL

END TERRITORY\_MARKET\_SUB\_MARKET,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

TOTAL\_NE,

DUPLICATE\_NE,

MATCHED\_NE,

PERCENT\_MATCHED,

PASSED\_AUDIT,

NAMING\_COMPLIANCE,

DISCOVERED\_XCONNECTS,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS ,

TOTAL\_COMPLIANCE

FROM

NETSMART\_ADM\_EQ\_SUMM\_TMSM;

BEGIN

OPEN NETSMART\_CUR;

LOOP

FETCH NETSMART\_CUR INTO OUT\_REC.TERRITORY\_MARKET\_SUB\_MARKET, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.TOTAL\_NE, OUT\_REC.DUPLICATE\_NE,

OUT\_REC.MATCHED\_NE, OUT\_REC.PERCENT\_MATCHED, OUT\_REC.PASSED\_AUDIT, OUT\_REC.NAMING\_COMPLIANCE, OUT\_REC.DISCOVERED\_XCONNECTS, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.TOTAL\_COMPLIANCE;

EXIT WHEN NETSMART\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE NETSMART\_CUR;

RETURN;

END NS\_ADM\_SUMMARY\_TBL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_DETAIL\_TBL \*

\* \*

\* Parameters \*

\* SUB\_MARKET\_IN SUB\_MARKET Name (from CLLI\_DOMAIN\_MAP) \*

\* use NULL for all SUB\_MARKETs. \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_DETAIL\_TBL(SUB\_MARKET\_IN IN VARCHAR2, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED IS

OUT\_REC NETSMART\_DETAIL\_REC := NETSMART\_DETAIL\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR ALL\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (SELECT

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET IS NOT NULL

GROUP BY

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

UNION

SELECT

'unknown' TERRITORY,

'unknown' MARKET\_TERRITORY,

'unknown' SUB\_MARKET,

NULL LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET IS NULL

GROUP BY

'unknown',

'unknown',

'unknown',

NULL,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

)

ORDER BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR UNKNOWN\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (SELECT

'unknown' TERRITORY,

'unknown' MARKET\_TERRITORY,

'unknown' SUB\_MARKET,

'UNKNOWN\_DOMAIN' LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET IS NULL

GROUP BY

'unknown',

'unknown',

'unknown',

'UNKNOWN\_DOMAIN',

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

ORDER BY

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR SUB\_MARKET\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (SELECT

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET = SUB\_MARKET\_IN

GROUP BY

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

ORDER BY

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

BEGIN

IF (SUB\_MARKET\_IN IS NULL) THEN

OPEN ALL\_CUR;

LOOP

FETCH ALL\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN ALL\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE ALL\_CUR;

ELSIF (UPPER(SUB\_MARKET\_IN) = 'UNKNOWN') THEN

OPEN UNKNOWN\_CUR;

LOOP

FETCH UNKNOWN\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN UNKNOWN\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE UNKNOWN\_CUR;

ELSE

OPEN SUB\_MARKET\_CUR;

LOOP

FETCH SUB\_MARKET\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN SUB\_MARKET\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE SUB\_MARKET\_CUR;

END IF ;

RETURN;

END NS\_DETAIL\_TBL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_ADM\_DETAIL\_TBL \*

\* \*

\* Parameters \*

\* SUB\_MARKET\_IN SUB\_MARKET Name (from CLLI\_DOMAIN\_MAP) \*

\* use NULL for all SUB\_MARKETs. \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_ADM\_DETAIL\_TBL(SUB\_MARKET\_IN IN VARCHAR2, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED IS

OUT\_REC NETSMART\_DETAIL\_REC := NETSMART\_DETAIL\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR ALL\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (SELECT

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_ADM\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET IS NOT NULL

GROUP BY

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

UNION

SELECT

'unknown' TERRITORY,

'unknown' MARKET\_TERRITORY,

'unknown' SUB\_MARKET,

NULL LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_ADM\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET IS NULL

GROUP BY

'unknown',

'unknown',

'unknown',

NULL,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

)

ORDER BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR UNKNOWN\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (SELECT

'unknown' TERRITORY,

'unknown' MARKET\_TERRITORY,

'unknown' SUB\_MARKET,

'UNKNOWN\_DOMAIN' LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_ADM\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET IS NULL

GROUP BY

'unknown',

'unknown',

'unknown',

'UNKNOWN\_DOMAIN',

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

ORDER BY

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR SUB\_MARKET\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (SELECT

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_ADM\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET = SUB\_MARKET\_IN

GROUP BY

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

ORDER BY

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

BEGIN

IF (SUB\_MARKET\_IN IS NULL) THEN

OPEN ALL\_CUR;

LOOP

FETCH ALL\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN ALL\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE ALL\_CUR;

ELSIF (UPPER(SUB\_MARKET\_IN) = 'UNKNOWN') THEN

OPEN UNKNOWN\_CUR;

LOOP

FETCH UNKNOWN\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN UNKNOWN\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE UNKNOWN\_CUR;

ELSE

OPEN SUB\_MARKET\_CUR;

LOOP

FETCH SUB\_MARKET\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN SUB\_MARKET\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE SUB\_MARKET\_CUR;

END IF ;

RETURN;

END NS\_ADM\_DETAIL\_TBL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_DETAIL\_TBL\_WRK (Internal) \*

\* \*

\* Parameters \*

\* SUB\_MARKET\_IN SUB\_MARKET Name (from CLLI\_DOMAIN\_MAP) \*

\* use NULL for all SUB\_MARKETs. \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_DETAIL\_TBL\_WRK(START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED IS

OUT\_REC NETSMART\_DETAIL\_REC := NETSMART\_DETAIL\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR ALL\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (

SELECT

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

netsmart\_xconnect\_audit XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NOT NULL

/\* AND UPPER(SUB\_MARKET) <> 'NNO' \*/

GROUP BY

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

UNION

SELECT

'unknown' TERRITORY,

'unknown' MARKET\_TERRITORY,

'unknown' SUB\_MARKET,

NULL LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

netsmart\_xconnect\_audit XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON ((CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME) AND (UPPER(CDM.TERRITORY) != 'NNO'))

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NULL

GROUP BY

'unknown',

'unknown',

'unknown',

NULL,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

)

ORDER BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

WHERE

ROWNUMBER BETWEEN 1 AND 10000

ORDER BY

ROWNUMBER;

BEGIN

OPEN ALL\_CUR;

LOOP

FETCH ALL\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN ALL\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE ALL\_CUR;

RETURN;

END NS\_DETAIL\_TBL\_WRK;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_ADM\_DETAIL\_TBL\_WRK (Internal) \*

\* \*

\* Parameters \*

\* SUB\_MARKET\_IN SUB\_MARKET Name (from CLLI\_DOMAIN\_MAP) \*

\* use NULL for all SUB\_MARKETs. \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_ADM\_DETAIL\_TBL\_WRK(START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_DETAIL\_TBL PIPELINED IS

OUT\_REC NETSMART\_DETAIL\_REC := NETSMART\_DETAIL\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR ALL\_CUR IS

WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TERRITORY, MARKET\_TERRITORY, SUB\_MARKET, LEAF\_DOMAIN\_NAME, TID, EQP\_NAME)) ROWNUMBER,

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

TID\_TYPE,

LOCATION,

CONTACTS,

IP\_ADDR,

MATCH\_CODE,

MATCH\_STATUS,

NE\_INST\_ID,

EQP\_NAME,

SITE\_REFERENCE\_ID,

SITE\_NAME,

EQP\_TYPE,

EQP\_VENDOR,

EQP\_MODEL,

NUM\_XCONNECT,

DISCREPANCY,

LIVE\_XCONNECTS,

OTHER\_XCONNECTS,

COMPLIANCE

FROM (

SELECT

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_ADM\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NOT NULL

/\* AND UPPER(SUB\_MARKET) <> 'NNO' \*/

GROUP BY

CDM.TERRITORY,

CDM.MARKET\_TERRITORY,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

UNION

SELECT

'unknown' TERRITORY,

'unknown' MARKET\_TERRITORY,

'unknown' SUB\_MARKET,

NULL LEAF\_DOMAIN\_NAME,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) NUM\_XCONNECT,

SUM(DECODE(CASE WHEN XC.XCONNECT\_ASIDE IS NOT NULL THEN XC.XCONNECT\_ASIDE||','||XC.XCONNECT\_ZSIDE ELSE NULL END, NULL, 0, 1)) - SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) DISCREPANCY,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) = 'LIVE'

THEN 1

ELSE 0

END) LIVE\_XCONNECTS,

SUM(CASE WHEN UPPER(TRIM(XC.MATCH\_CODE)) = 'MATCH' AND UPPER(XC.PATH\_STATUS) != 'LIVE'

THEN 1

ELSE 0

END) OTHER\_XCONNECTS,

CASE

WHEN SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1)) != 0

THEN TO\_CHAR(ROUND((SUM(DECODE(UPPER(XC.MATCH\_CODE), 'MATCH', 1, 0)) / SUM(DECODE(XC.XCONNECT\_ASIDE, NULL, 0, 1))) \* 100,2), '990.00')

ELSE

TO\_CHAR(0,'990.00')

END COMPLIANCE

FROM

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE

LEFT OUTER JOIN

NETSMART\_ADM\_XCONNECT\_AUDIT XC ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON ((CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME) AND (UPPER(CDM.SUB\_MARKET) != 'NNO'))

LEFT OUTER JOIN

NG\_REPORTS.EQUIPMENT EQ ON (EQ.EQP\_REFERENCE\_ID = NE.NE\_INST\_ID)

LEFT OUTER JOIN

NG\_REPORTS.SITE S ON (S.SITE\_REFERENCE\_ID = EQ.SITE\_REFERENCE\_ID)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NULL

GROUP BY

'unknown',

'unknown',

'unknown',

NULL,

NE.TID,

NE.TID\_TYPE,

NE.LOCATION,

NE.CONTACTS,

NE.IP\_ADDR,

NE.MATCH\_CODE,

NE.MATCH\_STATUS,

NE.NE\_INST\_ID,

EQ.EQP\_NAME,

EQ.SITE\_REFERENCE\_ID,

S.SITE\_NAME,

EQ.EQP\_TYPE,

EQ.EQP\_VENDOR,

EQ.EQP\_MODEL

)

ORDER BY

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME,

TID,

EQP\_NAME

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

BEGIN

OPEN ALL\_CUR;

LOOP

FETCH ALL\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TERRITORY, OUT\_REC.MARKET\_TERRITORY, OUT\_REC.SUB\_MARKET, OUT\_REC.LEAF\_DOMAIN\_NAME, OUT\_REC.TID, OUT\_REC.TID\_TYPE, OUT\_REC.LOCATION, OUT\_REC.CONTACTS, OUT\_REC.IP\_ADDR, OUT\_REC.MATCH\_CODE, OUT\_REC.MATCH\_STATUS, OUT\_REC.NE\_INST\_ID,

OUT\_REC.EQP\_NAME, OUT\_REC.SITE\_REFERENCE\_ID, OUT\_REC.SITE\_NAME, OUT\_REC.EQP\_TYPE, OUT\_REC.EQP\_VENDOR, OUT\_REC.EQP\_MODEL, OUT\_REC.NUM\_XCONNECT, OUT\_REC.DISCREPANCY, OUT\_REC.LIVE\_XCONNECTS,

OUT\_REC.OTHER\_XCONNECTS, OUT\_REC.COMPLIANCE;

EXIT WHEN ALL\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE ALL\_CUR;

RETURN;

END NS\_ADM\_DETAIL\_TBL\_WRK;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_XConnect\_TBL \*

\* \*

\* Parameters \*

\* TID\_IN Target Id Name (use NULL for all Target Ids) \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_XCONNECT\_TBL(TID\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED IS

OUT\_REC NETSMART\_XCONNECT\_REC := NETSMART\_XCONNECT\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR ALL\_CUR IS

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TID,XC.XCONNECT\_ASIDE,XC.XCONNECT\_ZSIDE)) ROWNUMBER,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NG\_REPORTS.NETSMART\_XCONNECT\_AUDIT XC

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID

/\* WHERE

XC.TID = TID\_IN \*/

ORDER BY

TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR DISC\_CUR IS

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TID,XC.XCONNECT\_ASIDE,XC.XCONNECT\_ZSIDE)) ROWNUMBER,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NG\_REPORTS.NETSMART\_XCONNECT\_AUDIT XC

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID

WHERE

XC.TID = TID\_IN

AND XC.MATCH\_CODE != 'MATCH'

ORDER BY

TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR TID\_CUR IS

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TID,XC.XCONNECT\_ASIDE,XC.XCONNECT\_ZSIDE)) ROWNUMBER,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NG\_REPORTS.NETSMART\_XCONNECT\_AUDIT XC

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID

WHERE

XC.TID = TID\_IN

ORDER BY

TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

BEGIN

IF (TID\_IN IS NULL) THEN

OPEN ALL\_CUR;

LOOP

FETCH ALL\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN ALL\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE ALL\_CUR;

ELSIF (ONLY\_DISCREPANCIES\_IN = 1) THEN

OPEN DISC\_CUR;

LOOP

FETCH DISC\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN DISC\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE DISC\_CUR;

ELSE

OPEN TID\_CUR;

LOOP

FETCH TID\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN TID\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE TID\_CUR;

END IF ;

RETURN;

END NS\_XCONNECT\_TBL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_ADM\_XConnect\_TBL \*

\* \*

\* Parameters \*

\* TID\_IN Target Id Name (use NULL for all Target Ids) \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_ADM\_XCONNECT\_TBL(TID\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED IS

OUT\_REC NETSMART\_XCONNECT\_REC := NETSMART\_XCONNECT\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR ALL\_CUR IS

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TID,XC.XCONNECT\_ASIDE,XC.XCONNECT\_ZSIDE)) ROWNUMBER,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID

/\* WHERE

XC.TID = TID\_IN \*/

ORDER BY

TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR DISC\_CUR IS

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TID,XC.XCONNECT\_ASIDE,XC.XCONNECT\_ZSIDE)) ROWNUMBER,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID

WHERE

XC.TID = TID\_IN

AND XC.MATCH\_CODE != 'MATCH'

ORDER BY

TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR TID\_CUR IS

SELECT

\*

FROM (SELECT

(ROW\_NUMBER () OVER (ORDER BY TID,XC.XCONNECT\_ASIDE,XC.XCONNECT\_ZSIDE)) ROWNUMBER,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NG\_REPORTS.NETSMART\_ADM\_XCONNECT\_AUDIT XC

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID

WHERE

XC.TID = TID\_IN

ORDER BY

TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

BEGIN

IF (TID\_IN IS NULL) THEN

OPEN ALL\_CUR;

LOOP

FETCH ALL\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN ALL\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE ALL\_CUR;

ELSIF (ONLY\_DISCREPANCIES\_IN = 1) THEN

OPEN DISC\_CUR;

LOOP

FETCH DISC\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN DISC\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE DISC\_CUR;

ELSE

OPEN TID\_CUR;

LOOP

FETCH TID\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN TID\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE TID\_CUR;

END IF ;

RETURN;

END NS\_ADM\_XCONNECT\_TBL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_XCONNECT\_SUMMARY \*

\* \*

\* Parameters \*

\* SUB\_MARKET\_IN SUB\_MARKET Name \*

\* ONLY\_DISCREPANCIES\_IN (1 = true, 0 = false) \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_XCONNECT\_SUMMARY(SUB\_MARKET\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED IS

OUT\_REC NETSMART\_XCONNECT\_REC := NETSMART\_XCONNECT\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR SUB\_MARKET\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

(ROW\_NUMBER () OVER (ORDER BY CDM.SUB\_MARKET, NE.LEAF\_DOMAIN\_NAME, XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET = SUB\_MARKET\_IN

/\*AND XC.MATCH\_CODE <> 'MATCH'\*/

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR DISC\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

(ROW\_NUMBER () OVER (ORDER BY CDM.SUB\_MARKET, NE.LEAF\_DOMAIN\_NAME, XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET = SUB\_MARKET\_IN

AND XC.MATCH\_CODE <> 'MATCH'

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR UNKNOWN\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (

SELECT

(ROW\_NUMBER () OVER (ORDER BY 'unknown', 'unknown', XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

'unknown' SUB\_MARKET,

'UNKNOWN\_DOMAIN' LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NULL

/\*AND XC.MATCH\_CODE <> 'MATCH'\*/

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR UNKNOWN\_DISC\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (

SELECT

(ROW\_NUMBER () OVER (ORDER BY 'unknown', 'UNKNOWN\_DOMAIN', XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

'unknown' SUB\_MARKET,

'UNKNOWN\_DOMAIN' LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NULL

AND XC.MATCH\_CODE <> 'MATCH'

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

BEGIN

IF (ONLY\_DISCREPANCIES\_IN = 1) THEN

IF (SUB\_MARKET\_IN = 'Unknown') THEN

OPEN DISC\_CUR;

LOOP

FETCH UNKNOWN\_DISC\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN UNKNOWN\_DISC\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE UNKNOWN\_DISC\_CUR;

ELSE

OPEN DISC\_CUR;

LOOP

FETCH DISC\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN DISC\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE DISC\_CUR;

END IF;

ELSE

IF (SUB\_MARKET\_IN = 'Unknown') THEN

OPEN UNKNOWN\_CUR;

LOOP

FETCH UNKNOWN\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN UNKNOWN\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE UNKNOWN\_CUR;

ELSE

OPEN SUB\_MARKET\_CUR;

LOOP

FETCH SUB\_MARKET\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN SUB\_MARKET\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE SUB\_MARKET\_CUR;

END IF ;

END IF ;

RETURN;

END NS\_XCONNECT\_SUMMARY;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Table Function NS\_ADM\_XCONNECT\_SUMMARY \*

\* \*

\* Parameters \*

\* SUB\_MARKET\_IN SUB\_MARKET Name \*

\* ONLY\_DISCREPANCIES\_IN (1 = true, 0 = false) \*

\* START\_IN Starting Row Number \*

\* STOP\_IN Ending Row Number \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FUNCTION NS\_ADM\_XCONNECT\_SUMMARY(SUB\_MARKET\_IN IN VARCHAR2, ONLY\_DISCREPANCIES\_IN IN NUMBER, START\_IN IN NUMBER, STOP\_IN IN NUMBER) RETURN NETSMART\_XCONNECT\_TBL PIPELINED IS

OUT\_REC NETSMART\_XCONNECT\_REC := NETSMART\_XCONNECT\_REC(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

CURSOR SUB\_MARKET\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

(ROW\_NUMBER () OVER (ORDER BY CDM.SUB\_MARKET, NE.LEAF\_DOMAIN\_NAME, XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_ADM\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET = SUB\_MARKET\_IN

/\*AND XC.MATCH\_CODE <> 'MATCH'\*/

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR DISC\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (WITH DOMAINS AS

(

SELECT DISTINCT

TERRITORY,

MARKET\_TERRITORY,

SUB\_MARKET,

LEAF\_DOMAIN\_NAME

FROM

CLLI\_DOMAIN\_MAP\_V

)

SELECT

(ROW\_NUMBER () OVER (ORDER BY CDM.SUB\_MARKET, NE.LEAF\_DOMAIN\_NAME, XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

CDM.SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_ADM\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

DOMAINS CDM ON (CDM.LEAF\_DOMAIN\_NAME = NE.LEAF\_DOMAIN\_NAME)

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

SUB\_MARKET = SUB\_MARKET\_IN

AND XC.MATCH\_CODE <> 'MATCH'

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR UNKNOWN\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (

SELECT

(ROW\_NUMBER () OVER (ORDER BY 'unknown', 'unknown', XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

'unknown' SUB\_MARKET,

'UNKNOWN\_DOMAIN' LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_ADM\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NULL

/\*AND XC.MATCH\_CODE <> 'MATCH'\*/

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

CURSOR UNKNOWN\_DISC\_CUR IS

SELECT

ROWNUMBER,

TID,

DEVICE\_XCONNECTS,

AUDIT\_STATUS,

TRAIL\_ID,

TRAIL\_NAME,

PATH\_STATUS,

BANDWIDTH,

COMMENTS

FROM (

SELECT

(ROW\_NUMBER () OVER (ORDER BY 'unknown', 'UNKNOWN\_DOMAIN', XC.TID, XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE)) ROWNUMBER,

'unknown' SUB\_MARKET,

'UNKNOWN\_DOMAIN' LEAF\_DOMAIN\_NAME,

XC.TID,

XC.XCONNECT\_ASIDE || ',' || XC.XCONNECT\_ZSIDE DEVICE\_XCONNECTS,

XC.MATCH\_CODE AUDIT\_STATUS,

XC.TRAIL\_ID,

CPI.TRAIL\_NAME,

CPI.STATUS PATH\_STATUS,

CPI.BANDWIDTH,

XC.MATCH\_STATUS COMMENTS

FROM

NETSMART\_ADM\_XCONNECT\_AUDIT XC

JOIN

NETSMART\_ADM\_NE\_VS\_NG\_AUDIT NE ON XC.TID = NE.TID

LEFT OUTER JOIN

NG\_TOPOLOGY.TRAIL CPI ON CPI.TRAIL\_ID = XC.TRAIL\_ID AND (XC.A\_EQUIPMENT\_ID IS NOT NULL AND XC.A\_EQUIPMENT\_ID = XC.Z\_EQUIPMENT\_ID)

WHERE

NE.LEAF\_DOMAIN\_NAME IS NULL

AND XC.MATCH\_CODE <> 'MATCH'

ORDER BY

SUB\_MARKET,

NE.LEAF\_DOMAIN\_NAME,

XC.TID,

DEVICE\_XCONNECTS

)

WHERE

ROWNUMBER BETWEEN START\_IN AND STOP\_IN

ORDER BY

ROWNUMBER;

BEGIN

IF (ONLY\_DISCREPANCIES\_IN = 1) THEN

IF (SUB\_MARKET\_IN = 'Unknown') THEN

OPEN DISC\_CUR;

LOOP

FETCH UNKNOWN\_DISC\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN UNKNOWN\_DISC\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE UNKNOWN\_DISC\_CUR;

ELSE

OPEN DISC\_CUR;

LOOP

FETCH DISC\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN DISC\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE DISC\_CUR;

END IF;

ELSE

IF (SUB\_MARKET\_IN = 'Unknown') THEN

OPEN UNKNOWN\_CUR;

LOOP

FETCH UNKNOWN\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN UNKNOWN\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE UNKNOWN\_CUR;

ELSE

OPEN SUB\_MARKET\_CUR;

LOOP

FETCH SUB\_MARKET\_CUR INTO OUT\_REC.ROWNUMBER, OUT\_REC.TID, OUT\_REC.DEVICE\_XCONNECTS, OUT\_REC.AUDIT\_STATUS, OUT\_REC.TRAIL\_ID, OUT\_REC.TRAIL\_NAME, OUT\_REC.PATH\_STATUS,

OUT\_REC.BANDWIDTH, OUT\_REC.COMMENTS;

EXIT WHEN SUB\_MARKET\_CUR%NOTFOUND;

PIPE ROW(OUT\_REC);

END LOOP;

CLOSE SUB\_MARKET\_CUR;

END IF ;

END IF ;

RETURN;

END NS\_ADM\_XCONNECT\_SUMMARY;

--

-- Both MSPP and ADM Functions

--

PROCEDURE RUNAUDIT IS

PROCESSNAME VARCHAR2(100);

BEGIN

BEGIN

PROCESSNAME := 'NETSMART\_AUDIT';

-- Tell watchDog the process has started

--WATCHDOG.UPDATEPROCESSSTART (PROCESSNAME);

BEGIN

-- Start NetSmart MSPP Audit

-- find the top level Fujitsu equipment (or containers) in Nautilus Inventory

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.GETNAUTILUSEQUIP();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.GETNAUTILUSEQUIP()');

GETNAUTILUSEQUIP ();

-- Audit the loaded device files against the Fujitsu FlashWave devices in Nautilus

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.AUDITFWEQUIP();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.AUDITFWEQUIP()');

AUDITFWEQUIP ();

-- Audit the NetSmart XConnects against Nautilus

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.AUDIT\_XCONNECTS();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.AUDIT\_XCONNECTS()');

AUDIT\_XCONNECTS ();

-- Generate the overall summary table

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.GENSUMMARY();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.GENSUMMARY()');

GENSUMMARY ();

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'Completed NetSmart MSPP Audit', 'N');

-- Completed MSPP Audit

-- Start NetSmart ADM Audit

-- find the top level NetSmart equipment (or containers) in Nautilus Inventory

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.GETADMNAUTILUSEQUIP();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.GETADMNAUTILUSEQUIP()');

GETADMNAUTILUSEQUIP ();

-- Audit the loaded device files against the Fujitsu FlashWave devices in Nautilus

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.AUDITADMFWEQUIP();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.AUDITADMFWEQUIP()');

AUDITADMFWEQUIP ();

-- Audit the NetSmart XConnects against Nautilus

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.AUDIT\_ADM\_XCONNECTS();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.AUDIT\_ADM\_XCONNECTS()');

AUDIT\_ADM\_XCONNECTS ();

-- Generate the overall summary table

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'NETSMART\_AUDIT.GENADMSUMMARY();', 'N');

DBMS\_OUTPUT.PUT\_LINE('In NETSMART\_AUDIT.GENADMSUMMARY()');

GENADMSUMMARY ();

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_TRANSIENT', 'Completed NetSmart ADM Audit', 'N');

-- Completed NetSmart ADM Audit

-- Move the data from work to the display table for testing

--MOVE\_DATA\_FROM\_WK ();

-- mark successful completion of the task

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_SUCCESS', PROCESSNAME || ' ran successfully!', 'N');

--WATCHDOG.UPDATEPROCESSEND (PROCESSNAME, 'STATUS\_SUCCESS', 'Y');

--WATCHDOG.POPULATE\_LAST\_SUCCESS\_DT\_PROC(PROCESSNAME);

DBMS\_OUTPUT.PUT\_LINE('NetSmart\_Audit.RunAudit() ran successfully!');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(SUBSTR('Error: NetSmart\_Audit.runAudit(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(SUBSTR('Error: NetSmart\_Audit.RunAudit(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

-- mark failed completion of the task

--WATCHDOG.LOGERROR(PROCESSNAME, 'STATUS\_FAILURE',' Error: NetSmart\_Audit.RunAudit(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM||'!', 'N');

--WATCHDOG.UPDATEPROCESSEND (PROCESSNAME, 'STATUS\_FAILURE', 'N');

END;

END RUNAUDIT;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Procedure Truncate\_WK \*

\* \*

\* Purpose: Truncate work tables. \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE TRUNCATE\_WK IS

BEGIN

-- changed all truncates to deletes

--delete from netsmart\_eq\_summ\_SUB\_MARKET\_wk;

--

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_EQ\_SUMM\_TMSM';

DBMS\_OUTPUT.PUT\_LINE('Deleted from NETSMART\_EQ\_SUMM\_TMSM');

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_EQ\_SUMM\_TMSM';

DBMS\_OUTPUT.PUT\_LINE('Deleted from NETSMART\_ADM\_EQ\_SUMM\_TMSM');

-- -- changed all truncates to deletes

-- delete from netsmart\_eq\_detail\_TMSM\_wk;

--

-- COMMIT;

--

-- DBMS\_OUTPUT.PUT\_LINE('Deleted from netsmart\_eq\_detail\_TMSM\_wk');

-- changed all truncates to deletes

--delete from NETSMART\_NE\_VS\_NG\_AUDIT;

--COMMIT;

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_NE\_VS\_NG\_AUDIT';

DBMS\_OUTPUT.PUT\_LINE('Deleted from NETSMART\_NE\_VS\_NG\_AUDIT ');

EXECUTE IMMEDIATE 'TRUNCATE TABLE NETSMART\_ADM\_NE\_VS\_NG\_AUDIT';

DBMS\_OUTPUT.PUT\_LINE('Deleted from NETSMART\_ADM\_NE\_VS\_NG\_AUDIT ');

END TRUNCATE\_WK;

END NETSMART\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package Body NGMLS\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."NGMLS\_AUDIT" AS

--UIAM-11128 & UIAM-11113

PROCEDURE audit\_ngmls\_equip IS

methodname VARCHAR2(30) := 'audit\_ngMLS\_equip';

message VARCHAR2(500);

sqlstmt VARCHAR2(32767);

BEGIN

BEGIN

dbms\_output.put\_line('deleting');

DELETE FROM NGMLS\_EQUIP\_AUDIT nxa

WHERE

ngmls\_vendor IN (

'CISCO',

'ALCATEL-LUCENT'

);

--delete from NGMLS\_DEVICE\_AUDIT\_ISSUES iss

-- where ngMLS\_vendor = 'CI';

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t delete records from table: NGMLS\_EQUIP\_AUDIT';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

BEGIN

INSERT INTO NGMLS\_EQUIP\_AUDIT columns (

ngmls\_vendor,

ngmls\_device\_name,

ne\_device\_name,

gi\_device\_name,

eqp\_reference\_id,

live\_in\_ng,

inv\_status,

match\_code

)

WITH ncm\_ngmlss AS (

SELECT DISTINCT

CASE

WHEN ncm.hostname IS NOT NULL THEN ncm.hostname

WHEN nvm.hostname IS NOT NULL THEN nvm.hostname

END

host\_name,

CASE

WHEN ncm.hostname IS NOT NULL

AND nvm.hostname IS NOT NULL THEN 'BOTH'

ELSE 'NE ONLY'

END

match\_code

FROM

ncm\_ngmls\_wk ncm

FULL OUTER JOIN ncm\_ngmls\_vlan\_wk nvm ON ncm.hostname = nvm.hostname

WHERE

vlan\_number IS NOT NULL

),sam\_ngmlss AS (

SELECT DISTINCT

site\_name --, act\_mgmt\_ip sam\_ngMLS\_device\_ip

,

reachability sam\_reachability,

resync\_status sam\_resync\_status,

resync\_state sam\_resync\_state

--, server\_name sam\_server\_name, last\_resyn\_start sam\_last\_resync\_start, last\_resyn\_end sam\_last\_resync\_end

FROM

sam\_ebh\_health

WHERE

site\_name LIKE '%7750-0\_'

),ng AS (

--ngmls.pattern1=^([A-Z0-9]{1,4} |)[A-Z0-9]{8}[8-9]\\dA-P-(AL|CI)-(9010|7750)-\\d{2}(\\s\*\\[[^\\]\\[]+\\])?

--ngmls.pattern2=.\*([A-Z0-9]{8}[8-9]\\dA-P-(AL|CI)-(9010|7750)-\\d{2}).\*

SELECT

ngmls.\*,

CASE

WHEN regexp\_substr(eqp\_name,'([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}') IS NOT NULL THEN regexp\_substr(eqp\_name

,'([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}')

ELSE eqp\_name

END

ngmls\_device\_name

FROM

equipment ngmls

WHERE

eqp\_vendor IN (

'CISCO',

'ALCATEL-LUCENT'

)

AND ngmls.eqp\_type = 'NGMLS'

--and regexp\_like(eqp\_name, '([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}')

) SELECT

CASE

WHEN ng.ngmls\_device\_name IS NOT NULL THEN eqp\_vendor

WHEN nfj.host\_name IS NOT NULL THEN 'CISCO'

WHEN sam.site\_name IS NOT NULL THEN 'ALCATEL-LUCENT'

END

vendor

--Need to update

,

CASE

WHEN nfj.host\_name IS NOT NULL THEN nfj.host\_name

WHEN sam.site\_name IS NOT NULL THEN sam.site\_name

WHEN ng.ngmls\_device\_name IS NOT NULL THEN ng.ngmls\_device\_name --Need to update

ELSE 'zzzz'

|| ng.eqp\_name

END

ngmls\_device\_name,

CASE

WHEN nfj.host\_name IS NOT NULL THEN nfj.host\_name

WHEN sam.site\_name IS NOT NULL THEN sam.site\_name

-- else ne\_device\_name

END

ne\_device\_name,

ng.eqp\_name gi\_device\_name,

eqp\_reference\_id,

CASE

WHEN inv\_status = 'LIVE' THEN 'Y'

WHEN inv\_status IS NOT NULL THEN 'N'

END

live\_in\_ng,

inv\_status,

CASE

WHEN nfj.host\_name IS NOT NULL

AND ng.eqp\_reference\_id IS NOT NULL

AND match\_code = 'BOTH' THEN 'BOTH'

-- it is in both NCM extracts but not in ng then NE Only

WHEN nfj.host\_name IS NOT NULL

OR sam.site\_name IS NOT NULL

AND ng.eqp\_reference\_id IS NULL

--and match\_code = 'BOTH'

THEN 'NE Only'

WHEN sam.site\_name IS NOT NULL

AND ng.eqp\_reference\_id IS NOT NULL THEN 'BOTH'

WHEN nfj.host\_name IS NULL

AND sam.site\_name IS NULL

AND ng.eqp\_reference\_id IS NOT NULL THEN 'Nautilus Only'

ELSE match\_code -- when the device is not found in both NCM extracts

END

match\_code

-- , case when ng.EQP\_REFERENCE\_ID is null then nvl2(match\_status, '15, '||match\_status, '15')

-- else nvl2(parse\_status, parse\_status || ', ' ||match\_status, match\_status) --null -- when does this condition appear

-- end match\_status

-- , parse\_status match\_status

FROM

ncm\_ngmlss nfj

FULL OUTER JOIN ng ON ng.eqp\_name LIKE '%'

|| nfj.host\_name

|| '%'

FULL OUTER JOIN sam\_ngmlss sam ON ng.eqp\_name LIKE '%'

|| sam.site\_name

|| '%';

COMMIT;

END;

BEGIN

insert\_ngmls\_equip\_issues ();

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t record CI device errors';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

END audit\_ngmls\_equip;

--UIAM-11128 & UIAM-11113

PROCEDURE insert\_ngmls\_equip\_issues IS

methodname VARCHAR2(30) := 'insert\_ngMLS\_equip\_issues';

message VARCHAR2(500);

sqlstmt VARCHAR2(32767);

BEGIN

dbms\_output.put\_line('FYI: In '

|| methodname);

BEGIN

dbms\_output.put\_line('deleting');

DELETE FROM NGMLS\_DEVICE\_AUDIT\_ISSUES;

--delete from NGMLS\_DEVICE\_AUDIT\_ISSUES iss

-- where ngMLS\_vendor = 'CI';

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t delete records from table: NGMLS\_DEVICE\_AUDIT\_ISSUES';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

BEGIN

-- if some ne\_hostname matched > 1 granite EQP\_REFERENCE\_ID then mark it

--Duplicate in NE

sqlstmt := 'insert into NGMLS\_DEVICE\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, ngMLS\_issue\_id, ngMLS\_vendor)

select hostname, 10, ''CISCO''

from NCM\_ngMLS\_wk i

group by i.hostname

having count(device\_ip) > 1'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t insert into NGMLS\_DEVICE\_AUDIT\_ISSUES issue Duplicate in NE ';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

ROLLBACK;

RAISE;

END;

BEGIN

-- if some ne\_hostname matched > 1 granite EQP\_REFERENCE\_ID then mark it

--Duplicate in NE

sqlstmt := 'insert into NGMLS\_DEVICE\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, ngMLS\_issue\_id, ngMLS\_vendor)

select site\_name, 10, ''ALCATEL-LUCENT''

from sam\_ebh\_health i

group by i.site\_name

having count(site\_name) > 1'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t insert into NGMLS\_DEVICE\_AUDIT\_ISSUES issue Duplicate in NE ';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

ROLLBACK;

RAISE;

END;

BEGIN

-- if some ne\_hostname matched > 1 granite EQP\_REFERENCE\_ID then mark it

--Duplicate in GI

sqlstmt := ' insert into NGMLS\_DEVICE\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, ngMLS\_issue\_id, ngMLS\_vendor)

select ngMLS\_device\_name, 1, ''CISCO''

from NGMLS\_EQUIP\_AUDIT i

where match\_code <> ''Nautilus Only''

group by i.ngMLS\_device\_name

having count(1) > 1'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t mark Duplicate in Granite';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

BEGIN

-- if some ne\_hostname matched > 1 granite EQP\_REFERENCE\_ID then mark it

--Duplicate in GI

sqlstmt := ' insert into NGMLS\_DEVICE\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, ngMLS\_issue\_id, ngMLS\_vendor)

select ngMLS\_device\_name, 1, ''ALCATEL-LUCENT''

from NGMLS\_EQUIP\_AUDIT i

where match\_code <> ''Nautilus Only''

group by i.ngMLS\_device\_name

having count(1) > 1'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t mark Duplicate in Granite';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

BEGIN

dbms\_output.put\_line('FYI: Checking MTS Data Std');

sqlstmt := ' insert into NGMLS\_DEVICE\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, ngMLS\_issue\_id, ngMLS\_vendor)

select eqp\_name ngMLS\_device\_name,11, eqp\_vendor

from equipment ngMLS

where eqp\_vendor IN (''CISCO'',''ALCATEL-LUCENT'') and

ngMLS.eqp\_type = ''NGMLS''

and not regexp\_like(eqp\_name, ''[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}'')'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t update eq name does not match MTS Std';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

BEGIN

-- device not found in GI

sqlstmt := ' insert into NGMLS\_DEVICE\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, ngMLS\_issue\_id, ngMLS\_vendor)

select ngMLS\_device\_name, 15, ngMLS\_vendor

from NGMLS\_EQUIP\_AUDIT aud

where aud.eqp\_reference\_id is null'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Cant insert issue ''Not in GI''';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

BEGIN

sqlstmt := 'insert into NGMLS\_DEVICE\_AUDIT\_ISSUES columns (ngMLS\_device\_name, ngMLS\_issue\_id, ngMLS\_vendor)

select distinct ngMLS\_device\_name, 12, ngMLS\_vendor

from NGMLS\_EQUIP\_AUDIT aud

where ne\_DEVICE\_NAME is null'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t insert into NGMLS\_DEVICE\_AUDIT\_ISSUES issue 12';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

ROLLBACK;

RAISE;

END;

-- add more errors for not parseable etc????????????????????

BEGIN

-- IF BOTH and has issues then mark it NE

sqlstmt := ' update NGMLS\_EQUIP\_AUDIT aud

set match\_code = ''NE Only''

where exists (select 1

from NGMLS\_DEVICE\_AUDIT\_ISSUES cdai, ngmls\_issue n

where aud.MATCH\_CODE = ''BOTH''

and aud.ngMLS\_vendor IN (''CISCO'', ''ALCATEL-LUCENT'')

and N.IS\_CRITICAL=''Y''

and N.NGMLS\_ISSUE\_ID = CDAI.NGMLS\_ISSUE\_ID

and AUD.NGMLS\_DEVICE\_NAME = CDAI.NGMLS\_DEVICE\_NAME

)'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Cant update match\_status to ''NE Only'' issue ';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

END insert\_ngmls\_equip\_issues;

--UIAM-11128 & UIAM-11113

PROCEDURE assign\_area\_region\_to\_devices IS

methodname VARCHAR2(30) := 'assign\_area\_region\_to\_devices';

message VARCHAR2(300);

updstmt VARCHAR2(32767);

CURSOR area\_region\_dev IS WITH domains AS (

SELECT DISTINCT

substr(clli,1,6) clli\_6,

territory,

market\_territory,

sub\_market,

sub\_market\_leaf,

leaf\_domain\_id

FROM

domains\_leaf\_reporting dlf

LEFT OUTER JOIN clli\_domain\_map clli ON clli.leaf\_domain\_id = dlf.domain\_id

WHERE

sub\_market <> 'NNO'

AND sub\_market <> 'OSS'

) SELECT

di.territory,

di.market\_territory,

di.sub\_market,

di.sub\_market\_leaf,

di.leaf\_domain\_id,

ngmls\_device\_name

FROM

NGMLS\_EQUIP\_AUDIT aud

JOIN domains di ON di.clli\_6 = substr(aud.ngmls\_device\_name,1,6);

CURSOR eq\_area\_region\_dev IS WITH domains AS (

SELECT DISTINCT

substr(clli,1,6) clli\_6,

territory,

market\_territory,

sub\_market,

sub\_market\_leaf,

leaf\_domain\_id

FROM

domains\_leaf\_reporting dlf

LEFT OUTER JOIN clli\_domain\_map clli ON clli.leaf\_domain\_id = dlf.domain\_id

WHERE

sub\_market <> 'NNO'

AND sub\_market <> 'OSS'

) SELECT

aud.territory,

aud.market\_territory,

aud.sub\_market,

aud.market,

di.leaf\_domain\_id,

ngmls\_device\_name

FROM

NGMLS\_EQUIP\_AUDIT aud

JOIN ng\_reports.equipment\_domain\_map edm ON edm.eqp\_reference\_id = aud.eqp\_reference\_id

JOIN domains di ON di.leaf\_domain\_id = edm.domain\_id -- di.leaf\_domain\_id = edm.DOMAIN\_ID

WHERE

aud.territory IS NULL -- why?

;

BEGIN

updstmt := 'update NGMLS\_EQUIP\_AUDIT set territory = :territory

, market\_territory = :market\_territory, sub\_market = :sub\_market, market = :sub\_market\_leaf

, leaf\_domain\_id = :leaf\_domain\_id

where ngMLS\_device\_name = :ngMLS\_device\_name'

;

FOR rec IN area\_region\_dev LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,rec.market\_territory,rec.sub\_market,rec.sub\_market\_leaf,rec.leaf\_domain\_id,rec.ngmls\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market for '

|| rec.ngmls\_device\_name

|| ' in table: NGMLS\_EQUIP\_AUDIT';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

--watchdog.logerror (methodName,

-- 8000,

-- SUBSTR (message

-- || TO\_CHAR (SQLCODE)

-- || ': '

-- || SQLERRM,

-- 1,

-- 255

-- ),

-- 'Y'

-- );

ROLLBACK;

RAISE;

END;

END LOOP;

-- area region for put xng\_only based on the area/region of EQP\_REFERENCE\_ID???????????

FOR rec IN eq\_area\_region\_dev LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,rec.market\_territory,rec.sub\_market,rec.market,rec.leaf\_domain\_id,rec.ngmls\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from ng Only for '

|| rec.ngmls\_device\_name

|| ' in table: NGMLS\_EQUIP\_AUDIT';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

ROLLBACK;

RAISE;

END;

END LOOP;

-- if there still are some devices which cant be mapped to any domain then mark those as unknown

BEGIN

updstmt := ' update NGMLS\_EQUIP\_AUDIT

set territory = ''unknown''

, market\_territory = ''unknown''

, sub\_market = ''unknown''

, market = ''unknown''

, leaf\_domain\_id = 0

where territory is null '

;

EXECUTE IMMEDIATE updstmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from to unknown in table: NGMLS\_EQUIP\_AUDIT';

dbms\_output.put\_line(substr(message

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

ROLLBACK;

RAISE;

END;

END assign\_area\_region\_to\_devices;

--UIAM-11128 & UIAM-11113

PROCEDURE load\_ngMLS\_regional\_summary is

-- get the all ngMLSs found in NCM ngMLS Extract

cursor ngMLSs\_NCM is

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6,territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id

from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> 'NNO' AND sub\_market <> 'OSS'

)

,NCM as(

select distinct nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market,hostname

from ncm\_ngMLS\_wk ncm--, sam\_ebh\_health sam

left outer join DOMAINS di

on substr(hostname, 1, 6) = di.clli\_6

-- group by rollup(territory, market\_territory, sub\_market)

)

,SAM as (

select distinct nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, SITE\_NAME

from sam\_ebh\_health sam

left outer join DOMAINS di

on substr(sam.SITE\_NAME, 1, 6) = di.clli\_6

where site\_name LIKE '%7750-0\_'

-- group by rollup(territory, market\_territory, sub\_market)

)

select territory, market\_territory, sub\_market

, count (hostname) ncm\_ngMLSs

from NCM

group by rollup(territory, market\_territory, sub\_market)

union

select territory, market\_territory, sub\_market

, count(SITE\_NAME) ncm\_ngMLSs

from SAM

group by rollup(territory, market\_territory, sub\_market)

;

-- get the all ngMLSs found in NCM ngMLS-VLAN Extract

cursor ngMLSs\_VLAN\_NCM is

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6,territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> 'NNO' AND sub\_market <> 'OSS'

)

,

NCM as

(

select distinct nvl(territory, 'unknown') territory, nvl(market\_territory, 'unknown') market\_territory, nvl(sub\_market, 'unknown') sub\_market, hostname

from ncm\_ngMLS\_vlan\_wk ncm

left outer join DOMAINS di

on substr(hostname, 1, 6) = di.clli\_6

)

select territory, market\_territory, sub\_market, count (1) ncm\_ngMLSs

from NCM

group by rollup(territory, market\_territory, sub\_market)

;

-- make sure duplicate matches in GI are not considered as good match-------??????

-- DUP means > 1 Live ones in ng

-- if 1 is live and others are not, then that is NOT a DIP. Shirley's logic

-- this should be identifiable from the audit

-- get mismatches between ncm-ngMLS-vlan extract and gi

CURSOR ng\_mismatch\_ngMLSs is

WITH

--NCM GI mismatch details

SAM\_NCM\_NG\_MISMATCH as

(

-- any issue with the device is a mismatch

select distinct cdai.ngMLS\_device\_name -- distinct

from NGMLS\_DEVICE\_AUDIT\_ISSUES cdai

where ngMLS\_vendor IN ('CISCO', 'ALCATEL-LUCENT')

)

,

GUD\_MATCHES as

(

-- LR 11/07

select ngMLS\_device\_name

from NGMLS\_EQUIP\_AUDIT aud

where match\_code <> 'Nautilus Only'

minus

select cdai.ngMLS\_device\_name

from NGMLS\_DEVICE\_AUDIT\_ISSUES cdai, ngmls\_issue n

where ngMLS\_vendor IN ('CISCO', 'ALCATEL-LUCENT')

and N.IS\_CRITICAL='Y'

and N.NGMLS\_ISSUE\_ID = CDAI.NGMLS\_ISSUE\_ID

)

,

SAM\_NCM\_NG\_MATCH as

(

select aud.ngMLS\_device\_name

, case when live\_in\_ng = 'Y' then 1 else 0 end SAM\_NCM\_NG\_L\_MATCH

, case when live\_in\_ng <> 'Y' then 1 else 0 end SAM\_NCM\_NG\_NL\_MATCH

from NGMLS\_EQUIP\_AUDIT aud

join GUD\_MATCHES gud

on gud.ngMLS\_device\_name = aud.ngMLS\_device\_name

)

,

MATCH\_COUNTS AS

(

select

case when ma.ngMLS\_device\_name is not null then substr(ma.ngMLS\_device\_name, 1, 6)

when ms.ngMLS\_device\_name is not null then substr(ms.ngMLS\_device\_name, 1, 6)

end clli\_6

, count(ma.ngMLS\_device\_name) SAM\_NCM\_NG\_MATCH

, sum(ma.SAM\_NCM\_NG\_L\_MATCH) SAM\_NCM\_NG\_L\_MATCH

, sum(ma.SAM\_NCM\_NG\_NL\_MATCH) SAM\_NCM\_NG\_NL\_MATCH

, count(ms.ngMLS\_device\_name) SAM\_NCM\_NG\_MISMATCH

from SAM\_NCM\_NG\_MATCH ma

full outer join SAM\_NCM\_NG\_MISMATCH ms

on ma.ngMLS\_device\_name = ms.ngMLS\_device\_name

group by

case when ma.ngMLS\_device\_name is not null then substr(ma.ngMLS\_device\_name, 1, 6)

when ms.ngMLS\_device\_name is not null then substr(ms.ngMLS\_device\_name, 1, 6)

end

)

,

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6,territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> 'NNO' AND sub\_market <> 'OSS'

)

select territory, market\_territory, sub\_market

, sum(SAM\_NCM\_NG\_MATCH) SAM\_NCM\_NG\_MATCH

, sum(SAM\_NCM\_NG\_L\_MATCH) SAM\_NCM\_NG\_L\_MATCH

, sum(SAM\_NCM\_NG\_NL\_MATCH) SAM\_NCM\_NG\_NL\_MATCH

, sum(SAM\_NCM\_NG\_MISMATCH) SAM\_NCM\_NG\_MISMATCH

from MATCH\_COUNTS cnt

join DOMAINS di

on cnt.clli\_6 = di.clli\_6

group by rollup(territory, market\_territory, sub\_market)

;

methodName varchar2(30) := 'load\_ngMLS\_regional\_summary ';

message varchar2(300);

updStmt varchar2(32767);

updAreaStmt varchar2(32767);

updVzwStmt varchar2(32767);

insStmt varchar2(32767);

insAreaStmt varchar2(32767);

insVzwStmt varchar2(32767);

sqlStmt varchar2(32767);

l\_cnt number;

BEGIN

dbms\_output.put\_line('FYI: In '||methodName);

sqlStmt := ' delete from NGMLS\_REGION\_SUMM ';

execute immediate sqlStmt;

commit;

BEGIN

-- from ng insert all CLLI and ng# in the table

sqlStmt := 'insert into NGMLS\_REGION\_SUMM columns (territory, market\_territory, sub\_market, NG\_NGMLSS, ng\_l\_ngMLSs, ng\_nl\_ngMLSs)

WITH

DOMAINS AS

(

select distinct territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id, substr(clli, 1, 6) clli\_6

from DOMAINS\_LEAF\_REPORTING dlf

left outer join CLLI\_DOMAIN\_MAP clli

on clli.leaf\_domain\_id = dlf.domain\_id

where sub\_market <> ''NNO'' AND sub\_market <> ''OSS''

union

select ''unknown'' territory, ''unknown'' market\_territory, ''unknown'' sub\_market, ''unknown'' sub\_market\_leaf, null leaf\_domain\_inst\_id, ''unknown'' clli\_6

from dual

)

,

ne\_name as(

select case when REGEXP\_SUBSTR(eqp\_name, ''([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}'') is not null then

REGEXP\_SUBSTR(eqp\_name, ''([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}'')

else eqp\_name end

ngMLS\_device\_name

, ngMLS.eqp\_reference\_id

, nvl(case when inv\_status = ''LIVE'' then 1 else 0 end, 0) l

, nvl(case when inv\_status <> ''LIVE'' then 1 else 0 end, 0) nl

, inv\_status

from equipment ngMLS

where eqp\_vendor IN (''CISCO'',''ALCATEL-LUCENT'') and

ngMLS.eqp\_type = ''NGMLS''

)

,

ng as( select distinct ngMLS\_device\_name

, substr(ngMLS\_device\_name, 1, 6) clli\_6 , ngMLS.eqp\_reference\_id

, ngMLS.l

, ngMLS.nl

from ne\_name ngMLS

)

,

GI\_CNTS as

(

select nvl(territory, ''unknown'') territory, nvl(market\_territory, ''unknown'') market\_territory, nvl(sub\_market, ''unknown'') sub\_market, case when di.clli\_6 is null then ''unknown'' else ng.clli\_6 end clli\_6

, sum(l) l

, sum(nl) nl

from DOMAINS di

right outer join ng

on ng.clli\_6 = di.clli\_6

group by nvl(territory, ''unknown''), nvl(market\_territory, ''unknown''), nvl(sub\_market, ''unknown'')

, case when di.clli\_6 is null then ''unknown'' else ng.clli\_6 end

)

select di.territory, di.market\_territory, di.sub\_market

, sum(nvl(l, 0)) + sum (nvl(nl, 0)) ng\_ngMLS

, sum(l) ng\_l\_ngMLSs

, sum(nl) ng\_nl\_ngMLSs

from DOMAINS di

left outer join GI\_CNTS ng

on ng.clli\_6 = di.clli\_6

group by rollup(di.territory, di.market\_territory, di.sub\_market)

order by di.territory, di.market\_territory, di.sub\_market

';

execute immediate sqlStmt;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '

||' GI metrics in table: NGMLS\_REGION\_SUMM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

-- insert new or update metric in the same table for total\_ngMLSs found in NCM in the ngMLS-extract

updStmt := 'update NGMLS\_REGION\_SUMM

set ncm\_ngMLSs = :ncm\_ngMLSs

where territory = :territory and market\_territory = :market\_territory

and sub\_market = :sub\_market

';

updAreaStmt := 'update NGMLS\_REGION\_SUMM

set ncm\_ngMLSs = :ncm\_ngMLSs

where territory = :territory and market\_territory = :market\_territory

and sub\_market is null

';

updVzwStmt := 'update NGMLS\_REGION\_SUMM

set ncm\_ngMLSs = :ncm\_ngMLSs

where territory is null and market\_territory is null

and sub\_market is null

';

for rec in ngMLSs\_NCM

loop

BEGIN

-- dbms\_output.put\_line(rec.territory ||'/'|| rec.market\_territory ||'/'|| rec.sub\_market||'/'||rec.ncm\_ngMLSs);

if rec.territory is null then

-- dbms\_output.put\_line('In ngMLSs\_NCM VZW');

execute immediate updVzwStmt using rec.ncm\_ngMLSs;

else

if rec.sub\_market is null then

-- dbms\_output.put\_line('In ngMLSs\_NCM territory');

execute immediate updAreaStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory;

else

-- dbms\_output.put\_line('In ngMLSs\_NCM sub\_market');

execute immediate updStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory, rec.sub\_market;

end if;

end if;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '

|| rec.territory ||'/'|| rec.market\_territory||'/'|| rec.sub\_market||'/'||rec.ncm\_ngMLSs

||' for NCM-ngMLS-extract in table: NGMLS\_REGION\_SUMM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

commit;

-- insert new or update metric in the same table for ncm total\_ngMLSs

updStmt := 'update NGMLS\_REGION\_SUMM

set SAM\_NCM\_NGMLSS\_W\_VLAN = :SAM\_NCM\_NGMLSS\_W\_VLAN

where territory = :territory and market\_territory = :market\_territory

and sub\_market = :sub\_market

';

updAreaStmt := 'update NGMLS\_REGION\_SUMM

set SAM\_NCM\_NGMLSS\_W\_VLAN = :SAM\_NCM\_NGMLSS\_W\_VLAN

where territory = :territory and market\_territory = :market\_territory

and sub\_market is null

';

updVzwStmt := 'update NGMLS\_REGION\_SUMM

set SAM\_NCM\_NGMLSS\_W\_VLAN = :SAM\_NCM\_NGMLSS\_W\_VLAN

where territory is null and market\_territory is null

and sub\_market is null

';

for rec in ngMLSs\_VLAN\_NCM

loop

BEGIN

if rec.territory is null then execute immediate updVzwStmt using rec.ncm\_ngMLSs;

else

if rec.sub\_market is null then execute immediate updAreaStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory;

else execute immediate updStmt using rec.ncm\_ngMLSs, rec.territory, rec.market\_territory, rec.sub\_market;

end if;

end if;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update '

|| rec.territory ||'/'|| rec.market\_territory||'/'|| rec.sub\_market||'/'|| rec.ncm\_ngMLSs

||' for NCM-ngMLS-VLAN-extract in table: NGMLS\_REGION\_SUMM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

-- calc matches / mismatches metric between ncm and gi

updStmt := 'update NGMLS\_REGION\_SUMM

set SAM\_NCM\_NG\_MISMATCH = :SAM\_NCM\_NG\_MISMATCH

, SAM\_NCM\_NG\_MATCH = :SAM\_NCM\_NG\_MATCH

, SAM\_NCM\_NG\_L\_MATCH = :SAM\_NCM\_NG\_L\_MATCH

, SAM\_NCM\_NG\_NL\_MATCH = :SAM\_NCM\_NG\_NL\_MATCH

where territory = :territory and market\_territory = :market\_territory

and sub\_market = :sub\_market

';

updAreaStmt := 'update NGMLS\_REGION\_SUMM

set SAM\_NCM\_NG\_MISMATCH = :SAM\_NCM\_NG\_MISMATCH

, SAM\_NCM\_NG\_MATCH = :SAM\_NCM\_NG\_MATCH

, SAM\_NCM\_NG\_L\_MATCH = :SAM\_NCM\_NG\_L\_MATCH

, SAM\_NCM\_NG\_NL\_MATCH = :SAM\_NCM\_NG\_NL\_MATCH

where territory = :territory and market\_territory = :market\_territory

and sub\_market is null

';

updVzwStmt := 'update NGMLS\_REGION\_SUMM

set SAM\_NCM\_NG\_MISMATCH = :SAM\_NCM\_NG\_MISMATCH

, SAM\_NCM\_NG\_MATCH = :SAM\_NCM\_NG\_MATCH

, SAM\_NCM\_NG\_L\_MATCH = :SAM\_NCM\_NG\_L\_MATCH

, SAM\_NCM\_NG\_NL\_MATCH = :SAM\_NCM\_NG\_NL\_MATCH

where territory is null and market\_territory is null

and sub\_market is null

';

for rec in ng\_mismatch\_ngMLSs

loop

BEGIN

if rec.territory is null then

execute immediate updVzwStmt using rec.SAM\_NCM\_NG\_MISMATCH, rec.SAM\_NCM\_NG\_MATCH, rec.SAM\_NCM\_NG\_L\_MATCH, rec.SAM\_NCM\_NG\_NL\_MATCH;

else

if rec.sub\_market is null then

execute immediate updAreaStmt using rec.SAM\_NCM\_NG\_MISMATCH, rec.SAM\_NCM\_NG\_MATCH, rec.SAM\_NCM\_NG\_L\_MATCH, rec.SAM\_NCM\_NG\_NL\_MATCH, rec.territory, rec.market\_territory;

else

execute immediate updStmt using rec.SAM\_NCM\_NG\_MISMATCH, rec.SAM\_NCM\_NG\_MATCH, rec.SAM\_NCM\_NG\_L\_MATCH, rec.SAM\_NCM\_NG\_NL\_MATCH, rec.territory, rec.market\_territory, rec.sub\_market;

end if;

end if;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update NCM-NG mismatch in table: NGMLS\_REGION\_SUMM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

-- RAISE;

END;

END LOOP;

commit;

-- calculate the percentages

BEGIN

update NGMLS\_REGION\_SUMM

set SAM\_NCM\_NGMLSS\_WO\_VLAN = nvl(ncm\_ngMLSs,0) - nvl(SAM\_NCM\_NGMLSS\_W\_VLAN,0)

,

SAM\_NCM\_NGMLSS\_W\_VLAN\_PER = round(nvl(SAM\_NCM\_NGMLSS\_W\_VLAN, 0) / decode (nvl(ncm\_ngMLSs,0), 0, 1, nvl(ncm\_ngMLSs,0)) \* 100, 2)

, ncm\_gi\_per = round(nvl(SAM\_NCM\_NG\_MATCH, 0) \* 2 / decode ((nvl(ncm\_ngMLSs,0) + nvl(NG\_NGMLSS,0)), 0, 1, nvl(ncm\_ngMLSs,0) + nvl(ng\_ngMLSs,0)) \* 100, 2)

;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update percentages in table: NGMLS\_REGION\_SUMM';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

END load\_ngMLS\_regional\_summary;

--UIAM-11127 & UIAM-11112

PROCEDURE load\_ntls\_ngMLS\_vlan\_paths is

sqlStmt varchar2(32000);

message varchar2(250);

methodName varchar2(30) := 'load\_ntls\_ngmls\_vlan\_paths';

BEGIN

message := 'Error: '|| methodName ||'(): Can''t truncate table: ng\_ngmls\_vlan\_paths\_wk';

sqlStmt := 'truncate table ng\_ngmls\_vlan\_paths\_wk' ;

execute immediate sqlStmt;

message := '';

-- this query looks at only the 1st leg of the VLAN

-- and 1st leg of the 1Gig and 10Gig path

sqlStmt:='

INSERT INTO ng\_ngmls\_vlan\_paths\_wk columns (

eqp\_reference\_id, --ne\_inst\_id, removing EQUIP\_INST\_ID

card\_reference\_id,

ckt\_path\_reference\_id,

port\_bandwidth,

path\_type,

vlan\_reference\_id,

vlan\_name,

vlan\_number,

vlan\_status,

extract\_date

)

-- explain plan for

WITH ngmls\_gige\_paths AS (

SELECT DISTINCT

xne.eqp\_reference\_id, --xne

--xne.equip\_inst\_id,

p.card\_reference\_id, --p.CARD\_INST\_ID --, p.port\_inst\_id, pli.leg\_inst\_id,(pli.rel\_order), PLI.LEG\_NAME

cpi.trail\_id gige\_inst\_id, --cpi.circ\_path\_inst\_id gige\_inst\_id

cpi.bandwidth,

cpi.type

FROM

ng\_reports.equipment xne, --xng\_reports.xng\_ngMLS\_equip\_wk xne

ng\_reports.PORT p, --vzwnet.epa p,

ng\_topology.trail cpi, --vzwnet.circ\_path\_inst cpi,

ng\_topology.trail\_component pli, --vzwnet.path\_leg\_inst pli

ng\_topology.trail\_component\_element plm, --vzwnet.path\_leg\_member plm

ng\_topology.trail\_element cpe --vzwnet.circ\_path\_element cpe

WHERE

p.eqp\_reference\_id = xne.eqp\_reference\_id --p.equip\_inst\_id = xne.equip\_inst\_id

AND xne.eqp\_type = ''NGMLS''

AND cpi.trail\_id IN (

p.ckt\_path\_reference\_id, --p.circ\_path\_inst\_id

p.next\_path\_reference\_Id --p.next\_path\_inst\_id

)

AND cpi.bandwidth LIKE ''%Gbps''

AND pli.TRAIL\_ID = cpi.trail\_id --pli.circ\_path\_inst\_id = cip.circ\_path\_inst\_id

AND pli.component\_id = plm.component\_id --pli.leg\_inst\_id = plm.leg\_inst\_id

-- and plm.sequence = cpe.sequence

AND plm.element\_id = cpe.element\_id --plm.element\_inst\_id = cpe.element\_inst\_id \*\*confirm with KUMAR MANI

AND cpe.element\_ref\_id = p.port\_reference\_id AND cpe.element\_type=''E'' --CPE.PORT\_INST\_ID = p.port\_inst\_id

AND pli.sequence = (

SELECT

MIN(sequence) sequence

FROM

ng\_topology.trail\_component li --vzwnet.path\_leg\_inst li

WHERE

li.trail\_id = cpi.trail\_id --LI.CIRC\_PATH\_INST\_ID=cpi.circ\_path\_inst\_id

)

) --select count(\*) from ngmls\_gige\_paths

, gige\_vlan\_paths AS (

SELECT DISTINCT

gige.trail\_id gige\_inst\_id, --gige.circ\_path\_inst\_id gige\_inst\_id,

vlan.trail\_id vlan\_inst\_id, --vlan.circ\_path\_inst\_id vlan\_inst\_id,

vlan.trail\_name vlan\_hum\_id, --vlan.circ\_path\_hum\_id vlan\_hum\_id,

regexp\_substr(regexp\_substr(vlan.trail\_name,''VLAN(-|[[:space:]]|)\*[[:digit:]]+''),''[[:digit:]]+'') vlan\_number, --vlan.circ\_path\_hum\_id

vlan.status

FROM

ng\_topology.trail gige, --vzwnet.circ\_path\_inst gige,

ng\_topology.trail vlan, --vzwnet.circ\_path\_inst vlan,

ng\_topology.trail\_component pli, --vzwnet.path\_leg\_inst pli,

ng\_topology.trail\_component\_element plm, --vzwnet.path\_leg\_member plm,

ng\_topology.trail\_element cpe --vzwnet.circ\_path\_element cpe

WHERE

gige.bandwidth LIKE ''%Gbps''

AND vlan.bandwidth = ''VLAN''

AND vlan.type = ''EBH''

AND vlan.trail\_id = cpe.trail\_id --vlan.circ\_path\_inst\_id = cpe.circ\_path\_inst\_id

AND gige.trail\_id = cpe.parent\_trail\_id AND cpe.element\_type=''P'' --gige.circ\_path\_inst\_id = cpe.path\_inst\_id

AND vlan.trail\_id = pli.trail\_id --vlan.circ\_path\_inst\_id = pli.circ\_path\_inst\_id

AND pli.component\_id = plm.component\_id --pli.leg\_inst\_id = plm.leg\_inst\_id

--and plm.sequence = cpe.sequence

AND plm.element\_id = cpe.element\_id --plm.element\_inst\_id = cpe.element\_inst\_id

AND pli.sequence = ( --pli.rel\_order = (

SELECT

MIN(sequence)

FROM

ng\_topology.trail\_component pl --vzwnet.path\_leg\_inst pl

WHERE

pl.trail\_id = vlan.trail\_id --pl.circ\_path\_inst\_id = vlan.circ\_path\_inst\_id

)

) --select count(\*) from gige\_vlan\_paths

SELECT /\* parallel (10) \*/

ngp.eqp\_reference\_id,

--ngp.equip\_inst\_id,

ngp.card\_reference\_id,

ngp.gige\_inst\_id trail\_id, --circ\_path\_inst\_id

ngp.bandwidth,

ngp.type,

gvp.vlan\_inst\_id,

gvp.vlan\_hum\_id,

gvp.vlan\_number,

gvp.status vlan\_status,

SYSDATE

FROM

ngmls\_gige\_paths ngp,

gige\_vlan\_paths gvp

WHERE

ngp.gige\_inst\_id = gvp.gige\_inst\_id

';

execute immediate sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr(message||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END load\_ntls\_ngMLS\_vlan\_paths;

--UIAM-11127 & UIAM-11112

PROCEDURE audit\_CI\_ngMLS\_vlans IS

methodName varchar2(30) := 'audit\_CI\_ngMLS\_vlans';

message varchar2(500);

sqlStmt varchar2(32767);

BEGIN

dbms\_output.PUT\_LINE('FYI: In '||methodName);

BEGIN

execute immediate ('delete NGMLS\_VLAN\_AUDIT where ngMLS\_vendor in (''CISCO'', ''ALCATEL-LUCENT'')');

commit;

sqlStmt := 'insert into NGMLS\_VLAN\_AUDIT columns (ngMLS\_Device\_Name, ngMLS\_vendor

, ne\_hostName, ne\_sub\_interface\_name, ne\_port\_ip, ne\_vlan\_number

, match\_code, match\_status

, eqp\_reference\_id

, ng\_vlan\_number

, vlan\_reference\_id

, vlan\_status

)

WITH

CI\_VLANs as

(

select xnvp.\*

from ng\_ngmls\_vlan\_paths\_wk xnvp

join equipment xne

on xne.eqp\_reference\_id = xnvp.eqp\_reference\_id

where xne.eqp\_vendor in (''CISCO'')

and xnvp.vlan\_number is not null

)

,

MATCHES as

(

select aud.ngMLS\_Device\_Name, ncm.VLAN\_NUMBER ne\_vlan\_number

, sub\_interface\_name, port\_ip

, ''BOTH'' match\_code

, null match\_status

, xnvp.eqp\_reference\_id

, xnvp.vlan\_number ng\_vlan\_number

, xnvp.vlan\_reference\_id

, xnvp.vlan\_status

, aud.ngMLS\_vendor

from ncm\_ngmls\_vlan\_wk ncm

join NGMLS\_EQUIP\_AUDIT aud

on aud.ngMLS\_Device\_Name = ncm.hostname

join CI\_VLANs xnvp

on aud.eqp\_reference\_id = xnvp.eqp\_reference\_id

and ncm.VLAN\_NUMBER = xnvp.vlan\_number

where ncm.port\_ip is not null

)

,

NE\_ONLY as

(

select aud.ngMLS\_Device\_Name

, ncm.sub\_interface\_name, ncm.port\_ip

, ncm.VLAN\_NUMBER ne\_vlan\_number, ''NE Only'' match\_code

, ''Missing in Granite'' match\_status

, xnvp.eqp\_reference\_id

, xnvp.vlan\_number ng\_vlan\_number

, xnvp.vlan\_reference\_id

, null vlan\_status

, aud.ngMLS\_vendor

from ncm\_ngmls\_vlan\_wk ncm

join NGMLS\_EQUIP\_AUDIT aud

on aud.ngMLS\_Device\_Name = ncm.hostname

left outer join CI\_VLANs xnvp

on aud.eqp\_reference\_id = xnvp.eqp\_reference\_id

and ncm.VLAN\_NUMBER = xnvp.vlan\_number

where xnvp.vlan\_number is null

and ncm.VLAN\_NUMBER is not null

and ncm.port\_ip is not null

)

,

XNG\_ONLY as

(

select aud.ngMLS\_Device\_Name

, null sub\_interface\_name, null port\_ip

, ncm.VLAN\_NUMBER ne\_vlan\_number, ''Nautilus Only'' match\_code

, ''Missing in NE'' match\_status

, xnvp.eqp\_reference\_id

, xnvp.vlan\_number ng\_vlan\_number

, xnvp.vlan\_reference\_id

, xnvp.vlan\_status

, aud.ngMLS\_vendor

from CI\_VLANs xnvp

join NGMLS\_EQUIP\_AUDIT aud

on aud.eqp\_reference\_id = xnvp.eqp\_reference\_id

left outer join ncm\_ngmls\_vlan\_wk ncm

on aud.ngMLS\_Device\_Name = ncm.hostname

and ncm.VLAN\_NUMBER = xnvp.vlan\_number

where ncm.vlan\_number is null

)

select ngMLS\_Device\_Name, ngMLS\_vendor

, ngMLS\_Device\_Name, sub\_interface\_name, port\_ip, ne\_vlan\_number

, match\_code, match\_status

, eqp\_reference\_id

, ng\_vlan\_number, vlan\_reference\_id, vlan\_status

from MATCHES

union

select ngMLS\_Device\_Name, ngMLS\_vendor

, ngMLS\_Device\_Name, sub\_interface\_name, port\_ip, ne\_vlan\_number

, match\_code, match\_status

, eqp\_reference\_id

, ng\_vlan\_number, vlan\_reference\_id, vlan\_status

from NE\_ONLY

union

select ngMLS\_Device\_Name, ngMLS\_vendor

, null, sub\_interface\_name, port\_ip, ne\_vlan\_number

, match\_code, match\_status

, eqp\_reference\_id

, ng\_vlan\_number, vlan\_reference\_id, vlan\_status

from Xng\_ONLY

';

execute immediate (sqlStmt);

commit;

-- insert\_CI\_ngMLS\_vlan\_issues();

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t audit CI VLANs';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

/\*watchdog.logerror (methodName,

8000,

SUBSTR (message

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

),

'Y'

); \*/

RAISE;

END;

END audit\_CI\_ngMLS\_vlans;

--UIAM-11127 & UIAM-11112

PROCEDURE audit\_al\_ngmls\_vlans

IS

methodname VARCHAR2 (30) := 'audit\_al\_ngmls\_vlans';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

BEGIN

EXECUTE IMMEDIATE ('delete NGMLS\_VLAN\_AUDIT where ngMLS\_vendor = ''ALCATEL-LUCENT''');

COMMIT;

sqlstmt :=

'insert into NGMLS\_VLAN\_AUDIT columns (ngMLS\_Device\_Name, ngMLS\_vendor

, ne\_hostName, ne\_sub\_interface\_name, ne\_port\_ip, ne\_vlan\_number

, match\_code, match\_status

, eqp\_reference\_id

, ng\_vlan\_number

, vlan\_reference\_id

, vlan\_status

, port\_name

)

WITH al\_vlans AS (

SELECT

xnvp.\*,

CASE

WHEN regexp\_substr(eqp\_name,''([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}'') IS NOT NULL THEN regexp\_substr(eqp\_name

,''([A-Z0-9]{1-4} |)[A-Z0-9]{8}[8-9][0-9]A-P-(AL|CI)-(9010|7750)-[0-9]{2}'')

ELSE eqp\_name

END

ngmls\_device\_name

FROM

ng\_ngmls\_vlan\_paths\_wk xnvp

JOIN equipment xne ON xne.eqp\_reference\_id = xnvp.eqp\_reference\_id

WHERE

xne.eqp\_vendor = ''ALCATEL-LUCENT''

AND xnvp.vlan\_number IS NOT NULL

),matches AS (

SELECT

aud.ngmls\_device\_name,

sam.out\_encap\_value ne\_vlan\_number,

sam.displayed\_name ne\_sub\_interface\_name,

sam.node\_id ne\_port\_ip,

''BOTH'' match\_code,

NULL match\_status,

xnvp.eqp\_reference\_id,

xnvp.vlan\_number ng\_vlan\_number,

xnvp.vlan\_reference\_id,

xnvp.vlan\_status,

aud.ngmls\_vendor,

substr(sam.port\_name,0,instr(sam.port\_name,'':'') - 1) AS port\_name

FROM

sam\_ebh\_rtr sam

JOIN NGMLS\_EQUIP\_AUDIT aud ON aud.ngmls\_device\_name = sam.node\_name

JOIN al\_vlans xnvp ON aud.eqp\_reference\_id = xnvp.eqp\_reference\_id

AND sam.out\_encap\_value = xnvp.vlan\_number

WHERE

sam.node\_id IS NOT NULL

AND sam.node\_name LIKE ''%7750-0\_''

),ne\_only AS (

SELECT

aud.ngmls\_device\_name,

sam.displayed\_name ne\_sub\_interface\_name,

sam.node\_id ne\_port\_ip,

sam.out\_encap\_value ne\_vlan\_number,

''NE Only'' match\_code,

''Missing in Granite'' match\_status,

aud.eqp\_reference\_id,

NULL ng\_vlan\_number,

NULL vlan\_reference\_id,

NULL vlan\_status,

aud.ngmls\_vendor,

substr(sam.port\_name,0,instr(sam.port\_name,'':'') - 1) AS port\_name

FROM

sam\_ebh\_rtr sam

JOIN NGMLS\_EQUIP\_AUDIT aud ON aud.ngmls\_device\_name = sam.node\_name

WHERE

sam.out\_encap\_value IS NOT NULL

AND sam.node\_id IS NOT NULL

AND sam.node\_name LIKE ''%7750-0\_''

AND NOT EXISTS (

SELECT

1

FROM

al\_vlans xnvp

WHERE

aud.eqp\_reference\_id = xnvp.eqp\_reference\_id

AND xnvp.ngmls\_device\_name = aud.ngmls\_device\_name

AND sam.out\_encap\_value = xnvp.vlan\_number

)

),xng\_only AS (

SELECT

aud.ngmls\_device\_name,

NULL ne\_sub\_interface\_name,

NULL ne\_port\_ip,

NULL ne\_vlan\_number,

''Nautilus Only'' match\_code,

''Missing in NE'' match\_status,

xnvp.eqp\_reference\_id,

xnvp.vlan\_number ng\_vlan\_number,

xnvp.vlan\_reference\_id,

xnvp.vlan\_status,

aud.ngmls\_vendor,

NULL port\_name

FROM

al\_vlans xnvp

JOIN NGMLS\_EQUIP\_AUDIT aud ON aud.eqp\_reference\_id = xnvp.eqp\_reference\_id

AND aud.ngmls\_device\_name = xnvp.ngmls\_device\_name

AND NOT EXISTS (

SELECT

1

FROM

sam\_ebh\_rtr sam

WHERE

aud.ngmls\_device\_name = sam.node\_name

AND sam.out\_encap\_value = xnvp.vlan\_number

AND sam.out\_encap\_value IS NOT NULL

AND sam.node\_name LIKE ''%7750-0\_''

)

) SELECT

ngmls\_device\_name,

ngmls\_vendor,

ngmls\_device\_name,

ne\_sub\_interface\_name,

ne\_port\_ip,

ne\_vlan\_number,

match\_code,

match\_status,

eqp\_reference\_id,

ng\_vlan\_number,

vlan\_reference\_id,

vlan\_status,

port\_name

FROM

matches

UNION

SELECT

ngmls\_device\_name,

ngmls\_vendor,

ngmls\_device\_name,

ne\_sub\_interface\_name,

ne\_port\_ip,

ne\_vlan\_number,

match\_code,

match\_status,

eqp\_reference\_id,

ng\_vlan\_number,

vlan\_reference\_id,

vlan\_status,

port\_name

FROM

ne\_only

UNION

SELECT

ngmls\_device\_name,

ngmls\_vendor,

NULL,

ne\_sub\_interface\_name,

ne\_port\_ip,

ne\_vlan\_number,

match\_code,

match\_status,

eqp\_reference\_id,

ng\_vlan\_number,

vlan\_reference\_id,

vlan\_status,

port\_name

FROM

xng\_only

';

EXECUTE IMMEDIATE (sqlstmt);

COMMIT;

insert\_CI\_ngmls\_vlan\_issues ();

EXCEPTION

WHEN OTHERS

THEN

MESSAGE := 'Error: ' || methodname || '(): Can''t audit AL VLANs';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

/\*watchdog.logerror (methodName,

8000,

SUBSTR (message

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

),

'Y'

); \*/

RAISE;

END;

END audit\_al\_ngmls\_vlans;

--UIAM-11127 & UIAM-11112

PROCEDURE insert\_CI\_ngMLS\_vlan\_issues is

methodName varchar2(30) := 'insert\_CI\_ngMLS\_vlan\_issues';

message varchar2(300);

sqlStmt varchar2(32767);

CURSOR DUP\_L\_VLAN is

-- if the same DEVICE\_NAME, nxa.VLAN\_number has > 1 LIVE status then

-- mark it as dup

WITH

VLAN\_STATUS as

(

select nxa.ngMLS\_DEVICE\_NAME, nxa.ng\_vlan\_number, status, match\_code

, nxa.vlan\_reference\_id

from NGMLS\_VLAN\_AUDIT nxa

join NG\_TOPOLOGY.trail cpi

on cpi.TRAIL\_ID = nxa.vlan\_reference\_id

)

,

DUP\_L\_VLANS as

(

select nxa.ngmls\_device\_name, nxa.ng\_vlan\_number, count (vlan\_reference\_id)

from VLAN\_STATUS nxa

where status = 'LIVE'

and match\_code <> 'Nautilus Only'

group by nxa.ngmls\_device\_name, nxa.ng\_vlan\_number, status

having count(vlan\_reference\_id) > 1

)

SELECT nxa.ngmls\_device\_name, nxa.ng\_vlan\_number, status, match\_code

, nxa.vlan\_reference\_id

from DUP\_L\_VLANS dup

join VLAN\_STATUS nxa

on nxa.ngmls\_device\_name = dup.ngmls\_device\_name

and nxa.ng\_vlan\_number = dup.ng\_vlan\_number

order by nxa.ngmls\_device\_name, nxa.ng\_vlan\_number

;

-- if the same ngmls\_device\_name, nxa.VLAN\_number has > 1 status then

-- and 1 is LIVE and other is not then mark NL as 'NE Only' & 'LIVE path exists'

CURSOR LIVE\_VLAN\_EXISTS is

WITH

VLAN\_STATUS as

(

select nxa.ngmls\_device\_name, nxa.ng\_vlan\_number, status, match\_code

, nxa.vlan\_reference\_id

from NGMLS\_VLAN\_AUDIT nxa

join NG\_TOPOLOGY.trail cpi

on cpi.TRAIL\_ID = nxa.vlan\_reference\_id

)

,

L\_and\_NL\_VLANS as

(

select nxa.ngmls\_device\_name, nxa.ng\_vlan\_number

from VLAN\_STATUS nxa

where status = 'LIVE'

intersect

select nxa.ngmls\_device\_name, nxa.ng\_vlan\_number

from VLAN\_STATUS nxa

where status <> 'LIVE'

)

SELECT nxa.ngmls\_device\_name, nxa.ng\_vlan\_number, nxa.vlan\_reference\_id

from L\_and\_NL\_VLANS dup

join VLAN\_STATUS nxa

on nxa.ngmls\_device\_name = dup.ngmls\_device\_name

and nxa.ng\_vlan\_number = dup.ng\_vlan\_number

where nxa.status <> 'LIVE'

order by nxa.ngmls\_device\_name, nxa.ng\_vlan\_number

;

BEGIN

dbms\_output.PUT\_LINE('FYI: In '||methodName);

begin

execute immediate ('delete NGMLS\_VLAN\_AUDIT\_ISSUES where ngMLS\_vendor IN (''CISCO'', ''ALCATEL-LUCENT'')');

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t delete rows in table: NGMLS\_VLAN\_AUDIT\_ISSUES';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

-- mark duplicate in Granite as NE Only, or LIVE Path Exists

-- if there is a path revision then the same VLAN will be present twice.

-- BUT that is NOT a duplicate

-- if NOT a revision path and VLAN# is present multiple times THEN it is dup

-- mark duplicate in Granite as NE Only, or LIVE Path Exists

-- else 2 DUP vlans on a csr then Duplicate

sqlStmt := 'update NGMLS\_VLAN\_AUDIT

set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

, match\_status = nvl2(match\_status, match\_status|| '', Dup in GI'', ''Dup in GI'')

where vlan\_reference\_id = :p\_vlan\_reference\_id

';

for rec in DUP\_L\_VLAN

loop

BEGIN

execute immediate sqlStmt using rec.vlan\_reference\_id;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update ''Dup in GI'' in table: NGMLS\_VLAN\_AUDIT';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

commit;

begin

sqlStmt := 'insert into NGMLS\_VLAN\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, vlan\_number, ngMLS\_issue\_id, ngMLS\_vendor)

select distinct W.NGMLS\_DEVICE\_NAME

, case when ne\_vlan\_number is null then ng\_vlan\_number

else ne\_vlan\_number

end vlan\_number,1,ngMLS\_vendor

from NGMLS\_VLAN\_AUDIT w

where W.MATCH\_STATUS like ''%Dup in GI%''';

EXECUTE IMMEDIATE sqlStmt;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t insert into NGMLS\_VLAN\_AUDIT\_ISSUES issue Duplicate in GI ';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

commit;

/\* BEGIN

-- check the ne\_hostname matches MTCE std, else update match\_code = ''NE Only''

-- and match\_status = ''ne\_hostname not MTCE Stds compliant''

sqlStmt:= ' update NGMLS\_VLAN\_AUDIT aud set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

where exists (select 1

from ncm\_ngmls\_vlan\_wk i

where ngmls\_vendor in (''CISCO'', ''ALCATEL-LUCENT'')

and AUD.NGMLS\_DEVICE\_NAME = I.HOSTNAME

and parse\_status like ''%11%''

)'

;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt := 'insert into NGMLS\_VLAN\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, vlan\_number, ngMLS\_issue\_id, ngMLS\_vendor)

select distinct W.NGMLS\_DEVICE\_NAME

, case when ne\_vlan\_number is null then ng\_vlan\_number

else ne\_vlan\_number

end vlan\_number,17,''CI''

from NGMLS\_VLAN\_AUDIT w, ncm\_ngmls\_vlan\_wk n

where

W.NGMLS\_VENDOR = ''CI''

and W.NGMLS\_DEVICE\_NAME = N.HOSTNAME

and n.parse\_status like ''%11%'''

;

--and regexp\_substr(parse\_status, '[[:digit:]]+') = '11'

EXECUTE IMMEDIATE sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t update name does not match MTCE Std';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

\*/

/\*

BEGIN

-- check if ne\_hostnames' don't match first 6-char match some MTSO CLLI

-- then match\_code = ''NE Only'' and match\_status = ''1st 6-chars of don''''t match any MTSO CLLIs''

sqlStmt := 'insert into NGMLS\_VLAN\_AUDIT\_ISSUES columns (ngMLS\_device\_name, vlan\_number, ngMLS\_issue\_id, ngMLS\_vendor)

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_INST\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT ei.ngMLS\_device\_name,

SUBSTR (ei.ngMLS\_device\_name, 1, 8) ngMLS\_clli, cdm6.clli\_6 vsm\_clli

FROM NGMLS\_VLAN\_AUDIT ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.ngMLS\_device\_name, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.ngMLS\_device\_name, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.ngMLS\_device\_name

FROM NGMLS\_VLAN\_AUDIT ei

MINUS

SELECT ei.ngMLS\_device\_name

FROM NGMLS\_VLAN\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.ngMLS\_device\_name, 1, 8)

MINUS

SELECT ei.ngMLS\_device\_name

FROM NGMLS\_VLAN\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.ngMLS\_device\_name, 1, 6)

),

ISSUES AS

(

SELECT cm.ngMLS\_device\_name, 22 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.ngMLS\_device\_name, 2 issue\_id

FROM bad\_clli bc

)

SELECT distinct i.ngMLS\_device\_name, W.NE\_VLAN\_NUMBER, i.issue\_id, ''CI''

FROM ISSUES i, NGMLS\_VLAN\_AUDIT w

where i.ngMLS\_device\_name = W.NGMLS\_DEVICE\_NAME

and W.NE\_VLAN\_NUMBER is not null

';

execute immediate sqlStmt;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

null;

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t mark 1st 6-char don''t match MTSO CLLIs';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

\*/

BEGIN

-- device not found in ncm vlan extract

sqlStmt := ' insert into NGMLS\_VLAN\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, vlan\_number, ngMLS\_issue\_id, ngMLS\_vendor)

select distinct aud.NGMLS\_DEVICE\_NAME, ng\_vlan\_number, 13, ngMLS\_vendor

from NGMLS\_VLAN\_AUDIT aud

where aud.ne\_hostname is null'

;

execute immediate sqlStmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Cant insert issue ''Not in VLAN Extract''';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

BEGIN

-- device not found in GI

sqlStmt := ' insert into NGMLS\_VLAN\_AUDIT\_ISSUES

columns (ngMLS\_device\_name, vlan\_number, ngMLS\_issue\_id, ngMLS\_vendor)

select distinct aud.NGMLS\_DEVICE\_NAME, ne\_vlan\_number, 15, ngMLS\_vendor

from NGMLS\_VLAN\_AUDIT aud

where aud.vlan\_reference\_id is null'

;

execute immediate sqlStmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Cant insert issue ''Not in GI''';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END;

-- if there is a path revision then the same VLAN will be present twice.

-- BUT that is NOT a duplicate

-- if NOT a revision path and VLAN# is present multiple times THEN it is dup

-- if 1 == LIVE and the other <> LIVE then mark <> LIVE as 'LIVE in ng'

sqlStmt := 'update NCM\_NGMLS\_VLAN\_AUDIT

set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

, match\_status = nvl2(match\_status, ''Live Path Exists,'' ||match\_status , ''Live Path Exists'')

where vlan\_reference\_id = :p\_vlan\_reference\_id

--and ngmls\_vendor = :p\_ngMLS\_vendor

';

for rec in LIVE\_VLAN\_EXISTS

loop

BEGIN

execute immediate sqlStmt using rec.vlan\_reference\_id; --, p\_ngMLS\_vendor;

commit;

EXCEPTION

WHEN OTHERS then

message := 'Error: '|| methodName ||'(): Can''t update ''LIVE Path Exists'' in table: NGMLS\_VLAN\_AUDIT';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

rollback;

RAISE;

END;

end loop;

commit;

-- IF BOTH and has issues then mark it NE

sqlStmt := ' update NGMLS\_VLAN\_AUDIT aud

set match\_code = ''NE Only''

where exists (select 1

from

NGMLS\_VLAN\_AUDIT\_ISSUES vi,

ngmls\_issue n

where aud.MATCH\_CODE = ''BOTH''

and aud.ngMLS\_vendor in (''CISCO'', ''ALCATEL-LUCENT'')

and VI.NGMLS\_VENDOR in (''CISCO'', ''ALCATEL-LUCENT'')

and AUD.NGMLS\_DEVICE\_NAME = VI.NGMLS\_DEVICE\_NAME

and AUD.NE\_VLAN\_NUMBER = VI.VLAN\_NUMBER

and N.IS\_CRITICAL=''Y''

and N.NGMLS\_ISSUE\_ID = vi.NGMLS\_ISSUE\_ID

)';

execute immediate sqlStmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Cant update match\_status to ''NE Only''';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END insert\_CI\_ngMLS\_vlan\_issues;

--UIAM-11127 & UIAM-11112

PROCEDURE load\_CI\_vlan\_device\_summary IS

methodName varchar2(30) := 'load\_CI\_vlan\_device\_summary';

message varchar2(300);

sqlStmt varchar2(32767);

BEGIN

dbms\_output.PUT\_LINE('FYI: In '||methodName);

sqlStmt:= 'delete from NGMLS\_VLAN\_DEVICE\_SUMM where ngMLS\_vendor = ''CISCO''';

execute immediate sqlStmt;

commit;

-- generate summary for CI devices ONLY

sqlStmt := ' insert into NGMLS\_VLAN\_DEVICE\_SUMM

columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id

, ngmls\_device\_name, ngMLS\_Vendor

, ne\_total\_ngmls\_vlan

, matched\_l\_ngmls\_vlan

, matched\_nl\_ngmls\_vlan

, mismatched\_ngmls\_vlan

, gi\_nl\_2\_total\_vlan\_pct

, vlan\_COMPLIANCE)

WITH

ngMLS as

(

select distinct territory, market\_territory

, sub\_market, market, leaf\_domain\_id

, ngMLS\_device\_name, ngMLS\_vendor

from NGMLS\_EQUIP\_AUDIT ng

JOIN NCM\_NGMLS\_VLAN\_WK ncm

ON ng.ne\_device\_name = ncm.hostname

-- where ng.ncm\_ngmls\_vlan\_hostname is not null

),

ngMLS\_VLAN as

(

select distinct hostname NGMLS\_DEVICE\_NAME, vlan\_number

from ncm\_ngmls\_vlan\_wk ng

where ng.vlan\_number > 0

and ng.port\_ip is not null

)

,

CNT\_ngMLS\_VLANs as

(

select distinct ng.NGMLS\_DEVICE\_NAME, count(vlan\_number) total\_ne\_ngmls\_vlan

from ngMLS\_VLAN nv

join ngMLS ng

on ng.NGMLS\_DEVICE\_NAME = nv.NGMLS\_DEVICE\_NAME

group by ng.NGMLS\_DEVICE\_NAME

),

LIVE\_match as

(

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number, vlan\_status

from NGMLS\_VLAN\_AUDIT aud

where match\_code = ''BOTH''

and aud.ngMLS\_vendor = ''CISCO''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status = ''LIVE''

)

,nonLIVE\_match as

(

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number

from NGMLS\_VLAN\_AUDIT aud

where match\_code = ''BOTH''

and aud.ngMLS\_vendor = ''CISCO''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status <> ''LIVE''

minus

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number

from NGMLS\_VLAN\_AUDIT aud

where match\_code = ''BOTH''

and aud.ngMLS\_vendor = ''CISCO''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status = ''LIVE''

) ,

CNT\_BOTH\_LIVE\_VLANs as

( -- good matches

select NGMLS\_DEVICE\_NAME, nvl (count(NGMLS\_DEVICE\_NAME), 0) matched\_l\_ngmls\_vlan

from LIVE\_match aud

group by NGMLS\_DEVICE\_NAME

),

CNT\_BOTH\_NONLIVE\_VLANs as

( -- good matches

select NGMLS\_DEVICE\_NAME, nvl( count(NGMLS\_DEVICE\_NAME), 0) matched\_nl\_ngmls\_vlan

-- , nvl(sum(case when vlan\_status = ''LIVE'' then 1 else 0 end), 0) matched\_l\_ngmls\_vlan

-- , nvl(sum(case when vlan\_status <> ''LIVE'' then 1 else 0 end), 0) matched\_nl\_ngmls\_vlan

from nonLIVE\_match aud

group by aud.NGMLS\_DEVICE\_NAME

) ,

MISMATCH\_VLANs as

( -- bad matches

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number, aud.ng\_vlan\_number, vlan\_status

from NGMLS\_VLAN\_AUDIT aud

left outer join ng\_reports.NGMLS\_VLAN\_AUDIT\_ISSUES d on aud.NGMLS\_DEVICE\_NAME = d.NGMLS\_DEVICE\_NAME

where match\_code <>''BOTH''

and aud.ngMLS\_vendor =''CISCO''

and (aud.ne\_vlan\_number > 0 or aud.ng\_vlan\_number > 0)

and (aud.NE\_VLAN\_NUMBER = D.VLAN\_NUMBER or aud.ng\_vlan\_number = D.VLAN\_NUMBER)

),

CNT\_MISMATCH\_VLANs as

( -- bad match cnts

select aud.NGMLS\_DEVICE\_NAME

, count(1) mismatch\_cnt

from MISMATCH\_VLANs aud

join ngMLS ng

on ng.NGMLS\_DEVICE\_NAME = aud.NGMLS\_DEVICE\_NAME

group by aud.NGMLS\_DEVICE\_NAME

)

select territory, market\_territory, sub\_market, market, leaf\_domain\_id

, ng.ngmls\_device\_name, ngMLS\_Vendor

, sum(nvl(nv.total\_ne\_ngmls\_vlan, 0)) ne\_total\_ngmls\_vlan

, sum(nvl(matched\_l\_ngmls\_vlan, 0)) matched\_l\_ngmls\_vlan

, sum(nvl(matched\_nl\_ngmls\_vlan, 0)) matched\_nl\_ngmls\_vlan

, sum(nvl(mismatch\_cnt, 0)) mismatched\_ngmls\_vlan

, round(sum(NVL(matched\_nl\_ngmls\_vlan, 0)) / decode(sum(NVL(matched\_l\_ngmls\_vlan, 0)), 0, 1, sum(nvl(matched\_l\_ngmls\_vlan, 0)))

\* 100, 2) gi\_nl\_2\_total\_vlan\_pct

, round (sum(nvl(matched\_l\_ngmls\_vlan,0)) / decode (sum(nvl(nv.total\_ne\_ngmls\_vlan, 0)) - sum(nvl(matched\_nl\_ngmls\_vlan, 0)), 0 , 1

, sum(nvl(nv.total\_ne\_ngmls\_vlan,0)) - sum(nvl(matched\_nl\_ngmls\_vlan,0)))

\* 100 , 2) vlan\_COMPLIANCE

from ngMLS ng

left outer join CNT\_ngMLS\_VLANs nv

on nv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

left outer join CNT\_BOTH\_LIVE\_VLANs lmv

on lmv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

left outer join CNT\_BOTH\_NONLIVE\_VLANs nlmv

on nlmv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

left outer join CNT\_MISMATCH\_VLANs mv

on mv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

group by ng.territory, ng.market\_territory, ng.sub\_market, ng.market, leaf\_domain\_id, ngMLS\_Vendor, ng.NGMLS\_DEVICE\_NAME

';

execute immediate sqlStmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t summarize CI devices';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END load\_CI\_vlan\_device\_summary;

--UIAM-11127 & UIAM-11112

PROCEDURE load\_al\_vlan\_device\_summary

IS

methodname VARCHAR2 (30) := 'load\_AL\_vlan\_device\_summary';

MESSAGE VARCHAR2 (300);

sqlstmt VARCHAR2 (32767);

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

sqlstmt :=

'delete from NGMLS\_VLAN\_DEVICE\_SUMM where ngMLS\_vendor = ''ALCATEL-LUCENT''';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

-- generate summary for CI devices ONLY

sqlstmt :=

' insert into NGMLS\_VLAN\_DEVICE\_SUMM

columns (territory, market\_territory, sub\_market, market, leaf\_domain\_id

, ngmls\_device\_name, ngMLS\_Vendor

, ne\_total\_ngmls\_vlan

, matched\_l\_ngmls\_vlan

, matched\_nl\_ngmls\_vlan

, mismatched\_ngmls\_vlan

, gi\_nl\_2\_total\_vlan\_pct

, vlan\_COMPLIANCE)

WITH

ngMLS as

(

select distinct territory, market\_territory

, sub\_market, market, leaf\_domain\_id

, ngMLS\_device\_name, ngMLS\_vendor

from NGMLS\_EQUIP\_AUDIT ng

JOIN SAM\_EBH\_RTR sam

ON ng.ngMLS\_device\_name = sam.node\_name

-- where ng.NGMLS\_DEVICE\_NAME is not null

),

ngMLS\_VLAN as

(

select distinct node\_name NGMLS\_DEVICE\_NAME, OUT\_ENCAP\_VALUE vlan\_number

from SAM\_EBH\_RTR ng

where ng.OUT\_ENCAP\_VALUE > 0

and

node\_name LIKE ''%7750-0\_''

)

,

CNT\_ngMLS\_VLANs as

(

select distinct ng.NGMLS\_DEVICE\_NAME, count(vlan\_number) total\_ne\_ngmls\_vlan

from ngMLS\_VLAN nv

join ngMLS ng

on ng.NGMLS\_DEVICE\_NAME = nv.NGMLS\_DEVICE\_NAME

group by ng.NGMLS\_DEVICE\_NAME

),

LIVE\_match as

(

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number, vlan\_status

from NGMLS\_VLAN\_AUDIT aud

where match\_code = ''BOTH''

and aud.ngMLS\_vendor = ''ALCATEL-LUCENT''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status = ''LIVE''

)

,nonLIVE\_match as

(

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number

from NGMLS\_VLAN\_AUDIT aud

where match\_code = ''BOTH''

and aud.ngMLS\_vendor = ''ALCATEL-LUCENT''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status <> ''LIVE''

minus

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number

from NGMLS\_VLAN\_AUDIT aud

where match\_code = ''BOTH''

and aud.ngMLS\_vendor = ''ALCATEL-LUCENT''

and aud.ne\_vlan\_number > 0

and aud.vlan\_status = ''LIVE''

) ,

CNT\_BOTH\_LIVE\_VLANs as

( -- good matches

select NGMLS\_DEVICE\_NAME, nvl (count(NGMLS\_DEVICE\_NAME), 0) matched\_l\_ngmls\_vlan

from LIVE\_match aud

group by NGMLS\_DEVICE\_NAME

),

CNT\_BOTH\_NONLIVE\_VLANs as

( -- good matches

select NGMLS\_DEVICE\_NAME, nvl( count(NGMLS\_DEVICE\_NAME), 0) matched\_nl\_ngmls\_vlan

-- , nvl(sum(case when vlan\_status = ''LIVE'' then 1 else 0 end), 0) matched\_l\_ngmls\_vlan

-- , nvl(sum(case when vlan\_status <> ''LIVE'' then 1 else 0 end), 0) matched\_nl\_ngmls\_vlan

from nonLIVE\_match aud

group by aud.NGMLS\_DEVICE\_NAME

) ,

MISMATCH\_VLANs as

( -- bad matches

select distinct aud.NGMLS\_DEVICE\_NAME, aud.ne\_vlan\_number, aud.ng\_vlan\_number, vlan\_status

from NGMLS\_VLAN\_AUDIT aud

where match\_code <> ''BOTH''

and aud.ngMLS\_vendor = ''ALCATEL-LUCENT''

and (aud.ne\_vlan\_number > 0 or aud.ng\_vlan\_number > 0)

)

,

CNT\_MISMATCH\_VLANs as

( -- bad match cnts

select aud.NGMLS\_DEVICE\_NAME

, count(1) mismatch\_cnt

from MISMATCH\_VLANs aud

join ngMLS ng

on ng.NGMLS\_DEVICE\_NAME = aud.NGMLS\_DEVICE\_NAME

group by aud.NGMLS\_DEVICE\_NAME

)

select territory, market\_territory, sub\_market, market, leaf\_domain\_id

, ng.ngmls\_device\_name, ngMLS\_Vendor

, sum(nvl(nv.total\_ne\_ngmls\_vlan, 0)) ne\_total\_ngmls\_vlan

, sum(nvl(matched\_l\_ngmls\_vlan, 0)) matched\_l\_ngmls\_vlan

, sum(nvl(matched\_nl\_ngmls\_vlan, 0)) matched\_nl\_ngmls\_vlan

, sum(nvl(mismatch\_cnt, 0)) mismatched\_ngmls\_vlan

, round(sum(NVL(matched\_nl\_ngmls\_vlan, 0)) / decode(sum(NVL(matched\_l\_ngmls\_vlan, 0)), 0, 1, sum(nvl(matched\_l\_ngmls\_vlan, 0)))

\* 100, 2) gi\_nl\_2\_total\_vlan\_pct

, round (sum(nvl(matched\_l\_ngmls\_vlan,0)) / decode (sum(nvl(nv.total\_ne\_ngmls\_vlan, 0)) - sum(nvl(matched\_nl\_ngmls\_vlan, 0)), 0 , 1

, sum(nvl(nv.total\_ne\_ngmls\_vlan,0)) - sum(nvl(matched\_nl\_ngmls\_vlan,0)))

\* 100 , 2) vlan\_COMPLIANCE

from ngMLS ng

left outer join CNT\_ngMLS\_VLANs nv

on nv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

left outer join CNT\_BOTH\_LIVE\_VLANs lmv

on lmv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

left outer join CNT\_BOTH\_NONLIVE\_VLANs nlmv

on nlmv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

left outer join CNT\_MISMATCH\_VLANs mv

on mv.NGMLS\_DEVICE\_NAME = ng.NGMLS\_DEVICE\_NAME

group by ng.territory, ng.market\_territory, ng.sub\_market, ng.market, leaf\_domain\_id, ngMLS\_Vendor, ng.NGMLS\_DEVICE\_NAME

';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname || '(): Can''t summarize AL devices';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END load\_al\_vlan\_device\_summary;

--UIAM-11127 & UIAM-11112

PROCEDURE load\_vlan\_regional\_summary IS

methodName varchar2(30) := 'load\_vlan\_regional\_summary';

message varchar2(500);

sqlStmt varchar2(32767);

BEGIN

execute immediate 'truncate table NGMLS\_VLAN\_REGION\_SUMM';

sqlStmt:= ' insert into NGMLS\_VLAN\_REGION\_SUMM

columns (territory, market\_territory, sub\_market

, ne\_total\_ngmls\_vlan

, matched\_l\_ngmls\_vlan

, matched\_nl\_ngmls\_vlan

, mismatched\_ngmls\_vlan

, gi\_nl\_2\_total\_vlan\_pct

, vlan\_compliance)

select dlr.territory, dlr.market\_territory, dlr.sub\_market

, sum(ne\_total\_ngmls\_vlan) ne\_total\_ngmls\_vlan

, sum(matched\_l\_ngmls\_vlan) matched\_l\_ngmls\_vlan

, sum(matched\_nl\_ngmls\_vlan) matched\_nl\_ngmls\_vlan

, sum(mismatched\_ngmls\_vlan) mismatched\_ngmls\_vlan

, ROUND(SUM(NVL(MATCHED\_NL\_NGMLS\_VLAN, 0)) / DECODE(SUM(NVL(MATCHED\_L\_NGMLS\_VLAN, 0)), 0, 1, SUM(NVL(MATCHED\_L\_NGMLS\_VLAN, 0)))

\* 100, 2) GI\_NL\_2\_TOTAL\_VLAN\_PCT

, ROUND (SUM(NVL(MATCHED\_L\_NGMLS\_VLAN,0)) /

DECODE (SUM(NVL(NE\_TOTAL\_NGMLS\_VLAN, 0)) - SUM(NVL(MATCHED\_NL\_NGMLS\_VLAN, 0)), 0, 1, SUM(NVL(NE\_TOTAL\_NGMLS\_VLAN,0)) - SUM(NVL(MATCHED\_NL\_NGMLS\_VLAN,0)))

\* 100, 2) VLAN\_COMPLIANCE2

from domains\_leaf\_reporting dlr

left outer join NGMLS\_VLAN\_DEVICE\_SUMM ng

on dlr.sub\_market\_leaf = ng.market

where dlr.territory not in (''NNO'', ''OSS'')

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

ORDER BY territory, market\_territory, sub\_market '

;

execute immediate sqlStmt;

COMMIT;

sqlstmt:= 'update NGMLS\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update NGMLS\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = market\_territory where sub\_market is null and market\_territory is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update NGMLS\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = territory where sub\_market is null and market\_territory is null' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

-- sqlstmt:= 'update NGMLS\_VLAN\_REGION\_SUMM wk set market\_territory =''Unknown'',sub\_market=''Unknown'' where territory=''Unknown'' or territory=''unknown''' ;

-- EXECUTE IMMEDIATE sqlstmt;

commit;

EXCEPTION

WHEN OTHERS THEN

message := 'Error: '|| methodName ||'(): Can''t generate regional summary';

dbms\_output.put\_line(SubStr(message

||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END load\_vlan\_regional\_summary;

END ngmls\_audit;

/

--------------------------------------------------------

-- DDL for Package Body NORTEL\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."NORTEL\_AUDIT"

IS

-- Purpose: To create nortel audit tables for voice spans

PROCEDURE nt\_cdma\_audit

IS

-- Two BSM's with same MTX,DCG,SPAN is wrong

CURSOR t1\_dup IS

WITH t1\_audit AS

(SELECT ne\_span\_type, termination\_type, vsm\_device\_name\_mtx,vsm\_device\_name\_bsm,ne\_dcg\_number,ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit

WHERE ne\_span\_type = 'VOICE\_SPAN' AND termination\_type = 'T1'

)

SELECT tgt.vsm\_device\_name\_bsm, tgt.ne\_dcg\_number,tgt.ne\_span\_number

FROM t1\_audit tgt,t1\_audit src

WHERE tgt.vsm\_device\_name\_mtx = src.vsm\_device\_name\_mtx

AND tgt.ne\_dcg\_number = src.ne\_dcg\_number

AND tgt.ne\_span\_number = src.ne\_span\_number

AND tgt.vsm\_device\_name\_bsm <> src.vsm\_device\_name\_bsm;

-- Two BSM's with same MTX,DCG is wrong for EBH

CURSOR ebh\_dup IS

WITH ebh\_audit AS

(SELECT vsm\_device\_name\_mtx,vsm\_device\_name\_bsm,ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit

WHERE ne\_span\_type = 'VOICE\_SPAN' AND termination\_type = 'Ethernet'

)

SELECT tgt.vsm\_device\_name\_bsm, tgt.ne\_dcg\_number

FROM ebh\_audit tgt,ebh\_audit src

WHERE tgt.vsm\_device\_name\_mtx = src.vsm\_device\_name\_mtx

AND tgt.ne\_dcg\_number = src.ne\_dcg\_number

AND tgt.vsm\_device\_name\_bsm <> src.vsm\_device\_name\_bsm;

-- One Live and N non Live match

CURSOR t1\_nl\_match IS

SELECT vsm\_device\_name\_bsm,ne\_dcg\_number,ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit

WHERE match\_code='BOTH' AND ng\_path\_status='LIVE'

AND ne\_span\_type = 'VOICE\_SPAN'

AND termination\_type = 'T1';

-- One Live and N non Live match

CURSOR ebh\_nl\_match IS

SELECT vsm\_device\_name\_bsm,ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit

WHERE match\_code='BOTH' AND ng\_path\_status='LIVE'

AND ne\_span\_type = 'VOICE\_SPAN'

AND termination\_type = 'Ethernet';

CURSOR t1\_cdma\_multiples\_ng IS

WITH

DUPS AS

(

select distinct vsm\_device\_name\_bsm,ne\_dcg\_number,ne\_span\_number,

count(case when ng\_path\_status = 'LIVE' then 1 else null end) over (Partition by vsm\_device\_name\_bsm,ne\_dcg\_number,ne\_span\_number) as live\_count,

count(case when ng\_path\_status != 'LIVE' then 1 else null end) over (Partition by vsm\_device\_name\_bsm,ne\_dcg\_number,ne\_span\_number) as non\_live\_count

from ng\_reports.nortel\_ne\_vs\_ng\_audit src

where src.match\_code='BOTH'

and src.termination\_type='T1'

and src.ne\_span\_type='VOICE\_SPAN'

)

select tgt.rowid rd

from ng\_reports.nortel\_ne\_vs\_ng\_audit tgt

join DUPS dup

on dup.vsm\_device\_name\_bsm = tgt.vsm\_device\_name\_bsm

AND dup.ne\_dcg\_number = tgt.ne\_dcg\_number

AND dup.ne\_span\_number = tgt.ne\_span\_number

AND (dup.live\_count > 1 OR (dup.live\_count = 0 AND dup.non\_live\_count > 1)

AND tgt.termination\_type='T1'

AND tgt.match\_code='BOTH'

AND tgt.ne\_span\_type='VOICE\_SPAN'

);

cursor ethernet\_cdma\_multiples\_ng is

WITH

DUPS as

(

select distinct vsm\_device\_name\_bsm,ne\_dcg\_number,

count(case when ng\_path\_status = 'LIVE' then 1 else null end) over (Partition by vsm\_device\_name\_bsm,ne\_dcg\_number) as live\_count,

count(case when ng\_path\_status != 'LIVE' then 1 else null end) over (Partition by vsm\_device\_name\_bsm,ne\_dcg\_number) as non\_live\_count

from ng\_reports.nortel\_ne\_vs\_ng\_audit src

where src.match\_code='BOTH'

and src.termination\_type='Ethernet'

and src.ne\_span\_type='VOICE\_SPAN'

)

select tgt.rowid rd

from ng\_reports.nortel\_ne\_vs\_ng\_audit tgt

join DUPS dup

on dup.vsm\_device\_name\_bsm= tgt.vsm\_device\_name\_bsm

AND dup.ne\_dcg\_number = tgt.ne\_dcg\_number

AND (dup.live\_count > 1 OR (dup.live\_count = 0 AND dup.non\_live\_count > 1)

AND tgt.termination\_type='Ethernet'

AND tgt.match\_code='BOTH'

AND tgt.ne\_span\_type='VOICE\_SPAN'

);

cursor update\_bsm\_with\_unq\_mtx is

with

mtx\_cnt as

(select vsm\_device\_name\_mtx, count(vsm\_device\_name\_bsm)

from ng\_reports.bsm\_mtx\_map bmm

group by vsm\_device\_name\_mtx

having count(vsm\_device\_name\_bsm) = 1

)

select B.VSM\_DEVICE\_NAME\_BSM, B.VSM\_DEVICE\_NAME\_MTX

from ng\_reports.bsm\_mtx\_map b, mtx\_cnt

where mtx\_cnt.vsm\_device\_name\_mtx = B.VSM\_DEVICE\_NAME\_MTX

;

cursor update\_nt\_cdma\_wrk\_no\_csr\_port is

WITH

NXA\_PATHS as

(

select nxa.TRAIL\_ID

from ng\_reports.nortel\_ne\_vs\_ng\_audit nxa

where nxa.TERMINATION\_TYPE = 'Ethernet'

AND nxa.NG\_BANDWIDTH LIKE '%Mbps'

and nxa.match\_code = 'BOTH'

and nxa.NG\_PATH\_STATUS = 'LIVE'

)

,

PATHS\_W\_CSR as

(

select cpi.trail\_id

from NXA\_PATHS cpi

join ng\_topology.trail\_element cpe

on cpi.TRAIL\_ID = cpe.TRAIL\_ID

join NG\_REPORTS.PORT p

on CPE.ELEMENT\_REF\_ID = P.PORT\_REFERENCE\_ID AND CPE.ELEMENT\_TYPE='E'

join ng\_reports.equipment ei

on p.EQP\_REFERENCE\_ID = ei.EQP\_REFERENCE\_ID

and ei.eqp\_type = 'CSR'

)

select NXAP.TRAIL\_ID

from NXA\_PATHS nxap

MINUS

select pc.TRAIL\_ID

from PATHS\_W\_CSR pc;

sql\_stmt varchar2(32000);

BEGIN

sql\_stmt:='DELETE FROM NORTEL\_NE\_VS\_NG\_AUDIT WHERE ne\_span\_type = ''VOICE\_SPAN''';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in NgVsNortel();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

end;

commit;

sql\_stmt:=

'INSERT INTO ng\_reports.NORTEL\_NE\_VS\_NG\_AUDIT

(vsm\_device\_name\_bsm, ne\_dcg\_number, termination\_type,

ne\_span\_number, switch\_id, vsm\_device\_name\_mtx, trail\_name,

trail\_id, ng\_dcg\_number, ng\_span\_number,

ng\_path\_status, trail\_type,ng\_bandwidth, match\_code, match\_status,

ne\_span\_type, extract\_date, audit\_date)

WITH bsm\_ether\_terminations AS

(SELECT bei.\*,

SUBSTR (bmm.vsm\_device\_name\_mtx, 1, 6)

|| SUBSTR (bmm.vsm\_device\_name\_mtx, 11, 2) switch\_id,

''Ethernet'' termination\_type, bmm.vsm\_device\_name\_mtx,

bei.ROWID ne\_rowid-----ROWID is not there in bsm\_ethernet\_inv

FROM bsm\_ethernet\_inv bei JOIN ng\_reports.bsm\_mtx\_map bmm

ON bei.vsm\_device\_name\_bsm = bmm.vsm\_device\_name\_bsm

),

bsm\_t1\_terminations AS

(SELECT bdsm.\*,

SUBSTR (bmm.vsm\_device\_name\_mtx, 1, 6)

|| SUBSTR (bmm.vsm\_device\_name\_mtx, 11, 2) switch\_id,

''T1'' termination\_type, bmm.vsm\_device\_name\_mtx,

bdsm.ROWID ne\_rowid-----ROWID is not there in bsm\_dcg\_span\_voice\_map

FROM bsm\_dcg\_span\_voice\_map bdsm JOIN ng\_reports.bsm\_mtx\_map bmm

ON bdsm.vsm\_device\_name\_bsm = bmm.vsm\_device\_name\_bsm

),

bsm\_ether\_audit AS

(SELECT bet.vsm\_device\_name\_bsm, bet.dcg\_number ne\_dcg\_number,

bet.termination\_type, NULL ne\_span\_number, bet.switch\_id,

bet.vsm\_device\_name\_mtx, cpi.trail\_name trail\_name,

cpi.trail\_id, cpi.dcg\_number ng\_dcg\_number,

cpi.span\_number ng\_span\_number, cpi.ng\_path\_status,

cpi.ng\_path\_type trail\_type,cpi.bandwidth,

CASE

WHEN cpi.switch\_id IS NULL

THEN ''NE Only''

ELSE ''BOTH''

END match\_code,

CASE

WHEN cpi.span\_number IS NOT NULL AND UPPER(cpi.bandwidth) LIKE ''%DS1%''

THEN ''Pathname Invalid Format, Path Invalid Bandwidth''

WHEN cpi.span\_number IS NOT NULL AND UPPER(cpi.bandwidth) NOT LIKE ''%DS1%''

THEN ''Pathname Invalid Format''

WHEN cpi.span\_number IS NULL AND UPPER(cpi.bandwidth) LIKE ''%DS1%''

THEN ''Path Invalid Bandwidth''

ELSE parse\_status

END match\_status, ''VOICE\_SPAN'' ne\_span\_type,

bet.extract\_date, cpi.ROWID cpi\_rowid, ne\_rowid

FROM bsm\_ether\_terminations bet

LEFT OUTER JOIN

(SELECT \*

FROM vzw\_nt\_voice\_cell\_paths cp

WHERE cp.dcg\_number IS NOT NULL) cpi

ON bet.switch\_id = cpi.switch\_id

AND bet.dcg\_number = cpi.dcg\_number

),

bsm\_t1\_audit AS

(SELECT btt.vsm\_device\_name\_bsm, btt.dcg\_number ne\_dcg\_number,

btt.termination\_type, btt.span\_number ne\_span\_number,

btt.switch\_id, btt.vsm\_device\_name\_mtx,

cpi.trail\_name trail\_name, cpi.trail\_id,

cpi.dcg\_number ng\_dcg\_number,

cpi.span\_number ng\_span\_number, cpi.ng\_path\_status,

cpi.ng\_path\_type trail\_type,cpi.bandwidth,

CASE

WHEN cpi.switch\_id IS NULL

THEN ''NE Only''

ELSE ''BOTH''

END match\_code,

CASE

WHEN UPPER(cpi.bandwidth) NOT LIKE ''%DS1%''

THEN ''Path Invalid Bandwidth''

ELSE parse\_status

END match\_status, ''VOICE\_SPAN'' ne\_span\_type,

btt.extract\_date, cpi.ROWID cpi\_rowid, ne\_rowid

FROM bsm\_t1\_terminations btt

LEFT OUTER JOIN

(SELECT \*

FROM vzw\_nt\_voice\_cell\_paths cp

WHERE cp.dcg\_number IS NOT NULL

AND cp.span\_number IS NOT NULL) cpi

ON btt.switch\_id = cpi.switch\_id

AND btt.dcg\_number = cpi.dcg\_number

AND btt.span\_number = cpi.span\_number

),

bsm\_ne\_only\_and\_ng AS

(SELECT \*

FROM bsm\_ether\_audit ea

UNION

SELECT \*

FROM bsm\_t1\_audit t1a),

ng\_only\_path\_id AS

(SELECT ROWID cpi\_rowid

FROM vzw\_nt\_voice\_cell\_paths cpi

MINUS

SELECT cpi\_rowid

FROM bsm\_ne\_only\_and\_ng),

bsm\_ng\_only AS

(SELECT NULL vsm\_device\_name\_bsm, NULL ne\_dcg\_number,

CASE

WHEN cpi.bandwidth = ''DS1''

THEN ''T1''

ELSE ''Ethernet''

END termination\_type,

NULL ne\_span\_number, cpi.switch\_id, bmm.vsm\_device\_name\_mtx,

cpi.trail\_name trail\_name, cpi.trail\_id,

cpi.dcg\_number ng\_dcg\_number,

cpi.span\_number ng\_span\_number, cpi.ng\_path\_status,

cpi.ng\_path\_type trail\_type, cpi.bandwidth,''NG Only'' match\_code,

parse\_status match\_status, ''VOICE\_SPAN'' ne\_span\_type,

NULL extract\_date, cpi.ROWID cpi\_row\_id, NULL ne\_rowid

FROM vzw\_nt\_voice\_cell\_paths cpi,

ng\_reports.bsm\_mtx\_map bmm,

ng\_only\_path\_id xopi

WHERE cpi.switch\_id =

( SUBSTR (bmm.vsm\_device\_name\_mtx, 1, 6)

|| SUBSTR (bmm.vsm\_device\_name\_mtx, 11, 2)

)

AND cpi.ROWID = xopi.cpi\_rowid),

full\_audit AS

(SELECT \*

FROM bsm\_ne\_only\_and\_ng

UNION

SELECT \*

FROM bsm\_ng\_only)

SELECT fa.vsm\_device\_name\_bsm, fa.ne\_dcg\_number, fa.termination\_type,

fa.ne\_span\_number, fa.switch\_id, fa.vsm\_device\_name\_mtx, fa.trail\_name,

fa.trail\_id, fa.ng\_dcg\_number, fa.ng\_span\_number,

fa.ng\_path\_status, fa.trail\_type,fa.bandwidth, fa.match\_code, fa.match\_status,

fa.ne\_span\_type, fa.extract\_date, TRUNC (SYSDATE) AS audit\_date

FROM full\_audit fa'

;

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Nortel CDMA audit table created');

FOR cursorRec IN t1\_cdma\_multiples\_ng LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit

SET match\_status = nvl2(match\_status,match\_status||',Multiple spans found in NG same tokens','Multiple spans found in NG for same tokens')

WHERE rowid = cursorRec.rd;

END LOOP;

COMMIT;

FOR cursorRec IN ethernet\_cdma\_multiples\_ng LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit

SET match\_status =nvl2(match\_status,match\_status||',Multiple spans found in NG for same tokens','Multiple spans found in NG for same tokens')

WHERE rowid = cursorRec.rd

AND termination\_type='Ethernet' ;

END LOOP;

COMMIT;

FOR cursorRec IN t1\_dup LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

SET match\_status = nvl2(match\_status,match\_status||',Multiple BSM with the Same Tokens','Multiple BSM with the Same Tokens'), match\_code='NE Only'

WHERE nnvxa.vsm\_device\_name\_bsm=cursorRec.vsm\_device\_name\_bsm

AND nnvxa.ne\_dcg\_number=cursorRec.ne\_dcg\_number

AND nnvxa.ne\_span\_number=cursorRec.ne\_span\_number

AND (nnvxa.match\_code ='BOTH' OR nnvxa.ng\_dcg\_number IS NULL)

AND nnvxa.termination\_type='T1'

AND nnvxa.ne\_span\_type='VOICE\_SPAN';

END LOOP;

COMMIT;

FOR cursorRec IN ebh\_dup LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

SET match\_status = nvl2(match\_status,match\_status||',Multiple BSM with the Same Tokens','Multiple BSM with the Same Tokens'),match\_code='NE Only'

WHERE nnvxa.vsm\_device\_name\_bsm=cursorRec.vsm\_device\_name\_bsm

AND nnvxa.ne\_dcg\_number=cursorRec.ne\_dcg\_number

AND nnvxa.termination\_type='Ethernet'

AND (nnvxa.match\_code ='BOTH' OR nnvxa.ng\_dcg\_number IS NULL)

AND nnvxa.ne\_span\_type='VOICE\_SPAN';

END LOOP;

COMMIT;

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit

SET match\_code = 'NE Only'

WHERE match\_code = 'BOTH' AND match\_status IS NOT NULL and ne\_span\_type='VOICE\_SPAN';

COMMIT;

FOR cursorRec IN t1\_nl\_match LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

SET match\_status = nvl2(match\_status, 'Live Path Exists,'||match\_status,'Live Path Exists')

WHERE nnvxa.vsm\_device\_name\_bsm=cursorRec.vsm\_device\_name\_bsm

AND nnvxa.ne\_dcg\_number=cursorRec.ne\_dcg\_number

AND nnvxa.ne\_span\_number=cursorRec.ne\_span\_number

AND nnvxa.ng\_path\_status <> 'LIVE'

AND nnvxa.termination\_type='T1'

AND nnvxa.ne\_span\_type='VOICE\_SPAN';

END LOOP;

COMMIT;

FOR cursorRec IN ebh\_nl\_match LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

SET match\_status = nvl2(match\_status, 'Live Path Exists,'||match\_status,'Live Path Exists')

WHERE nnvxa.vsm\_device\_name\_bsm=cursorRec.vsm\_device\_name\_bsm

AND nnvxa.ne\_dcg\_number=cursorRec.ne\_dcg\_number

AND nnvxa.termination\_type='Ethernet'

AND nnvxa.ng\_path\_status <> 'LIVE'

AND nnvxa.ne\_span\_type='VOICE\_SPAN';

END LOOP;

COMMIT;

FOR cursorRec IN update\_bsm\_with\_unq\_mtx LOOP

update ng\_reports.nortel\_ne\_vs\_ng\_audit nxa set

vsm\_device\_name\_bsm = cursorRec.vsm\_device\_name\_bsm

where vsm\_device\_name\_bsm is null

and vsm\_device\_name\_mtx = cursorRec.vsm\_device\_name\_mtx

and nxa.MATCH\_CODE = 'NG Only';

END LOOP;

COMMIT;

FOR cursorRec IN update\_nt\_cdma\_wrk\_no\_csr\_port LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nxa

SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS =

CASE

when MATCH\_STATUS = 'BOTH/IP Match'

then 'No CSR port on the e-pipe'

else

nvl2(match\_status, 'No CSR port on the e-pipe, '||match\_status, 'No CSR port on the e-pipe')

end

WHERE

NXA.TRAIL\_ID = cursorRec.TRAIL\_ID

;

END LOOP;

commit;

dbms\_output.put\_line('Nortel audit completd successfully');

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in NgVsNortel(): '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

end nt\_cdma\_audit;

PROCEDURE nt\_evdo\_audit

IS

-- One Live and N non Live match

CURSOR t1\_nl\_match IS

SELECT ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number,ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit

WHERE match\_code='BOTH' AND ng\_path\_status='LIVE'

AND ne\_status='Up'

AND ne\_span\_type = 'DATA\_SPAN'

AND termination\_type = 'T1';

-- One Live and N non Live match

CURSOR ebh\_nl\_match IS

SELECT ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit

WHERE match\_code='BOTH' AND ng\_path\_status='LIVE' AND ne\_status='Up'

AND ne\_span\_type = 'DATA\_SPAN'

AND termination\_type = 'Ethernet';

-- if both and t1 and ebh are turned up, then flag as NE Only and change the match\_status

cursor t1\_ebh\_both\_up is

WITH

DUPS as

(

select distinct ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number, ne\_status

from ng\_reports.nortel\_ne\_vs\_ng\_audit src

where

src.termination\_type='Ethernet'

and src.ne\_span\_type='DATA\_SPAN'

and SRC.NE\_STATUS='Up'

)

select distinct tgt.ne\_dom\_ip,tgt.ne\_dcg\_number,tgt.ne\_slot\_number -- tgt.rowid rd

from ng\_reports.nortel\_ne\_vs\_ng\_audit tgt

join DUPS dup

on dup.ne\_dom\_ip = tgt.ne\_dom\_ip

AND dup.ne\_dcg\_number = tgt.ne\_dcg\_number

AND dup.ne\_slot\_number = tgt.ne\_slot\_number

and dup.ne\_status=tgt.ne\_status

AND tgt.termination\_type='T1'

AND tgt.ne\_span\_type='DATA\_SPAN'

and tgt.ne\_status='Up'

;

cursor t1\_evdo\_multiples is

WITH

DUPS as

(

select distinct ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number,ne\_span\_number,

count(case when ng\_path\_status = 'LIVE' then 1 else null end) over (Partition by ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number,ne\_span\_number) as live\_count,

count(case when ng\_path\_status != 'LIVE' then 1 else null end) over (Partition by ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number,ne\_span\_number) as non\_live\_count

from ng\_reports.nortel\_ne\_vs\_ng\_audit src

where src.match\_code='BOTH'

and src.termination\_type='T1'

and src.ne\_span\_type='DATA\_SPAN'

)

select tgt.rowid rd

from ng\_reports.nortel\_ne\_vs\_ng\_audit tgt

join DUPS dup

on dup.ne\_dom\_ip = tgt.ne\_dom\_ip

AND dup.ne\_dcg\_number = tgt.ne\_dcg\_number

AND dup.ne\_slot\_number = tgt.ne\_slot\_number

AND dup.ne\_span\_number = tgt.ne\_span\_number

AND (dup.live\_count > 1 OR (dup.live\_count = 0 AND dup.non\_live\_count > 1)

AND tgt.termination\_type='T1'

AND tgt.match\_code='BOTH'

AND tgt.ne\_span\_type='DATA\_SPAN'

);

cursor ethernet\_evdo\_multiples is

WITH

DUPS as

(

select distinct ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number,

count(case when ng\_path\_status = 'LIVE' then 1 else null end) over (Partition by ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number) as live\_count,

count(case when ng\_path\_status != 'LIVE' then 1 else null end) over (Partition by ne\_dom\_ip,ne\_dcg\_number,ne\_slot\_number) as non\_live\_count

from ng\_reports.nortel\_ne\_vs\_ng\_audit src

where src.match\_code='BOTH'

and src.termination\_type='Ethernet'

and src.ne\_span\_type='DATA\_SPAN'

)

select tgt.rowid rd

from ng\_reports.nortel\_ne\_vs\_ng\_audit tgt

join DUPS dup

on dup.ne\_dom\_ip = tgt.ne\_dom\_ip

AND dup.ne\_dcg\_number = tgt.ne\_dcg\_number

AND dup.ne\_slot\_number = tgt.ne\_slot\_number

AND (dup.live\_count > 1 OR (dup.live\_count = 0 AND dup.non\_live\_count > 1)

AND tgt.termination\_type='Ethernet'

AND tgt.match\_code='BOTH'

AND tgt.ne\_span\_type='DATA\_SPAN'

);

cursor update\_nt\_evdo\_wrk\_no\_csr\_port is

WITH

NXA\_PATHS as

(

select nxa.TRAIL\_ID

from ng\_reports.nortel\_ne\_vs\_ng\_audit nxa

where nxa.TERMINATION\_TYPE = 'Ethernet'

AND nxa.NG\_BANDWIDTH LIKE '%Mbps'

and nxa.match\_code = 'BOTH'

and nxa.NG\_PATH\_STATUS = 'LIVE'

)

,

PATHS\_W\_CSR as

(

select cpi.trail\_id

from NXA\_PATHS cpi

join ng\_topology.trail\_element cpe

on cpi.TRAIL\_ID = cpe.TRAIL\_ID

join NG\_REPORTS.PORT p

on CPE.ELEMENT\_REF\_ID = P.PORT\_REFERENCE\_ID AND CPE.ELEMENT\_TYPE='E'

join ng\_reports.equipment ei

on p.EQP\_REFERENCE\_ID = ei.EQP\_REFERENCE\_ID

and ei.eqp\_type = 'CSR'

)

select NXAP.TRAIL\_ID

from NXA\_PATHS nxap

MINUS

select pc.TRAIL\_ID

from PATHS\_W\_CSR pc;

sql\_stmt varchar2(32000);

BEGIN

sql\_stmt:='DELETE FROM NORTEL\_NE\_VS\_NG\_AUDIT WHERE ne\_span\_type = ''DATA\_SPAN''';

begin

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sql\_stmt);

dbms\_output.put\_line(SubStr('Error in nt\_evdo\_audit();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

end;

commit;

sql\_stmt:= 'INSERT INTO NORTEL\_NE\_VS\_NG\_AUDIT

(vsm\_device\_name\_bsm, vsm\_device\_name\_ems, ne\_dcg\_number,

ne\_slot\_number, ne\_span\_number, ne\_dom\_ip, ne\_status,

termination\_type, switch\_id, trail\_name, trail\_id,

ng\_dcg\_number, ng\_slot\_number, ng\_span\_number, ng\_dom\_ip,

ng\_path\_status, trail\_type,ng\_bandwidth, match\_code, match\_status,

ne\_span\_type,extract\_date,audit\_date)

WITH t1\_ne\_spans AS

(SELECT bdi.ems\_name vsm\_device\_name\_ems, bdi.vsm\_device\_name\_bsm,

bdi.dcg\_number, bdi.slot\_number, epi.port\_num span\_number,

bdi.dom\_ip, epi.status, ''T1'' port\_type , epi.extract\_date,

epi.ROWID ne\_rowid

FROM nortel\_dom\_inventory bdi JOIN nortel\_port\_inventory epi

ON bdi.seq\_num = epi.dom\_seq\_num AND epi.port\_type = ''T1E1''

),

ether\_ne\_spans AS

(

-- if any of the ethernet ports are hot then it is an ebh site

SELECT DISTINCT bdi.ems\_name vsm\_device\_name\_ems,

bdi.vsm\_device\_name\_bsm, bdi.dcg\_number,

bdi.slot\_number, NULL span\_number, bdi.dom\_ip,

epi.status, port\_type, epi.extract\_date,

bdi.ROWID ne\_rowid

FROM nortel\_dom\_inventory bdi JOIN nortel\_port\_inventory epi

ON bdi.seq\_num = epi.dom\_seq\_num

AND epi.port\_type = ''Ethernet''

AND primary\_ip <> ''0.0.0.0''

AND epi.status = ''Up''

),

ether\_audit AS

(SELECT ne.vsm\_device\_name\_bsm, ne.vsm\_device\_name\_ems,

ne.dcg\_number ne\_dcg\_number, ne.slot\_number ne\_slot\_number,

ne.span\_number ne\_span\_number, ne.dom\_ip ne\_dom\_ip,

ne.status ne\_status, ne.port\_type termination\_type,

NVL (vcpi.switch\_id, ''UNKNOWN'') switch\_id, vcpi.trail\_name,

vcpi.trail\_id, vcpi.dcg\_number ng\_dcg\_number,

vcpi.slot\_number ng\_slot\_number,

vcpi.span\_number ng\_span\_number, vcpi.dom\_node\_ip ng\_dom\_ip,

vcpi.ng\_path\_status, vcpi.ng\_path\_type,vcpi.bandwidth,

CASE

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

and vcpi.SPAN\_NUMBER is null

and (upper(vcpi.BANDWIDTH) like ''%MBPS''

or upper(vcpi.BANDWIDTH) = ''AGGREGATE'')

THEN ''BOTH''

ELSE ''NE Only''

END match\_code,

CASE

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

and vcpi.SPAN\_NUMBER is null

and (upper(vcpi.BANDWIDTH) like ''%MBPS''

or upper(vcpi.BANDWIDTH) = ''AGGREGATE'')

THEN ''BOTH/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

and vcpi.SPAN\_NUMBER is null

and upper(vcpi.BANDWIDTH) not like ''%MBPS''

and upper(vcpi.BANDWIDTH) <> ''AGGREGATE''

THEN ''Path Invalid Bandwidth/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

and vcpi.SPAN\_NUMBER is not null

and upper(vcpi.BANDWIDTH) not like ''%MBPS''

and upper(vcpi.BANDWIDTH) <> ''AGGREGATE''

THEN ''Pathname Invalid Format, Path Invalid Bandwidth/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

and vcpi.SPAN\_NUMBER is not null

and (upper(vcpi.BANDWIDTH) like ''%MBPS''

or upper(vcpi.BANDWIDTH) = ''AGGREGATE'')

THEN ''Pathname Invalid Format/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND (vcpi.dcg\_number <> ne.dcg\_number

or vcpi.slot\_number <> ne.slot\_number)

and vcpi.SPAN\_NUMBER is not null

and (upper(vcpi.BANDWIDTH) like ''%MBPS''

or upper(vcpi.BANDWIDTH) = ''AGGREGATE'')

THEN ''Pathname Token Mismatch, Pathname Invalid Format/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND (vcpi.dcg\_number <> ne.dcg\_number

or vcpi.slot\_number <> ne.slot\_number)

and vcpi.SPAN\_NUMBER is not null

and upper(vcpi.BANDWIDTH) not like ''%MBPS''

and upper(vcpi.BANDWIDTH) <> ''AGGREGATE''

THEN ''Pathname Token Mismatch, Pathname Invalid Format, Path Invalid Bandwidth/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND (vcpi.dcg\_number <> ne.dcg\_number

or vcpi.slot\_number <> ne.slot\_number)

and vcpi.SPAN\_NUMBER is null

and upper(vcpi.BANDWIDTH) not like ''%MBPS''

and upper(vcpi.BANDWIDTH) <> ''AGGREGATE''

THEN ''Pathname Token Mismatch, Path Invalid Bandwidth/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND (vcpi.dcg\_number <> ne.dcg\_number

or vcpi.slot\_number <> ne.slot\_number)

and vcpi.SPAN\_NUMBER is null

and (upper(vcpi.BANDWIDTH) like ''%MBPS''

or upper(vcpi.BANDWIDTH) = ''AGGREGATE'')

THEN ''Pathname Token Mismatch/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number is null

and (upper(vcpi.BANDWIDTH) like ''%MBPS''

or upper(vcpi.BANDWIDTH) = ''AGGREGATE'')

THEN ''Pathname Invalid Format/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number is null

and upper(vcpi.BANDWIDTH) not like ''%MBPS''

and upper(vcpi.BANDWIDTH) <> ''AGGREGATE''

THEN ''Pathname Invalid Format, Path Invalid Bandwidth/IP Match''

WHEN parse\_status IS NOT NULL

THEN ''??????'' || parse\_status || ''/IP Match''

END match\_status,

''DATA\_SPAN'' ne\_span\_type, ne.extract\_date,

vcpi.ROWID cpi\_rowid, ne\_rowid

FROM ether\_ne\_spans ne LEFT OUTER JOIN vzw\_nt\_evdo\_cell\_paths vcpi

ON ne.dom\_ip = vcpi.dom\_node\_ip

),

t1\_audit AS

(SELECT ne.vsm\_device\_name\_bsm, ne.vsm\_device\_name\_ems,

ne.dcg\_number ne\_dcg\_number, ne.slot\_number ne\_slot\_number,

ne.span\_number ne\_span\_number, ne.dom\_ip ne\_dom\_ip,

ne.status ne\_status, ne.port\_type termination\_type,

NVL (vcpi.switch\_id, ''UNKNOWN'') switch\_id, vcpi.trail\_name,

vcpi.trail\_id, vcpi.dcg\_number ng\_dcg\_number,

vcpi.slot\_number ng\_slot\_number,

vcpi.span\_number ng\_span\_number, vcpi.dom\_node\_ip ng\_dom\_ip,

vcpi.ng\_path\_status, vcpi.ng\_path\_type,vcpi.bandwidth,

CASE

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

AND vcpi.span\_number = ne.span\_number

AND vcpi.BANDWIDTH = ''DS1''

THEN ''BOTH''

ELSE ''NE Only''

END match\_code,

CASE

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

AND vcpi.span\_number = ne.span\_number

AND vcpi.BANDWIDTH = ''DS1''

THEN ''BOTH/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

AND vcpi.span\_number = ne.span\_number

AND vcpi.BANDWIDTH <> ''DS1''

THEN ''Path Invalid Bandwidth/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

AND vcpi.span\_number is null

AND vcpi.BANDWIDTH <> ''DS1''

THEN ''Pathname Invalid Format, Path Invalid Bandwidth/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND vcpi.dcg\_number = ne.dcg\_number

AND vcpi.slot\_number = ne.slot\_number

AND vcpi.span\_number is null

AND vcpi.BANDWIDTH = ''DS1''

THEN ''Pathname Invalid Format/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND (vcpi.dcg\_number <> ne.dcg\_number

or vcpi.slot\_number <> ne.slot\_number

or vcpi.span\_number <> ne.span\_number)

AND vcpi.BANDWIDTH = ''DS1''

THEN ''Pathname Token Mismatch/IP Match''

WHEN vcpi.dom\_node\_ip IS NOT NULL

AND ne.dom\_ip IS NOT NULL

AND (vcpi.dcg\_number <> ne.dcg\_number

or vcpi.slot\_number <> ne.slot\_number

or vcpi.span\_number <> ne.span\_number)

AND vcpi.BANDWIDTH <> ''DS1''

THEN ''Pathname Token Mismatch, Path Invalid Bandwidth/IP Match''

WHEN parse\_status IS NOT NULL

THEN ''?????'' || parse\_status ||''/IP Match''

END match\_status,

''DATA\_SPAN'' ne\_span\_type, ne.extract\_date,

vcpi.ROWID cpi\_rowid, ne\_rowid

FROM t1\_ne\_spans ne LEFT OUTER JOIN vzw\_nt\_evdo\_cell\_paths vcpi

ON ne.dom\_ip = vcpi.dom\_node\_ip

AND ne.span\_number = vcpi.span\_number

),

t1\_ether\_audit AS

(SELECT ne.vsm\_device\_name\_bsm, ne.vsm\_device\_name\_ems, ne\_dcg\_number,

ne\_slot\_number, ne\_span\_number, ne\_dom\_ip, ne\_status,

termination\_type, switch\_id, trail\_name, trail\_id,

ng\_dcg\_number, ng\_slot\_number, ng\_span\_number, ng\_dom\_ip,

ng\_path\_status, ng\_path\_type,bandwidth,

match\_code, match\_status,

ne\_span\_type, ne.extract\_date, cpi\_rowid, ne\_rowid

FROM t1\_audit ne

UNION

SELECT ne.vsm\_device\_name\_bsm, ne.vsm\_device\_name\_ems, ne\_dcg\_number,

ne\_slot\_number, ne\_span\_number, ne\_dom\_ip, ne\_status,

termination\_type, switch\_id, trail\_name, trail\_id,

ng\_dcg\_number, ng\_slot\_number, ng\_span\_number, ng\_dom\_ip,

ng\_path\_status, ng\_path\_type, bandwidth,match\_code, match\_status,

ne\_span\_type, ne.extract\_date, cpi\_rowid, ne\_rowid

FROM ether\_audit ne),

ng\_only\_paths\_id AS

(SELECT ROWID cpi\_rowid

FROM vzw\_nt\_evdo\_cell\_paths cpi

WHERE cpi.dom\_node\_ip IS NOT NULL

MINUS

SELECT cpi\_rowid

FROM t1\_ether\_audit ea),

ng\_only AS

(SELECT NULL vsm\_device\_name\_bsm, NULL vsm\_device\_name\_ems,

NULL ne\_dcg\_number, NULL ne\_slot\_number, NULL ne\_span\_number,

NULL ne\_dom\_ip, NULL ne\_status,

CASE

WHEN cpi.bandwidth = ''DS1''

THEN ''T1''

ELSE ''Ethernet''

END termination\_type,

NVL (cpi.switch\_id, ''UNKNOWN'') switch\_id, cpi.trail\_name,

cpi.trail\_id, cpi.dcg\_number ng\_dcg\_number,

cpi.slot\_number ng\_slot\_number,

cpi.span\_number ng\_span\_number, cpi.dom\_node\_ip ng\_dom\_ip,

cpi.ng\_path\_status, cpi.ng\_path\_type,cpi.bandwidth, ''NG Only'' match\_code,

CASE

WHEN parse\_status IS NULL

THEN ''Invalid IP''

ELSE parse\_status || ''/Invalid IP''

END match\_status,

''DATA\_SPAN'' ne\_span\_type, NULL extract\_date,

cpi.ROWID cpi\_rowid, NULL ne\_rowid

FROM vzw\_nt\_evdo\_cell\_paths cpi JOIN ng\_only\_paths\_id xngo

ON xngo.cpi\_rowid = cpi.ROWID

),

full\_audit AS

(SELECT \*

FROM t1\_ether\_audit

UNION

SELECT \*

FROM ng\_only)

SELECT fa.vsm\_device\_name\_bsm, fa.vsm\_device\_name\_ems,fa.ne\_dcg\_number,

fa.ne\_slot\_number, fa.ne\_span\_number, fa.ne\_dom\_ip, fa.ne\_status,

fa.termination\_type, fa.switch\_id, fa.trail\_name, fa.trail\_id,

fa.ng\_dcg\_number, fa.ng\_slot\_number, fa.ng\_span\_number, fa.ng\_dom\_ip,

fa.ng\_path\_status, fa.ng\_path\_type,fa.bandwidth,fa.match\_code, fa.match\_status,

fa.ne\_span\_type, fa.extract\_date,TRUNC (SYSDATE) AS audit\_date

FROM full\_audit fa';

EXECUTE IMMEDIATE sql\_stmt;

commit;

dbms\_output.put\_line('Nortel EVDO audit completed');

FOR cursorRec IN t1\_evdo\_multiples LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit

set match\_status = nvl2(match\_status,match\_status||',Multiple spans found in Ng for same tokens','Multiple spans found in Ng for same tokens') , match\_code='NE Only'

WHERE rowid = cursorRec.rd;

END LOOP;

commit;

FOR cursorRec IN ethernet\_evdo\_multiples LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit

set match\_status = nvl2(match\_status,match\_status||',Multiple spans found in Ng for same tokens','Multiple spans found in Ng for same tokens') , match\_code='NE Only'

WHERE rowid = cursorRec.rd;

END LOOP;

commit;

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit

SET match\_code = 'NE Only'

WHERE match\_code = 'BOTH' AND match\_status <>'BOTH/IP Match' and ne\_span\_type='DATA\_SPAN';

COMMIT;

FOR cursorRec IN t1\_nl\_match LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

SET match\_status = nvl2(match\_status, 'Live Path Exists,'||match\_status,'Live Path Exists')

WHERE nnvxa.ne\_dom\_ip=cursorRec.ne\_dom\_ip

AND nnvxa.ne\_dcg\_number=cursorRec.ne\_dcg\_number

AND nnvxa.ne\_slot\_number=cursorRec.ne\_slot\_number

AND nnvxa.ne\_span\_number=cursorRec.ne\_span\_number

AND nnvxa.ng\_path\_status <> 'LIVE' AND ne\_status='Up'

AND nnvxa.termination\_type='T1'

AND nnvxa.ne\_span\_type='DATA\_SPAN';

END LOOP;

COMMIT;

FOR cursorRec IN ebh\_nl\_match LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

SET match\_status = nvl2(match\_status, 'Live Path Exists,'||match\_status,'Live Path Exists')

WHERE nnvxa.ne\_dom\_ip=cursorRec.ne\_dom\_ip

AND nnvxa.ne\_dcg\_number=cursorRec.ne\_dcg\_number

AND nnvxa.ne\_slot\_number=cursorRec.ne\_slot\_number

AND nnvxa.termination\_type='Ethernet'

AND nnvxa.ng\_path\_status <> 'LIVE' AND ne\_status='Up'

AND nnvxa.ne\_span\_type='DATA\_SPAN';

END LOOP;

COMMIT;

FOR cursorRec IN update\_nt\_evdo\_wrk\_no\_csr\_port LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nxa

SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS =

CASE

when MATCH\_STATUS = 'BOTH/IP Match'

then 'No CSR port on the e-pipe'

else

nvl2(match\_status, 'No CSR port on the e-pipe, '||match\_status, 'No CSR port on the e-pipe')

end

WHERE

nxa.TRAIL\_ID = cursorRec.TRAIL\_ID

;

END LOOP;

commit;

FOR cursorRec IN t1\_ebh\_both\_up LOOP

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

SET MATCH\_CODE = 'NE Only',

MATCH\_STATUS =

CASE

when MATCH\_STATUS = 'BOTH/IP Match'

then 'T1 and EBH are both Up'

else

nvl2(match\_status, 'T1 and EBH are both Up, '||match\_status, 'T1 and EBH are both Up')

end

WHERE nnvxa.ne\_dom\_ip=cursorRec.ne\_dom\_ip

AND nnvxa.ne\_dcg\_number=cursorRec.ne\_dcg\_number

AND nnvxa.ne\_slot\_number=cursorRec.ne\_slot\_number

AND nnvxa.ne\_status='Up'

;

END LOOP;

commit;

--update vsm\_device\_name\_mtx when mtx is null

UPDATE ng\_reports.nortel\_ne\_vs\_ng\_audit w

set w.vsm\_device\_name\_mtx=(select m.vsm\_device\_name\_mtx from ng\_reports.bsm\_mtx\_map m where m.vsm\_device\_name\_bsm=w.vsm\_device\_name\_bsm)

where w.vsm\_device\_name\_mtx is null

and w.vsm\_device\_name\_ems is not null

;

commit;

EXCEPTION

WHEN OTHERS THEN

RAISE;

END nt\_evdo\_audit;

procedure populate\_cdma\_regional\_summary is

cursor updateCompliance is

select summ.rowid rd,summ.\* from ng\_reports.NORTEL\_CDMA\_AUDIT\_REG\_SUMM summ;

sqlStmt VARCHAR2(32767);

sqlStmtT1Comp VARCHAR2(32767);

sqlStmtEbhComp VARCHAR2(32767);

sqlStmtOvlComp VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

t1Reported NUMBER;

ebhReported NUMBER;

BEGIN

sqlStmt := 'truncate table NORTEL\_CDMA\_AUDIT\_REG\_SUMM';

execute immediate sqlStmt;

sqlStmt := 'insert into NORTEL\_CDMA\_AUDIT\_REG\_SUMM

(territory, market\_territory, sub\_market,t1\_discovered,t1\_matched\_live,t1\_matched\_nonlive,ebh\_discovered,ebh\_matched\_live,ebh\_matched\_nonlive,ebh\_csr\_matched\_live,ebh\_csr\_matched\_non\_live

)

WITH mtx\_region\_map AS

(SELECT DISTINCT cdmp.territory, cdmp.market\_territory, cdmp.sub\_market, m.vsm\_device\_name\_mtx,

( SUBSTR (vsm\_device\_name\_mtx, 1, 6)

|| SUBSTR (vsm\_device\_name\_mtx, 11, 2)

) switch\_id

FROM ng\_reports.clli\_domain\_map\_v cdmp

LEFT OUTER JOIN

(SELECT \*

FROM ng\_reports.bsm\_mtx\_map bmm

WHERE bmm.vsm\_device\_name\_bsm LIKE ''%BSM%''

AND bmm.mtx\_status = ''Live'') m

ON SUBSTR (m.vsm\_device\_name\_mtx, 1, 6) =

SUBSTR (cdmp.clli, 1, 6)

WHERE cdmp.territory <> ''NNO''

),

t1\_discovered AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) t1\_discovered

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,nnvxa.vsm\_device\_name\_bsm,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.match\_code IN (''BOTH'', ''NE Only''))

GROUP BY vsm\_device\_name\_mtx),

ethernet\_discovered AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_discovered

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,nnvxa.vsm\_device\_name\_bsm,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.match\_code IN (''BOTH'', ''NE Only''))

GROUP BY vsm\_device\_name\_mtx),

t1\_matched\_live AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) t1\_matched\_live

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH'')

GROUP BY vsm\_device\_name\_mtx),

t1\_matched\_nonlive AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) t1\_matched\_nonlive

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status <> ''LIVE''

AND nnvxa.match\_code = ''BOTH''

MINUS

SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH'')

GROUP BY vsm\_device\_name\_mtx),

ethernet\_matched\_live AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_matched\_live

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH''

)

GROUP BY vsm\_device\_name\_mtx),

ethernet\_matched\_nonlive AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_matched\_nonlive

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status <> ''LIVE''

AND nnvxa.match\_code = ''BOTH''

MINUS

SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH''

)

GROUP BY vsm\_device\_name\_mtx)

,

ethernet\_csr\_matched\_live as

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_csr\_matched\_live

FROM ( SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx, nnvxa.ne\_dcg\_number, NNVXA.NG\_BANDWIDTH

FROM mtx\_region\_map mrp, ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

join ng\_topology.trail cpi on NNVXA.TRAIL\_ID = CPI.TRAIL\_ID

join ng\_reports.PORT p on cpi.TRAIL\_ID = P.CKT\_PATH\_REFERENCE\_ID

join NG\_REPORTS.EQUIPMENT ei on EI.EQP\_REFERENCE\_ID = P.EQP\_REFERENCE\_ID

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx = mrp.vsm\_device\_name\_mtx

and NNVXA.NG\_BANDWIDTH like ''%Mbps''

and EI.EQP\_TYPE = ''CSR'' )

GROUP BY vsm\_device\_name\_mtx)

,

ethernet\_csr\_matched\_non\_live as

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_csr\_matched\_non\_live

FROM ( SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx, nnvxa.ne\_dcg\_number, NNVXA.NG\_BANDWIDTH

FROM mtx\_region\_map mrp, ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

join ng\_topology.trail cpi ON nnvxa.trail\_id = cpi.trail\_id

join ng\_reports.port p ON cpi.trail\_id = p.ckt\_path\_reference\_id

join ng\_reports.equipment ei ON ei.eqp\_reference\_id = p.eqp\_reference\_id

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status <> ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx = mrp.vsm\_device\_name\_mtx

and NNVXA.NG\_BANDWIDTH like ''%Mbps''

and EI.EQP\_TYPE = ''CSR'' )

GROUP BY vsm\_device\_name\_mtx)

SELECT mrm.territory, mrm.market\_territory, mrm.sub\_market,

SUM (NVL (t1\_discovered, 0)) AS t1\_discovered,

SUM (NVL (t1\_matched\_live, 0)) AS t1\_matched\_live,

SUM (NVL (t1\_matched\_nonlive, 0)) AS t1\_matched\_nonlive,

SUM (NVL (ethernet\_discovered, 0)) AS ethernet\_discovered,

SUM (NVL (ethernet\_matched\_live, 0)) AS ethernet\_matched\_live,

SUM (NVL (ethernet\_matched\_nonlive, 0)) AS ethernet\_matched\_nonlive,

SUM (NVL (ethernet\_csr\_matched\_live, 0)) AS ethernet\_csr\_matched\_live,

SUM (NVL (ethernet\_csr\_matched\_non\_live, 0)) AS ethernet\_csr\_matched\_non\_live

FROM mtx\_region\_map mrm LEFT OUTER JOIN t1\_discovered td

ON mrm.vsm\_device\_name\_mtx = td.vsm\_device\_name\_mtx

LEFT OUTER JOIN t1\_matched\_live tm

ON mrm.vsm\_device\_name\_mtx = tm.vsm\_device\_name\_mtx

LEFT OUTER JOIN t1\_matched\_nonlive tum

ON mrm.vsm\_device\_name\_mtx = tum.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_discovered ed

ON mrm.vsm\_device\_name\_mtx = ed.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_matched\_live em

ON mrm.vsm\_device\_name\_mtx = em.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_matched\_nonlive eum

ON mrm.vsm\_device\_name\_mtx = eum.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_csr\_matched\_live emcsr

ON mrm.vsm\_device\_name\_mtx = emcsr.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_csr\_matched\_non\_live eumcsr

ON mrm.vsm\_device\_name\_mtx = eumcsr.vsm\_device\_name\_mtx

GROUP BY ROLLUP (mrm.territory, mrm.market\_territory, mrm.sub\_market)

ORDER BY mrm.territory, mrm.market\_territory, mrm.sub\_market'

;

execute immediate sqlStmt;

commit;

sqlStmtT1Comp := 'update NORTEL\_CDMA\_AUDIT\_REG\_SUMM set t1\_comp =:x ,t1\_reported =:y where rowid = :a';

sqlStmtEbhComp := 'update NORTEL\_CDMA\_AUDIT\_REG\_SUMM set ebh\_comp =:x,ebh\_reported =:y where rowid =:a';

sqlStmtOvlComp := 'update NORTEL\_CDMA\_AUDIT\_REG\_SUMM set overall\_comp =:x where rowid =:a';

for cur in updateCompliance

loop

t1Reported := nvl(cur.t1\_discovered,0) - nvl(cur.t1\_matched\_nonlive,0);

ebhReported := nvl(cur.ebh\_discovered,0) - nvl(cur.ebh\_csr\_matched\_non\_live,0);

overallComp := nvl(cur.t1\_discovered,0) + nvl(cur.ebh\_discovered,0) ;

overallComp := overallComp - nvl(cur.t1\_matched\_nonlive,0) - nvl(cur.ebh\_csr\_matched\_non\_live,0);

if(t1Reported >0) then

t1Comp :=(nvl(cur.t1\_matched\_live,0)/t1Reported) \*100;

t1Comp := round(t1Comp,2);

execute immediate sqlStmtT1Comp using t1Comp,t1Reported,cur.rd;

--dbms\_output.put\_line('Non null '|| t1Comp);

end if;

if(ebhReported>0) then

ebhComp :=(nvl(cur.ebh\_csr\_matched\_live,0)/ebhReported) \*100;

ebhComp := round(ebhComp,2);

execute immediate sqlStmtEbhComp using ebhComp,ebhReported,cur.rd;

--dbms\_output.put\_line('good EBH '||ebhComp);

end if;

if(overallComp>0) then

overallComp :=((nvl(cur.t1\_matched\_live,0)+nvl(cur.ebh\_csr\_matched\_live,0))/overallComp) \*100;

overallComp := round(overallComp,2);

execute immediate sqlStmtOvlComp using overallComp,cur.rd;

--dbms\_output.put\_line('good EBH '||overallComp);

end if;

end loop;

commit;

sqlStmt:= 'update NORTEL\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt:= 'update NORTEL\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = MARKET\_TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt:= 'update NORTEL\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is null' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt:= 'update NORTEL\_CDMA\_AUDIT\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB =''Unknown'',wk.sub\_market=''Unknown'' where territory=''Unknown''' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in populate\_cdma\_regional\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END;

procedure populate\_cdma\_device\_summary is

cursor updateCompliance is

select summ.rowid rd,summ.\* from ng\_reports.NORTEL\_CDMA\_AUDIT\_DEV\_SUMM summ;

sqlStmt VARCHAR2(32767);

sqlStmtT1Comp VARCHAR2(32767);

sqlStmtEbhComp VARCHAR2(32767);

sqlStmtOvlComp VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

BEGIN

sqlStmt := 'truncate table ng\_reports.NORTEL\_CDMA\_AUDIT\_DEV\_SUMM';

execute immediate sqlStmt;

sqlStmt := 'insert into ng\_reports.NORTEL\_CDMA\_AUDIT\_DEV\_SUMM

(territory, market\_territory, sub\_market,leaf\_domain,vsm\_device\_name\_mtx,switch\_id,t1\_discovered,t1\_matched\_live,t1\_matched\_nonlive,ebh\_discovered,ebh\_matched\_live,ebh\_matched\_nonlive, ebh\_csr\_matched\_live, ebh\_csr\_matched\_non\_live

)

with mtx\_region\_map AS

(SELECT DISTINCT cdmp.territory,cdmp.market\_territory,cdmp.sub\_market,cdmp.leaf\_domain\_name, m.vsm\_device\_name\_mtx ,

( SUBSTR (vsm\_device\_name\_mtx, 1, 6)

|| SUBSTR (vsm\_device\_name\_mtx, 11, 2)

) switch\_id

FROM ng\_reports.clli\_domain\_map\_v cdmp

RIGHT OUTER JOIN (SELECT \* FROM ng\_reports.bsm\_mtx\_map bmm

WHERE bmm.vsm\_device\_name\_bsm LIKE ''%BSM%''

AND bmm.mtx\_status =''Live'') m

ON substr(m.vsm\_device\_name\_mtx,1,6)=substr(cdmp.clli,1,6)

WHERE cdmp.territory <> ''NNO''

),

t1\_discovered AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) t1\_discovered

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,nnvxa.vsm\_device\_name\_bsm,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.match\_code IN (''BOTH'', ''NE Only'')

AND nnvxa.vsm\_device\_name\_mtx IN

(mrp.vsm\_device\_name\_mtx))

GROUP BY vsm\_device\_name\_mtx),

ethernet\_discovered AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_discovered

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,nnvxa.vsm\_device\_name\_bsm,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.match\_code IN (''BOTH'', ''NE Only'')

AND nnvxa.vsm\_device\_name\_mtx IN

(mrp.vsm\_device\_name\_mtx))

GROUP BY vsm\_device\_name\_mtx),

t1\_matched\_live AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) t1\_matched\_live

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx =

mrp.vsm\_device\_name\_mtx

)

GROUP BY vsm\_device\_name\_mtx),

ethernet\_matched\_live AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_matched\_live

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx =

mrp.vsm\_device\_name\_mtx)

GROUP BY vsm\_device\_name\_mtx),

t1\_matched\_nonlive AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) t1\_matched\_nonlive

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status <> ''LIVE''

AND nnvxa.match\_code =''BOTH''

AND nnvxa.vsm\_device\_name\_mtx =

mrp.vsm\_device\_name\_mtx

MINUS

SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number, nnvxa.ne\_span\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''T1''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code =''BOTH''

AND nnvxa.vsm\_device\_name\_mtx =

mrp.vsm\_device\_name\_mtx)

GROUP BY vsm\_device\_name\_mtx),

ethernet\_matched\_nonlive AS

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_matched\_nonlive

FROM (SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status <> ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx = mrp.vsm\_device\_name\_mtx

MINUS

SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx,

nnvxa.ne\_dcg\_number

FROM ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa,

mtx\_region\_map mrp

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx = mrp.vsm\_device\_name\_mtx)

GROUP BY vsm\_device\_name\_mtx)

,

ethernet\_csr\_matched\_live as

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_csr\_matched\_live

FROM ( SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx, nnvxa.ne\_dcg\_number, NNVXA.NG\_BANDWIDTH

FROM mtx\_region\_map mrp, ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

join ng\_topology.trail cpi ON nnvxa.trail\_id = cpi.trail\_id

join ng\_reports.port p ON cpi.trail\_id = p.ckt\_path\_reference\_id

join ng\_reports.equipment ei ON ei.eqp\_reference\_id = p.eqp\_reference\_id

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status = ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx = mrp.vsm\_device\_name\_mtx

and NNVXA.NG\_BANDWIDTH like ''%Mbps''

and EI.EQP\_TYPE = ''CSR'' )

GROUP BY vsm\_device\_name\_mtx)

,

ethernet\_csr\_matched\_non\_live as

(SELECT vsm\_device\_name\_mtx, COUNT (1) ethernet\_csr\_matched\_non\_live

FROM ( SELECT DISTINCT nnvxa.vsm\_device\_name\_mtx, nnvxa.ne\_dcg\_number, NNVXA.NG\_BANDWIDTH

FROM mtx\_region\_map mrp, ng\_reports.nortel\_ne\_vs\_ng\_audit nnvxa

join ng\_topology.trail cpi ON nnvxa.trail\_id = cpi.trail\_id

join ng\_reports.port p ON cpi.trail\_id = p.ckt\_path\_reference\_id

join ng\_reports.equipment ei ON ei.eqp\_reference\_id = p.eqp\_reference\_id

WHERE nnvxa.termination\_type = ''Ethernet''

AND nnvxa.ne\_span\_type = ''VOICE\_SPAN''

AND nnvxa.ng\_path\_status <> ''LIVE''

AND nnvxa.match\_code = ''BOTH''

AND nnvxa.vsm\_device\_name\_mtx = mrp.vsm\_device\_name\_mtx

and NNVXA.NG\_BANDWIDTH like ''%Mbps''

and EI.EQP\_TYPE = ''CSR'' )

GROUP BY vsm\_device\_name\_mtx)

SELECT mrm.territory, mrm.market\_territory, mrm.sub\_market,mrm.leaf\_domain\_name,mrm.vsm\_device\_name\_mtx, mrm.switch\_id,

NVL (t1\_discovered, 0) AS t1\_discovered,

NVL (t1\_matched\_live, 0) AS t1\_matched\_live,

NVL (t1\_matched\_nonlive, 0) AS t1\_matched\_nonlive,

NVL (ethernet\_discovered, 0) AS ethernet\_discovered,

NVL (ethernet\_matched\_live, 0) AS ethernet\_matched\_live,

NVL (ethernet\_matched\_nonlive, 0) AS ethernet\_matched\_nonlive,

NVL (ethernet\_csr\_matched\_live, 0) AS ethernet\_csr\_matched\_live,

NVL (ethernet\_csr\_matched\_non\_live, 0) AS ethernet\_csr\_matched\_non\_live

FROM mtx\_region\_map mrm LEFT OUTER JOIN t1\_discovered td

ON mrm.vsm\_device\_name\_mtx = td.vsm\_device\_name\_mtx

LEFT OUTER JOIN t1\_matched\_live tm

ON mrm.vsm\_device\_name\_mtx = tm.vsm\_device\_name\_mtx

LEFT OUTER JOIN t1\_matched\_nonlive tum

ON mrm.vsm\_device\_name\_mtx = tum.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_discovered ed

ON mrm.vsm\_device\_name\_mtx = ed.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_matched\_live em

ON mrm.vsm\_device\_name\_mtx = em.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_matched\_nonlive eum

ON mrm.vsm\_device\_name\_mtx = eum.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_csr\_matched\_live emcsr

ON mrm.vsm\_device\_name\_mtx = emcsr.vsm\_device\_name\_mtx

LEFT OUTER JOIN ethernet\_csr\_matched\_non\_live eumcsr

ON mrm.vsm\_device\_name\_mtx = eumcsr.vsm\_device\_name\_mtx

ORDER BY mrm.vsm\_device\_name\_mtx'

;

execute immediate sqlStmt;

commit;

dbms\_output.put\_line('Device Summary created');

sqlStmtT1Comp := 'update ng\_reports.NORTEL\_CDMA\_AUDIT\_DEV\_SUMM set t1\_comp =:x where rowid = :a';

sqlStmtEbhComp := 'update ng\_reports.NORTEL\_CDMA\_AUDIT\_DEV\_SUMM set ebh\_comp =:x where rowid = :a';

sqlStmtOvlComp := 'update ng\_reports.NORTEL\_CDMA\_AUDIT\_DEV\_SUMM set overall\_comp =:x where rowid = :a';

for cur in updateCompliance

loop

t1Comp := nvl(cur.t1\_discovered,0) - nvl(cur.t1\_matched\_nonlive,0);

--ebhComp = nvl(cur.ebh\_discovered,0) - nvl(cur.ebh\_matched\_nonlive,0);

ebhComp := nvl(cur.ebh\_discovered,0) - nvl(cur.ebh\_csr\_matched\_non\_live,0);

overallComp := nvl(cur.t1\_discovered,0) + nvl(cur.ebh\_discovered,0) ;

--overallComp = overallComp - nvl(cur.t1\_matched\_nonlive,0) - nvl(cur.ebh\_matched\_nonlive,0);

overallComp := overallComp - nvl(cur.t1\_matched\_nonlive,0) - nvl(cur.ebh\_csr\_matched\_non\_live,0);

if(t1Comp >0) then

t1Comp :=(nvl(cur.t1\_matched\_live,0)/t1Comp) \*100;

t1Comp := round(t1Comp,2);

execute immediate sqlStmtT1Comp using t1Comp,cur.rd;

--dbms\_output.put\_line('Non null '|| t1Comp);

end if;

if(ebhComp>0) then

ebhComp :=(nvl(cur.ebh\_csr\_matched\_live,0)/ebhComp) \*100;

ebhComp := round(ebhComp,2);

execute immediate sqlStmtEbhComp using ebhComp,cur.rd;

--dbms\_output.put\_line('good EBH '||ebhComp);

end if;

if(overallComp>0) then

overallComp :=((nvl(cur.t1\_matched\_live,0)+nvl(cur.ebh\_csr\_matched\_live,0))/overallComp) \*100;

overallComp := round(overallComp,2);

execute immediate sqlStmtOvlComp using overallComp,cur.rd;

--dbms\_output.put\_line('good EBH '||overallComp);

end if;

end loop;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in populate\_cdma\_device\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END;

procedure populate\_evdo\_regional\_summary is

cursor updateCompliance is

select summ.rowid rd,summ.\* from ng\_reports.NT\_EVDO\_BTS\_TERM\_REG\_SUMM summ;

cursor totUpEbhSpans is

WITH

-- Total EBH spans

LIVE\_EBH\_PORTS as

(

select distinct npi.DOM\_SEQ\_NUM

from ng\_reports.NORTEL\_PORT\_INVENTORY npi

where npi.PORT\_TYPE = 'Ethernet'

and npi.STATUS = 'Up'

)

,

TOTAL\_LIVE\_EBH\_SPANS as

(

select cdm.leaf\_domain\_id,

ndi.\*

from ng\_reports.NORTEL\_DOM\_INVENTORY ndi

join LIVE\_EBH\_PORTS npi

on ndi.SEQ\_NUM = npi.DOM\_SEQ\_NUM

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from ng\_reports.clli\_domain\_map )cdm

on cdm.clli = substr(ndi.EMS\_NAME, 1, 6)

)

select territory, market\_territory, sub\_market,

count(ebh.dcg\_number) tot\_up\_ebh\_spans

from ng\_reports.domains\_leaf\_reporting dlr

join TOTAL\_LIVE\_EBH\_SPANS ebh

on ebh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

;

cursor MatchedUpLiveT1Spans is

WITH

MATCHED\_LIVE\_T1\_SPANS as

(

select cdm.leaf\_domain\_id

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from ng\_reports.clli\_domain\_map )cdm

on cdm.clli = substr(aud.VSM\_device\_name\_EMS, 1, 6)

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'T1'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

select territory, market\_territory, sub\_market

, count(lt1.ne\_dcg\_number ) tot\_up\_t1\_spans\_w\_l\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_LIVE\_T1\_SPANS lt1

on lt1.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

;

cursor MatchedUpLiveEbhSpans is

WITH

MATCHED\_LIVE\_EBH\_SPANS as

(

select cdm.leaf\_domain\_id

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from ng\_reports.clli\_domain\_map )cdm

on cdm.CLLI = substr(aud.VSM\_device\_name\_EMS, 1, 6)

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

select territory, market\_territory, sub\_market

, count(lebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_l\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_LIVE\_EBH\_SPANS lebh

on lebh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

;

cursor MatchedUpLiveEbhSpansCSR is

WITH

cdm\_map as

(

select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from ng\_reports.clli\_domain\_map

),

MATCHED\_LIVE\_EBH\_CSR\_SPANS as

(

select cdm.leaf\_domain\_id

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join cdm\_map cdm on cdm.clli = substr(aud.VSM\_device\_name\_EMS, 1, 6)

join ng\_topology.trail cpi on aud.trail\_id = CPI.trail\_id

join ng\_reports.port p on cpi.trail\_id = P.ckt\_path\_reference\_id

join ng\_reports.equipment ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

and aud.NG\_BANDWIDTH like '%Mbps'

and EI.EQP\_TYPE = 'CSR'

)

select territory, market\_territory, sub\_market

, count(lebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_l\_pathscsr

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_LIVE\_EBH\_CSR\_SPANS lebh

on lebh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

;

cursor MatchedUpNonLiveT1Spans is

-- NE Up spans matching non-Live T1 spans

WITH

MATCHED\_LIVE\_T1\_SPANS as

(

select aud.NE\_DOM\_IP, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'T1'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

,

MATCHED\_NLIVE\_T1\_SPANS as

(

-- all Up spans that have non-live paths on them

-- minus Up spans that have live paths

select aud.NE\_DOM\_IP, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'T1'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS <> 'LIVE'

minus

select aud.NE\_DOM\_IP, aud.NE\_SPAN\_NUMBER

from MATCHED\_LIVE\_T1\_SPANS aud

)

,

MATCHED\_NL\_T1\_SPANS\_W\_DOMAIN AS

(

select cdm.leaf\_domain\_id

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

JOIN MATCHED\_NLIVE\_T1\_SPANS nl

on aud.NE\_DOM\_IP = nl.NE\_DOM\_IP

and aud.NE\_SPAN\_NUMBER = nl.NE\_SPAN\_NUMBER

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.CLLI = substr(aud.VSM\_device\_name\_EMS, 1, 6)

)

select territory, market\_territory, sub\_market

, count(nlt1.ne\_dcg\_number) tot\_up\_t1\_spans\_w\_nl\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_NL\_T1\_SPANS\_W\_DOMAIN nlt1

on nlt1.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

;

cursor MatchedUpNonLiveEbhSpans is

-- NE Up spans matching non-Live EBH spans

WITH

MATCHED\_LIVE\_EBH\_SPANS as

(

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

,

MATCHED\_NLIVE\_EBH\_SPANS as

(

-- all Up spans that have non-live paths on them

-- minus Up spans that have live paths

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS <> 'LIVE'

minus

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from MATCHED\_LIVE\_EBH\_SPANS aud

)

,

MATCHED\_NL\_EBH\_SPANS\_W\_DOMAIN AS

(

select cdm.leaf\_domain\_id

--, nl.VSM\_DEVICE\_NAME\_EMS

, nl.NE\_DOM\_IP

, nl.ne\_dcg\_number

, nl.ne\_slot\_number

, null NE\_SPAN\_NUMBER

from MATCHED\_NLIVE\_EBH\_SPANS nl

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.clli = substr(nl.VSM\_device\_name\_EMS, 1, 6)

)

select territory, market\_territory, sub\_market

, count(nlebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_nl\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_NL\_EBH\_SPANS\_W\_DOMAIN nlEbh

on nlEbh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

;

cursor MatchedUpNonLiveEbhSpansCSR is

-- NE Up spans matching non-Live EBH spans with CSR Ports

WITH

MATCHED\_LIVE\_EBH\_SPANS\_CSR as

(

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join ng\_topology.trail cpi on aud.trail\_id = CPI.trail\_id

join ng\_reports.port p on cpi.trail\_id = P.ckt\_path\_reference\_id

join ng\_reports.equipment ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

and aud.NG\_BANDWIDTH like '%Mbps'

and EI.EQP\_TYPE = 'CSR'

)

,

MATCHED\_NLIVE\_EBH\_SPANS\_CSR as

(

-- all Up spans that have non-live paths on them

-- minus Up spans that have live paths

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join ng\_topology.trail cpi on aud.trail\_id = CPI.trail\_id

join ng\_reports.port p on cpi.trail\_id = P.ckt\_path\_reference\_id

join ng\_reports.equipment ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS <> 'LIVE'

and aud.NG\_BANDWIDTH like '%Mbps'

and EI.EQP\_TYPE = 'CSR'

minus

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from MATCHED\_LIVE\_EBH\_SPANS\_CSR aud

)

,

MATCHED\_NL\_EBH\_SPANS\_W\_DOM\_CSR AS

(

select cdm.leaf\_domain\_id

--, nl.VSM\_DEVICE\_NAME\_EMS

, nl.NE\_DOM\_IP

, nl.ne\_dcg\_number

, nl.ne\_slot\_number

, null NE\_SPAN\_NUMBER

from MATCHED\_NLIVE\_EBH\_SPANS\_CSR nl

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.clli = substr(nl.VSM\_device\_name\_EMS, 1, 6)

)

select territory, market\_territory, sub\_market

, count(nlebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_nl\_pathscsr

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_NL\_EBH\_SPANS\_W\_DOM\_CSR nlEbh

on nlEbh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market)

;

-- NT\_EVDO\_BTS\_TERM\_REG\_SUMM varchar2(30) = 'Nortel\_EVDO\_BTS\_TERM\_REG\_SUMM';

sqlStmt VARCHAR2(32767);

sqlStmtVZW VARCHAR2(32767);

sqlStmtTerritoty VARCHAR2(32767);

sqlStmtMktTerritory VARCHAR2(32767);

sqlStmtSubMkt VARCHAR2(32767);

sqlStmtT1Comp VARCHAR2(32767);

sqlStmtEbhComp VARCHAR2(32767);

sqlStmtOvlComp VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

t1Reported NUMBER;

ebhReported NUMBER;

BEGIN

sqlStmt := 'truncate table NT\_EVDO\_BTS\_TERM\_REG\_SUMM';

execute immediate sqlStmt;

sqlStmt := 'insert into NT\_EVDO\_BTS\_TERM\_REG\_SUMM

(territory, market\_territory, sub\_market,tot\_up\_t1\_spans,tot\_up\_t1\_spans\_w\_l\_paths,tot\_up\_t1\_spans\_w\_nl\_paths,

tot\_up\_ebh\_spans,tot\_up\_ebh\_spans\_w\_l\_paths,tot\_up\_ebh\_spans\_w\_nl\_paths,tot\_up\_ebh\_spans\_w\_l\_pathscsr,tot\_up\_ebh\_spans\_w\_nl\_pathscsr)

-- Total Up T1 spans

WITH

TOTAL\_LIVE\_T1\_SPANS as

(

select cdm.leaf\_domain\_id,

ndi.\*

from ng\_reports.NORTEL\_DOM\_INVENTORY ndi

join ng\_reports.NORTEL\_PORT\_INVENTORY npi

on ndi.SEQ\_NUM = npi.DOM\_SEQ\_NUM

and npi.PORT\_TYPE = ''T1E1''

and npi.STATUS = ''Up''

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.clli = substr(ndi.EMS\_NAME, 1, 6)

)

select territory, market\_territory, sub\_market, count(t1.dcg\_number) tot\_up\_t1\_spans, null, null, null, null, null, null, null

from ng\_reports.domains\_leaf\_reporting dlr

left outer join TOTAL\_LIVE\_T1\_SPANS t1

on t1.leaf\_domain\_id = dlr.DOMAIN\_ID

where dlr.DOMAIN\_ID not in(51, 45) -- NNO and OSS

group by rollup(dlr.territory, dlr.market\_territory, dlr.sub\_market)

order by dlr.territory, dlr.market\_territory, dlr.sub\_market'

;

execute immediate sqlStmt;

commit;

sqlStmtVZW := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans = :x where territory is null and market\_territory is null and sub\_market is null';

sqlStmtTerritoty := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans = :x where territory = :a and market\_territory is null and sub\_market is null';

sqlStmtMktTerritory := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans = :x where territory = :a and market\_territory = :b and sub\_market is null';

sqlStmtSubMkt := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans = :x where territory = :a and market\_territory = :b and sub\_market = :c';

for cur in totUpEbhSpans

loop

if (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans;

end if;

end loop;

commit;

sqlStmtVZW := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory is null and market\_territory is null and sub\_market is null';

sqlStmtTerritoty := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory = :a and market\_territory is null and sub\_market is null';

sqlStmtMktTerritory := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null ';

sqlStmtSubMkt := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c ';

for cur in matchedUpLiveT1Spans

loop

if (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_t1\_spans\_w\_l\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_t1\_spans\_w\_l\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_t1\_spans\_w\_l\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_t1\_spans\_w\_l\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory is null and market\_territory is null and sub\_market is null';

sqlStmtTerritoty := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory = :a and market\_territory is null and sub\_market is null';

sqlStmtMktTerritory := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null ';

sqlStmtSubMkt := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c ';

for cur in matchedUpLiveEbhSpans

loop

if (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_l\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_l\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_l\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_l\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory is null and market\_territory is null and sub\_market is null';

sqlStmtTerritoty := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory = :a and market\_territory is null and sub\_market is null';

sqlStmtMktTerritory := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market is null ';

sqlStmtSubMkt := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market = :c ';

for cur in matchedUpLiveEbhSpansCSR

loop

if (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr;

end if;

end loop;

commit;

sqlStmtVZW := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory is null and market\_territory is null and sub\_market is null';

sqlStmtTerritoty := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory is null and sub\_market is null';

sqlStmtMktTerritory := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null ';

sqlStmtSubMkt := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c ';

for cur in matchedUpNonLiveT1Spans

loop

if (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_t1\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_t1\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_t1\_spans\_w\_nl\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_t1\_spans\_w\_nl\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory is null and market\_territory is null and sub\_market is null';

sqlStmtTerritoty := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory is null and sub\_market is null';

sqlStmtMktTerritory := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null ';

sqlStmtSubMkt := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c ';

for cur in matchedUpNonLiveEbhSpans

loop

if (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_nl\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_nl\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory is null and market\_territory is null and sub\_market is null';

sqlStmtTerritoty := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory = :a and market\_territory is null and sub\_market is null';

sqlStmtMktTerritory := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market is null ';

sqlStmtSubMkt := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market = :c ';

for cur in matchedUpNonLiveEbhSpansCSR

loop

if (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr;

end if;

end loop;

commit;

sqlStmtT1Comp := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set t1\_comp =:x,t1\_reported =:y where rowid = :a';

sqlStmtEbhComp := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set ebh\_comp =:x,ebh\_reported =:y where rowid = :a';

sqlStmtOvlComp := 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM set overall\_comp =:x where rowid = :a';

for cur in updateCompliance

loop

t1Reported := nvl(cur.tot\_up\_t1\_spans,0) - nvl(cur.tot\_up\_t1\_spans\_w\_nl\_paths,0);

--ebhReported = nvl(cur.tot\_up\_ebh\_spans,0) - nvl(cur.tot\_up\_ebh\_spans\_w\_nl\_paths,0);

ebhReported := nvl(cur.tot\_up\_ebh\_spans,0) - nvl(cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr,0);

overallComp := nvl(cur.tot\_up\_t1\_spans,0) + nvl(cur.tot\_up\_ebh\_spans,0) ;

--overallComp = overallComp - nvl(cur.tot\_up\_t1\_spans\_w\_nl\_paths,0) - nvl(cur.tot\_up\_ebh\_spans\_w\_nl\_paths,0);

overallComp := overallComp - nvl(cur.tot\_up\_t1\_spans\_w\_nl\_paths,0) - nvl(cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr,0);

if(t1Reported >0) then

t1Comp :=(nvl(cur.tot\_up\_t1\_spans\_w\_l\_paths,0)/t1Reported) \*100;

t1Comp := round(t1Comp,2);

execute immediate sqlStmtT1Comp using t1Comp,t1Reported,cur.rd;

--dbms\_output.put\_line('Non null '|| t1Comp);

end if;

if(ebhReported>0) then

--ebhComp :=(nvl(cur.tot\_up\_ebh\_spans\_w\_l\_paths,0)/ebhReported) \*100;

ebhComp :=(nvl(cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr,0)/ebhReported) \*100;

ebhComp := round(ebhComp,2);

execute immediate sqlStmtEbhComp using ebhComp,ebhReported,cur.rd;

--dbms\_output.put\_line('good EBH '||ebhComp);

end if;

if(overallComp>0) then

--overallComp :=((nvl(cur.tot\_up\_t1\_spans\_w\_l\_paths,0)+nvl(cur.tot\_up\_ebh\_spans\_w\_l\_paths,0))/overallComp) \*100;

overallComp :=((nvl(cur.tot\_up\_t1\_spans\_w\_l\_paths,0)+nvl(cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr,0))/overallComp) \*100;

overallComp := round(overallComp,2);

execute immediate sqlStmtOvlComp using overallComp,cur.rd;

--dbms\_output.put\_line('good EBH '||overallComp);

end if;

end loop;

commit;

sqlStmt:= 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt:= 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = MARKET\_TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is not null';

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt:= 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null and wk.MARKET\_TERRITORY is null' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

sqlStmt:= 'update NT\_EVDO\_BTS\_TERM\_REG\_SUMM wk set wk.TERRITORY\_MARKET\_SUB =''Unknown'',wk.sub\_market=''Unknown'' where territory=''Unknown''' ;

EXECUTE IMMEDIATE sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in populate\_evdo\_regional\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END;

Procedure populate\_evdo\_device\_summary is

cursor updateCompliance is

select summ.rowid rd,summ.\* from ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM summ;

cursor totUpEbhSpans is

WITH

-- Total EBH spans

LIVE\_EBH\_PORTS as

(

select distinct npi.DOM\_SEQ\_NUM

from ng\_reports.NORTEL\_PORT\_INVENTORY npi

where npi.PORT\_TYPE = 'Ethernet'

and npi.STATUS = 'Up'

)

,

TOTAL\_LIVE\_EBH\_SPANS as

(

select cdm.leaf\_domain\_id,

ndi.\*

from ng\_reports.NORTEL\_DOM\_INVENTORY ndi

join LIVE\_EBH\_PORTS npi

on ndi.SEQ\_NUM = npi.DOM\_SEQ\_NUM

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.CLLI = substr(ndi.EMS\_NAME, 1, 6)

)

select territory, market\_territory, sub\_market, ebh.ems\_name vsm\_device\_name\_ems

, count(ebh.dcg\_number) tot\_up\_ebh\_spans

from ng\_reports.domains\_leaf\_reporting dlr

join TOTAL\_LIVE\_EBH\_SPANS ebh

on ebh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market, ems\_name)

;

cursor MatchedUpLiveT1Spans is

WITH

MATCHED\_LIVE\_T1\_SPANS as

(

select cdm.leaf\_domain\_id

, aud.vsm\_device\_name\_ems

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.CLLI = substr(aud.VSM\_device\_name\_EMS, 1, 6)

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'T1'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

select territory, market\_territory, sub\_market, vsm\_device\_name\_ems

, count(lt1.ne\_dcg\_number ) tot\_up\_t1\_spans\_w\_l\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_LIVE\_T1\_SPANS lt1

on lt1.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market, vsm\_device\_name\_ems)

;

cursor MatchedUpLiveEbhSpans is

WITH

MATCHED\_LIVE\_EBH\_SPANS as

(

select cdm.leaf\_domain\_id

, vsm\_device\_name\_ems

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.CLLI = substr(aud.VSM\_device\_name\_EMS, 1, 6)

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

select territory, market\_territory, sub\_market, vsm\_device\_name\_ems

, count(lebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_l\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_LIVE\_EBH\_SPANS lebh

on lebh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market, vsm\_device\_name\_ems)

;

cursor MatchedUpLiveEbhSpansCSR is

WITH

cdm\_map as

(

select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map

),

MATCHED\_LIVE\_EBH\_CSR\_SPANS as

(

select cdm.leaf\_domain\_id

, vsm\_device\_name\_ems

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join cdm\_map cdm on cdm.clli = substr(aud.VSM\_device\_name\_EMS, 1, 6)

join ng\_topology.trail cpi on aud.trail\_id = CPI.trail\_id

join ng\_reports.port p on cpi.trail\_id = P.ckt\_path\_reference\_id

join ng\_reports.equipment ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

and aud.NG\_BANDWIDTH like '%Mbps'

and EI.EQP\_TYPE = 'CSR'

)

select territory, market\_territory, sub\_market, vsm\_device\_name\_ems

, count(lebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_l\_pathscsr

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_LIVE\_EBH\_CSR\_SPANS lebh

on lebh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market, vsm\_device\_name\_ems)

;

cursor MatchedUpNonLiveT1Spans is

-- NE Up spans matching non-Live T1 spans

WITH

MATCHED\_LIVE\_T1\_SPANS as

(

select aud.NE\_DOM\_IP, aud.NE\_SPAN\_NUMBER, vsm\_device\_name\_ems

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'T1'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

,

MATCHED\_NLIVE\_T1\_SPANS as

(

-- all Up spans that have non-live paths on them

-- minus Up spans that have live paths

select aud.NE\_DOM\_IP, aud.NE\_SPAN\_NUMBER, vsm\_device\_name\_ems

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'T1'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS <> 'LIVE'

minus

select aud.NE\_DOM\_IP, aud.NE\_SPAN\_NUMBER, vsm\_device\_name\_ems

from MATCHED\_LIVE\_T1\_SPANS aud

)

,

MATCHED\_NL\_T1\_SPANS\_W\_DOMAIN AS

(

select cdm.leaf\_domain\_id

, aud.vsm\_device\_name\_ems

, aud.NE\_DOM\_IP

, aud.ne\_dcg\_number

, aud.ne\_slot\_number

, aud.NE\_SPAN\_NUMBER

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

JOIN MATCHED\_NLIVE\_T1\_SPANS nl

on aud.NE\_DOM\_IP = nl.NE\_DOM\_IP

and aud.NE\_SPAN\_NUMBER = nl.NE\_SPAN\_NUMBER

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.clli = substr(aud.VSM\_device\_name\_EMS, 1, 6)

)

select territory, market\_territory, sub\_market, vsm\_device\_name\_ems

, count(nlt1.ne\_dcg\_number) tot\_up\_t1\_spans\_w\_nl\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_NL\_T1\_SPANS\_W\_DOMAIN nlt1

on nlt1.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market, vsm\_device\_name\_ems)

;

cursor MatchedUpNonLiveEbhSpans is

-- NE Up spans matching non-Live EBH spans

WITH

MATCHED\_LIVE\_EBH\_SPANS as

(

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

)

,

MATCHED\_NLIVE\_EBH\_SPANS as

(

-- all Up spans that have non-live paths on them

-- minus Up spans that have live paths

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS <> 'LIVE'

minus

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from MATCHED\_LIVE\_EBH\_SPANS aud

)

,

MATCHED\_NL\_EBH\_SPANS\_W\_DOMAIN AS

(

select cdm.leaf\_domain\_id

, nl.vsm\_device\_name\_ems

, nl.NE\_DOM\_IP

, nl.ne\_dcg\_number

, nl.ne\_slot\_number

, null NE\_SPAN\_NUMBER

from MATCHED\_NLIVE\_EBH\_SPANS nl

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.clli = substr(nl.VSM\_device\_name\_EMS, 1, 6)

)

select territory, market\_territory, sub\_market, vsm\_device\_name\_ems

, count(nlebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_nl\_paths

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_NL\_EBH\_SPANS\_W\_DOMAIN nlEbh

on nlEbh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market, vsm\_device\_name\_ems)

;

cursor MatchedUpNonLiveEbhSpansCSR is

-- NE Up spans matching non-Live EBH spans with CSR ports

WITH

MATCHED\_LIVE\_EBH\_SPANS\_CSR as

(

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join ng\_topology.trail cpi on aud.trail\_id = CPI.trail\_id

join ng\_reports.port p on cpi.trail\_id = P.ckt\_path\_reference\_id

join ng\_reports.equipment ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS = 'LIVE'

and aud.NG\_BANDWIDTH like '%Mbps'

and EI.EQP\_TYPE = 'CSR'

)

,

MATCHED\_NLIVE\_EBH\_SPANS\_CSR as

(

-- all Up spans that have non-live paths on them

-- minus Up spans that have live paths

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from ng\_reports.nortel\_ne\_vs\_ng\_audit aud

join ng\_topology.trail cpi on aud.trail\_id = CPI.trail\_id

join ng\_reports.port p on cpi.trail\_id = P.ckt\_path\_reference\_id

join ng\_reports.equipment ei on EI.eqp\_reference\_id = P.eqp\_reference\_id

where aud.MATCH\_CODE = 'BOTH'

and aud.NE\_SPAN\_TYPE = 'DATA\_SPAN'

and aud.TERMINATION\_TYPE = 'Ethernet'

and aud.NE\_STATUS = 'Up'

and aud.NG\_PATH\_STATUS <> 'LIVE'

and aud.NG\_BANDWIDTH like '%Mbps'

and EI.EQP\_TYPE = 'CSR'

minus

select aud.VSM\_DEVICE\_NAME\_EMS, aud.NE\_DOM\_IP, aud.ne\_dcg\_number, aud.ne\_slot\_number

from MATCHED\_LIVE\_EBH\_SPANS\_CSR aud

)

,

MATCHED\_NL\_EBH\_SPANS\_W\_DOM\_CSR AS

(

select cdm.leaf\_domain\_id

, nl.VSM\_DEVICE\_NAME\_EMS

, nl.NE\_DOM\_IP

, nl.ne\_dcg\_number

, nl.ne\_slot\_number

, null NE\_SPAN\_NUMBER

from MATCHED\_NLIVE\_EBH\_SPANS\_CSR nl

join (select distinct leaf\_domain\_id,substr(CLLI, 1, 6) clli from clli\_domain\_map )cdm

on cdm.clli = substr(nl.VSM\_device\_name\_EMS, 1, 6)

)

select territory, market\_territory, sub\_market, vsm\_device\_name\_ems

, count(nlebh.ne\_dcg\_number) tot\_up\_ebh\_spans\_w\_nl\_pathscsr

from ng\_reports.domains\_leaf\_reporting dlr

join MATCHED\_NL\_EBH\_SPANS\_W\_DOM\_CSR nlEbh

on nlEbh.leaf\_domain\_id = dlr.DOMAIN\_ID

group by rollup (dlr.territory, dlr.market\_territory, dlr.sub\_market, vsm\_device\_name\_ems)

;

sqlStmt VARCHAR2(32767);

sqlStmtVZW VARCHAR2(32767);

sqlStmtTerritoty VARCHAR2(32767);

sqlStmtMktTerritory VARCHAR2(32767);

sqlStmtSubMkt VARCHAR2(32767);

sqlStmtEms VARCHAR2(32767);

sqlStmtT1Comp VARCHAR2(32767);

sqlStmtEbhComp VARCHAR2(32767);

sqlStmtOvlComp VARCHAR2(32767);

t1Comp NUMBER;

ebhComp NUMBER;

overallComp NUMBER;

--ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM varchar2(30) = 'Nortel\_EVDO\_BTS\_TERM\_EMS\_SUMM';

BEGIN

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' ');

dbms\_output.put\_line(' ');

sqlStmt := 'truncate table ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM';

execute immediate sqlStmt;

sqlStmt := 'insert into ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM

(territory, market\_territory, sub\_market,leaf\_domain\_id,vsm\_device\_name\_ems,tot\_up\_t1\_spans,tot\_up\_t1\_spans\_w\_l\_paths,tot\_up\_t1\_spans\_w\_nl\_paths,

tot\_up\_ebh\_spans,tot\_up\_ebh\_spans\_w\_l\_paths,tot\_up\_ebh\_spans\_w\_nl\_paths, tot\_up\_ebh\_spans\_w\_l\_pathscsr,tot\_up\_ebh\_spans\_w\_nl\_pathscsr)

-- Total Up T1 spans

WITH

ALL\_EMS as

(

select distinct dlr.territory, dlr.market\_territory, dlr.sub\_market, dlr.cdm.leaf\_domain\_id, ndi.EMS\_NAME

from ng\_reports.domains\_leaf\_reporting dlr

left outer join clli\_domain\_map cdm

on cdm.leaf\_domain\_id = dlr.DOMAIN\_ID

left outer join ng\_reports.NORTEL\_DOM\_INVENTORY ndi

on substr(CLLI, 1, 6) = substr(ndi.EMS\_NAME, 1, 6)

where dlr.DOMAIN\_ID not in(51, 45) -- NNO and OSS

)

,

TOTAL\_LIVE\_T1\_SPANS as

(

select ndi.\*

from ng\_reports.NORTEL\_DOM\_INVENTORY ndi

join ng\_reports.NORTEL\_PORT\_INVENTORY npi

on ndi.SEQ\_NUM = npi.DOM\_SEQ\_NUM

and npi.PORT\_TYPE = ''T1E1''

and npi.STATUS = ''Up''

)

select territory, market\_territory, sub\_market, dlr.leaf\_domain\_id

, dlr.ems\_name, count(t1.dcg\_number) tot\_up\_t1\_spans, null, null, null, null, null, null, null

from ALL\_EMS dlr

left outer join TOTAL\_LIVE\_T1\_SPANS t1

on t1.ems\_name = dlr.ems\_name

group by rollup (territory, market\_territory, sub\_market, dlr.leaf\_domain\_id, dlr.ems\_name)

order by territory, market\_territory, sub\_market, dlr.leaf\_domain\_id, dlr.ems\_name'

;

execute immediate sqlStmt;

commit;

-- dont need leaf domain rollups. delete those rows

sqlStmt := 'delete ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM where leaf\_domain\_id is not null and vsm\_device\_name\_ems is null';

execute immediate sqlStmt;

commit;

sqlStmtVZW := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans = :x where territory is null and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtTerritoty := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans = :x where territory = :a and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtMktTerritory := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans = :x where territory = :a and market\_territory = :b and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtSubMkt := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems is null';

sqlStmtEms := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems = :d';

for cur in totUpEbhSpans

loop

--dbms\_output.put\_line('In totUpEbhSpans');

if (cur.vsm\_device\_name\_ems is not null) then

execute immediate sqlStmtEms using cur.tot\_up\_ebh\_spans, cur.territory, cur.market\_territory, cur.sub\_market, cur.vsm\_device\_name\_ems;

elsif (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans;

end if;

end loop;

commit;

sqlStmtVZW := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory is null and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtTerritoty := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory = :a and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtMktTerritory := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtSubMkt := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems is null';

sqlStmtEms := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems = :d';

for cur in matchedUpLiveT1Spans

loop

if (cur.vsm\_device\_name\_ems is not null) then

--dbms\_output.put\_line('In matchedUpLiveT1Spans. '||sqlStmtEms||': '||cur.tot\_up\_t1\_spans\_w\_l\_paths||' , '|| cur.area||' , '|| cur.region ||' , '||cur.vsm\_device\_name\_ems);

execute immediate sqlStmtEms using cur.tot\_up\_t1\_spans\_w\_l\_paths, cur.territory, cur.market\_territory, cur.sub\_market, cur.vsm\_device\_name\_ems;

elsif (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_t1\_spans\_w\_l\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_t1\_spans\_w\_l\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_t1\_spans\_w\_l\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_t1\_spans\_w\_l\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory is null and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtTerritoty := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory = :a and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtMktTerritory := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtSubMkt := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems is null';

sqlStmtEms := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems = :d';

for cur in matchedUpLiveEbhSpans

loop

dbms\_output.put\_line('In matchedUpLiveEbhSpans. '||sqlStmtEms||': '||cur.tot\_up\_ebh\_spans\_w\_l\_paths||' , '|| cur.territory||' , '|| cur.market\_territory ||', '|| cur.sub\_market ||' , '||cur.vsm\_device\_name\_ems);

if (cur.vsm\_device\_name\_ems is not null) then

execute immediate sqlStmtEms using cur.tot\_up\_ebh\_spans\_w\_l\_paths, cur.territory, cur.market\_territory, cur.sub\_market, cur.vsm\_device\_name\_ems;

elsif (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_l\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_l\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_l\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_l\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory is null and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtTerritoty := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory = :a and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtMktTerritory := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtSubMkt := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems is null';

sqlStmtEms := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_l\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems = :d';

for cur in matchedUpLiveEbhSpansCSR

loop

--dbms\_output.put\_line('In matchedUpLiveEbhSpansCSR. '||sqlStmtEms||': '||cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr||' , '|| cur.area||' , '|| cur.region ||' , '||cur.vsm\_device\_name\_ems);

if (cur.vsm\_device\_name\_ems is not null) then

execute immediate sqlStmtEms using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr, cur.territory, cur.market\_territory, cur.sub\_market, cur.vsm\_device\_name\_ems;

elsif (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_l\_pathscsr;

end if;

end loop;

commit;

sqlStmtVZW := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory is null and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtTerritoty := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtMktTerritory := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtSubMkt := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems is null';

sqlStmtEms := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_t1\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems = :d';

for cur in matchedUpNonLiveT1Spans

loop

if (cur.vsm\_device\_name\_ems is not null) then

--dbms\_output.put\_line('In matchedUpNonLiveT1Spans. '||sqlStmtEms||': '||cur.tot\_up\_t1\_spans\_w\_nl\_paths||' , '|| cur.area||' , '|| cur.region ||' , '||cur.vsm\_device\_name\_ems);

execute immediate sqlStmtEms using cur.tot\_up\_t1\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory, cur.sub\_market, cur.vsm\_device\_name\_ems;

elsif (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_t1\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_t1\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_t1\_spans\_w\_nl\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_t1\_spans\_w\_nl\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory is null and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtTerritoty := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtMktTerritory := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtSubMkt := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems is null';

sqlStmtEms := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_paths = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems = :d';

for cur in matchedUpNonLiveEbhSpans

loop

if (cur.vsm\_device\_name\_ems is not null) then

--dbms\_output.put\_line('In matchedUpNonLiveEbhSpans. '||sqlStmtEms||': '||cur.tot\_up\_ebh\_spans\_w\_nl\_paths||' , '|| cur.area||' , '|| cur.region ||' , '||cur.vsm\_device\_name\_ems);

execute immediate sqlStmtEms using cur.tot\_up\_ebh\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory, cur.sub\_market, cur.vsm\_device\_name\_ems;

elsif (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_nl\_paths, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_nl\_paths, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_nl\_paths;

end if;

end loop;

commit;

sqlStmtVZW := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory is null and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtTerritoty := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory = :a and market\_territory is null and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtMktTerritory := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market is null and vsm\_device\_name\_ems is null';

sqlStmtSubMkt := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems is null';

sqlStmtEms := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set tot\_up\_ebh\_spans\_w\_nl\_pathscsr = :x where territory = :a and market\_territory = :b and sub\_market = :c and vsm\_device\_name\_ems = :d';

for cur in matchedUpNonLiveEbhSpansCSR

loop

if (cur.vsm\_device\_name\_ems is not null) then

--dbms\_output.put\_line('In matchedUpNonLiveEbhSpansCSR. '||sqlStmtEms||': '||cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr||' , '|| cur.area||' , '|| cur.region ||' , '||cur.vsm\_device\_name\_ems);

execute immediate sqlStmtEms using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr, cur.territory, cur.market\_territory, cur.sub\_market, cur.vsm\_device\_name\_ems;

elsif (cur.sub\_market is not null) then

execute immediate sqlStmtSubMkt using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr, cur.territory, cur.market\_territory, cur.sub\_market;

elsif (cur.market\_territory is not null) then

execute immediate sqlStmtMktTerritory using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr, cur.territory, cur.market\_territory;

elsif (cur.territory is not null) then

execute immediate sqlStmtTerritoty using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr, cur.territory;

else

execute immediate sqlStmtVZW using cur.tot\_up\_ebh\_spans\_w\_nl\_pathscsr;

end if;

end loop;

commit;

sqlStmtT1Comp := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set t1\_comp =:x where rowid = :a';

sqlStmtEbhComp := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set ebh\_comp =:x where rowid = :a';

sqlStmtOvlComp := 'update ng\_reports.NT\_EVDO\_BTS\_TERM\_EMS\_SUMM set overall\_comp =:x where rowid = :a';

for cur in updateCompliance

loop

t1Comp := nvl(cur.tot\_up\_t1\_spans,0) - nvl(cur.tot\_up\_t1\_spans\_w\_nl\_paths,0);

ebhComp := nvl(cur.tot\_up\_ebh\_spans,0) - nvl(cur.tot\_up\_ebh\_spans\_w\_nl\_paths,0);

overallComp := nvl(cur.tot\_up\_t1\_spans,0) + nvl(cur.tot\_up\_ebh\_spans,0) ;

overallComp := overallComp - nvl(cur.tot\_up\_t1\_spans\_w\_nl\_paths,0) - nvl(cur.tot\_up\_ebh\_spans\_w\_nl\_paths,0);

if(t1Comp >0) then

t1Comp :=(nvl(cur.tot\_up\_t1\_spans\_w\_l\_paths,0)/t1Comp) \*100;

t1Comp := round(t1Comp,2);

execute immediate sqlStmtT1Comp using t1Comp,cur.rd;

--dbms\_output.put\_line('Non null '|| t1Comp);

end if;

if(ebhComp>0) then

ebhComp :=(nvl(cur.tot\_up\_ebh\_spans\_w\_l\_paths,0)/ebhComp) \*100;

ebhComp := round(ebhComp,2);

execute immediate sqlStmtEbhComp using ebhComp,cur.rd;

--dbms\_output.put\_line('good EBH '||ebhComp);

end if;

if(overallComp>0) then

overallComp :=((nvl(cur.tot\_up\_t1\_spans\_w\_l\_paths,0)+nvl(cur.tot\_up\_ebh\_spans\_w\_l\_paths,0))/overallComp) \*100;

overallComp := round(overallComp,2);

execute immediate sqlStmtOvlComp using overallComp,cur.rd;

--dbms\_output.put\_line('good EBH '||overallComp);

end if;

end loop;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(sqlStmt);

dbms\_output.put\_line(SubStr('Error in populate\_device\_summary();: '||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

raise;

END;

END;

--/

/

--------------------------------------------------------

-- DDL for Package Body NTLS

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."NTLS" is

url1 varchar2(50);

procedure home (advSearch in varchar2 default 'N', pwdWarning in varchar2 default 'N' ) is

var\_url number;

req utl\_http.req;

res utl\_http.resp;

url varchar2(4000);

var\_name varchar2(4000);

p\_json varchar2(4000);

buffer varchar2(4000);

content varchar2(4000);

begin

p\_json :='Test';

url:= 'http://nj51nt4vda4v.nss.vzwnet.com:10027/location/search/v3'; -- use side car URL.

content := '{ "frRefKeyName": "GRANITE", "frRefKeyValues": [ "82442" ] }';

dbms\_output.put\_line(content);

dbms\_output.put\_line(url);

req := utl\_http.begin\_request(url, 'PUT',' HTTP/1.1');

utl\_http.set\_header(req, 'user-agent', 'mozilla/4.0');

utl\_http.set\_header(req, 'content-type', 'application/json');

utl\_http.set\_header(req, 'Content-Length', length(content));

utl\_http.set\_header(req, 'Authorization', 'Basic bnVsbDpudWxs');

utl\_http.write\_text(req, content);

res := utl\_http.get\_response(req);

begin

loop

utl\_http.read\_line(res, buffer);

dbms\_output.put\_line(buffer);

end loop;

utl\_http.end\_response(res);

exception

when utl\_http.end\_of\_body

then

utl\_http.end\_response(res);

end;

owa\_util.redirect\_url(curl => 'https://xngapp1.vh.vzwnet.com/GranitePortal/layout.html' );

--apex\_util.redirect\_url ( p\_url => 'http://www.oracle.com/' );

end home;

FUNCTION authorize return boolean is

begin

return true;

end authorize;

end ntls;

/

--------------------------------------------------------

-- DDL for Package Body RPT\_IAN\_NOCC\_SITE\_TEST

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."RPT\_IAN\_NOCC\_SITE\_TEST"

IS

--

-- Purpose: NOCC SITE ALARAMS AGGREAGATE SUMMARY and DETAIL reports.

--

-- MODIFICATION HISTORY

-- Person Date Comments

-- ---------------------- ------------ -----------------------------------------

-- Krishna Gadde Oct 2005 INitial release.

-- Krihsna Gadde March 2006 Tunned...

-- Krishna Gadde Aug 2006 fixed query. Issue with val\_attr\_group

-- Steve Ervin Mar 2016 Convert to Territory/Market/SubMarket

-- Steve Ervin Aug 2017 Add support for voluntarily lit sites

PROCEDURE getAllDomains

IS

TYPE l\_cursor\_Type IS REF CURSOR;

Cursor\_Domains L\_Cursor\_Type;

L\_Territory Xng\_Reports.Vzw\_Network\_Org.Territory%Type;

L\_Market\_Territory Xng\_Reports.Vzw\_Network\_Org.Market\_Territory%Type;

L\_Sub\_Market Xng\_Reports.Vzw\_Network\_Org.Market\_Territory%Type;

L\_Domainid Vzwnet.Domain\_Inst.Domain\_Inst\_Id%Type;

L\_Domainname Vzwnet.Domain\_Inst.Domain\_Name%Type;

sql\_Str VARCHAR2 (555);

BEGIN

--sql\_Str := 'SELECT vno.Territory, vno.Market\_Territory, vno.Sub\_Market, di.Domain\_inst\_id, di.domain\_name

-- FROM xng\_reports.VZW\_NetWork\_Org vno, vzwnet.Domain\_Inst di

-- WHERE di.domain\_name = vno.sub\_market\_leaf\_domain\_name

-- and di.domain\_inst\_id = 1012

-- GROUP BY vno.Territory, vno.Market\_Territory, vno.Sub\_Market, di.Domain\_inst\_id, di.domain\_name

-- ORDER BY vno.Territory, vno.Market\_Territory, vno.Sub\_Market';

sql\_Str := 'SELECT DLR.TERRITORY, DLR.MARKET\_TERRITORY, DLR.SUB\_MARKET, DI.DOMAIN\_INST\_ID, DI.DOMAIN\_NAME

FROM DOMAINS\_LEAF\_REPORTING DLR, VZWNET.DOMAIN\_INST DI

WHERE di.domain\_name = DLR.domain\_name

and di.domain\_inst\_id = 1012

GROUP BY DLR.TERRITORY, DLR.MARKET\_TERRITORY, DLR.SUB\_MARKET, DI.DOMAIN\_INST\_ID, DI.DOMAIN\_NAME

ORDER BY DLR.Territory, DLR.Market\_Territory, DLR.Sub\_Market';

OPEN cursor\_Domains FOR sql\_Str;

Loop

FETCH cursor\_Domains INTO l\_Territory, l\_Market\_Territory, l\_Sub\_Market, l\_DomainId, l\_DomainName;

EXIT WHEN cursor\_Domains%NOTFOUND;

DBMS\_OUTPUT.put\_line('Loading Domain...: ' || l\_DomainName );

-- Remove the old Detail data by domain's...

removeDetails( l\_DomainId );

-- get all Sites for a passed domain...

getAllSites( l\_DomainId, l\_DomainName, l\_Territory, l\_Market\_Territory, l\_Sub\_Market );

COMMIT;

END LOOP;

CLOSE cursor\_Domains;

-- Not enough rollback segment...Moved up into the loop COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

IF( cursor\_Domains%ISOPEN ) THEN

CLOSE cursor\_Domains;

END IF;

Dbms\_Output.Put\_Line( ' Territory : ' || L\_Territory ||

', Market\_Territory : ' || L\_Market\_Territory ||

', Sub\_Market : ' || L\_Sub\_Market ||

', DomainId : ' || L\_Domainid ||

', DomainName:' || L\_Domainname ||

', SQLCODE: ' || TO\_CHAR(SQLCODE));

RAISE;

END;

Procedure Getallsites(

P\_Domainid In Vzwnet.Domain\_Inst.Domain\_Inst\_Id%Type,

P\_Domainname In Vzwnet.Domain\_Inst.Domain\_Name%Type,

P\_Territory In Xng\_Reports.Vzw\_Network\_Org.Territory%Type,

P\_Market\_Territory In Xng\_Reports.Vzw\_Network\_Org.Market\_Territory%Type,

P\_Sub\_Market In Xng\_Reports.Vzw\_Network\_Org.Sub\_Market%Type)

IS

-- For NOC\_Aggs Summary report....

l\_EquipInstId vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE;

l\_TowerStatus vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE;

l\_AntennaVZWOwned VARCHAR2(255);

l\_HasAntennaTower VARCHAR2(30);

l\_LightingRequired VARCHAR2(30);

l\_LightingVoluntary VARCHAR2(30);

l\_LightingTestPeriod VARCHAR2(30);

l\_SiteInstId NOCC\_LAST\_TEST\_DETAILS.site\_inst\_id%TYPE;

l\_SiteName NOCC\_LAST\_TEST\_DETAILS.site\_name%TYPE;

l\_SiteId NOCC\_LAST\_TEST\_DETAILS.site\_id%TYPE; -- Site\_Inst.Base\_Num

l\_SiteTechName NOCC\_LAST\_TEST\_DETAILS.site\_tech\_name%TYPE;

l\_inventoried\_cells NOCC\_LAST\_TEST\_DETAILS.inventoried\_cells%TYPE;

l\_SiteType NOCC\_LAST\_TEST\_DETAILS.site\_type%TYPE; -- Site\_Inst.Num

l\_NOCC\_Monitored NOCC\_LAST\_TEST\_DETAILS.nocc\_monitored%TYPE;

l\_LightLastTestedDate NOCC\_LAST\_TEST\_DETAILS.light\_last\_tested%TYPE;

l\_LightNextTestDate NOCC\_LAST\_TEST\_DETAILS.light\_next\_test%TYPE;

l\_LightTestOverDue NOCC\_LAST\_TEST\_DETAILS.light\_test\_overdue%TYPE;

l\_EnvLastTestedDate NOCC\_LAST\_TEST\_DETAILS.env\_last\_tested%TYPE;

l\_EnvNextTestDate NOCC\_LAST\_TEST\_DETAILS.env\_next\_test%TYPE;

l\_EnvTestOverDue NOCC\_LAST\_TEST\_DETAILS.env\_test\_overdue%TYPE;

-- For NOC\_Aggs Summary report....

l\_Site\_Count NOCC\_LAST\_TEST\_SUMMARY.sites%TYPE := 0;

l\_HasAntennaTower\_Count NOCC\_LAST\_TEST\_SUMMARY.antennas%TYPE := 0;

l\_AntennaVZWOwned\_Count NOCC\_LAST\_TEST\_SUMMARY.vzw\_owned%TYPE := 0;

l\_LightingRequired\_Count NOCC\_LAST\_TEST\_SUMMARY.lights\_reqd%TYPE := 0;

l\_LightTestedDate\_Count NOCC\_LAST\_TEST\_SUMMARY.lights\_tested%TYPE := 0;

l\_LightTestOverDue\_Count NOCC\_LAST\_TEST\_SUMMARY.lights\_test\_overdue%TYPE := 0;

l\_LightTest\_OK\_Count NOCC\_LAST\_TEST\_SUMMARY.lights\_test\_ok%TYPE := 0;

l\_NOCC\_Monitored\_Count NOCC\_LAST\_TEST\_SUMMARY.NOCC\_monitored\_sites%TYPE := 0;

l\_EnvLastTestedDate\_Count NOCC\_LAST\_TEST\_SUMMARY.env\_tested%TYPE := 0;

l\_EnvTestOverDue\_Count NOCC\_LAST\_TEST\_SUMMARY.env\_test\_overdue%TYPE := 0;

l\_EnvTestOK\_Count NOCC\_LAST\_TEST\_SUMMARY.env\_test\_ok%TYPE := 0;

l\_saved\_at\_equipment\_level number := 1;

--## ADD Include List for SITE\_TYPES

CURSOR csr\_getAllSites( pp\_DomainId IN vzwnet.Domain\_Inst.domain\_inst\_id%Type )

IS

SELECT distinct si.Site\_Inst\_Id, si.Site\_hum\_Id, si.Base\_Num, si.Num

FROM vzwnet.Site\_Domain\_map sdm,

vzwnet.Site\_Inst si

WHERE sdm.Domain\_Inst\_Id = pp\_DomainId

AND si.Num in ('MICROWAVE')

AND si.Site\_Inst\_Id = sdm.Site\_Inst\_Id;

--and si.site\_inst\_id not in (select site\_inst\_id from vzwnet.site\_domain\_map where domain\_inst\_id = 1191); --to keep out ALLTEL sites

CURSOR csr\_getSiteEquipment( pp\_SiteInstId IN vzwnet.Site\_Inst.site\_inst\_id%Type )

IS

SELECT ei.Equip\_Inst\_Id

FROM vzwnet.Equip\_Inst ei

WHERE ei.Site\_Inst\_Id (+) = pp\_SiteInstId

AND UPPER(ei.type (+)) = 'ANTENNA SUPPORT STRUCTURE';

BEGIN

OPEN csr\_getAllSites( p\_DomainId );

LOOP

FETCH csr\_getAllSites INTO l\_SiteInstId, l\_SiteName, l\_SiteId, l\_SiteType;

EXIT WHEN csr\_getAllSites%NOTFOUND;

l\_NOCC\_Monitored := NULL;

l\_SiteTechName := NULL;

l\_inventoried\_cells := NULL;

l\_TowerStatus := NULL;

l\_EnvLastTestedDate := NULL;

l\_EnvNextTestDate := NULL;

L\_Envtestoverdue := Null;

l\_saved\_at\_equipment\_level := 1;

l\_Site\_Count := l\_Site\_Count + 1;

l\_SiteTechName := getSiteTechName( l\_SiteInstId );

l\_inventoried\_cells := getCellEquipDesc( l\_SiteInstId );

-- Environmental details...

l\_NOCC\_Monitored := getIsNOCCMonitored( l\_SiteInstId );

IF( upper(l\_NOCC\_Monitored) <> 'NO' ) THEN

l\_NOCC\_Monitored\_Count := l\_NOCC\_Monitored\_Count + 1;

l\_EnvLastTestedDate := varchar2date( getEnvLastTestedDate( l\_SiteInstId, l\_SiteType ));

IF( l\_EnvLastTestedDate IS NOT NULL ) THEN

l\_EnvLastTestedDate\_Count := l\_EnvLastTestedDate\_Count + 1;

-- Changed according to Jim 2/6/2006, --IF( (upper(l\_SiteType) = 'MTSO' ) OR (upper(l\_SiteType) = 'HUB' ) ) THEN

IF ( upper(l\_SiteType) = 'MTSO' ) THEN

l\_EnvNextTestDate := TO\_CHAR( ADD\_MONTHS( TO\_DATE( l\_EnvLastTestedDate, 'DD\_MON\_YY'), 6 ));

ELSIF ( upper(l\_SiteType) = 'CELL' ) THEN

-- Conditional interval based on Failed First Environmental Attempt UDA

l\_EnvNextTestDate := TO\_CHAR( ADD\_MONTHS( TO\_DATE( l\_EnvLastTestedDate, 'DD\_MON\_YY'),

12\* getFailedFirstEnvAttempt( l\_SiteInstId )));

ELSE

l\_EnvNextTestDate := TO\_CHAR( ADD\_MONTHS( TO\_DATE( l\_EnvLastTestedDate, 'DD\_MON\_YY'), 12 ));

END IF;

l\_EnvTestOverDue := TO\_CHAR( TO\_DATE( l\_EnvNextTestDate, 'DD\_MON\_YY') - TRUNC(SYSDATE ));

IF( TO\_NUMBER(l\_EnvTestOverDue ) <= 0 ) THEN

DBMS\_OUTPUT.put\_line('OVERDUE 1 '|| l\_SiteInstId || to\_char(l\_EnvTestOverDue\_Count));

l\_EnvTestOverDue\_Count := l\_EnvTestOverDue\_Count + 1;

ELSE

l\_EnvTestOK\_Count := l\_EnvTestOK\_Count + 1;

END IF;

ELSE

DBMS\_OUTPUT.put\_line('OVERDUE 2'|| l\_SiteInstId || to\_char(l\_EnvTestOverDue\_Count));

l\_EnvTestOverDue\_Count := l\_EnvTestOverDue\_Count + 1;

-- added 04/21/2006 JLA

l\_EnvNextTestDate := 'ASAP';

l\_EnvTestOverDue := -90;

--

END IF; -- l\_ENVLastTestedDate

END IF; -- l\_NOCC\_Monitored

l\_HasAntennaTower := NULL;

l\_AntennaVZWOwned := NULL;

l\_LightingRequired := NULL;

l\_LightingVoluntary := NULL;

l\_LightLastTestedDate := NULL;

l\_LightNextTestDate := NULL;

l\_LightTestOverDue := NULL;

-- Antenna details...

OPEN csr\_getSiteEquipment( l\_SiteInstId );

LOOP

FETCH csr\_getSiteEquipment INTO l\_EquipInstId;

EXIT WHEN csr\_getSiteEquipment%NOTFOUND;

l\_HasAntennaTower := NULL;

l\_AntennaVZWOwned := NULL;

l\_LightingRequired := NULL;

l\_LightingVoluntary := NULL;

l\_LightLastTestedDate := NULL;

l\_LightNextTestDate := NULL;

l\_LightTestOverDue := NULL;

l\_HasAntennaTower := getHasAntennaTower( l\_SiteInstId, l\_EquipInstId );

IF( l\_HasAntennaTower = 'Has Tower' ) THEN

l\_HasAntennaTower\_Count := l\_HasAntennaTower\_Count + 1;

--l\_HasAntennaTower := 'Has Tower';

l\_AntennaVZWOwned := getIsAntennaVZWOwned( l\_SiteInstId, l\_EquipInstId );

-- IF( upper(l\_AntennaVZWOwned) <> 'OTHER' ) THEN

IF( upper(l\_AntennaVZWOwned) = 'VERIZON WIRELESS' ) THEN

-- AntennaStructure is 'VERIZON WIRELESS' or empty, assume its VZW's.

l\_AntennaVZWOwned\_Count := l\_AntennaVZWOwned\_Count + 1;

l\_LightingRequired := getIsLightingRequired ( l\_SiteInstId, l\_EquipInstId );

l\_LightingVoluntary := getIsLightingVoluntary( l\_SiteInstId, l\_EquipInstId );

IF( upper(l\_LightingRequired) <> 'NO' OR l\_LightingVoluntary <> 'NO') THEN

-- If it is clearly specified 'NO' then its no, otherwise its required!.

l\_LightingRequired\_Count := l\_LightingRequired\_Count + 1;

l\_LightingTestPeriod := getLightingTestPeriod( l\_SiteInstId, l\_EquipInstId );

l\_LightLastTestedDate := varchar2date( getTowerLightLastTestedDate( l\_SiteInstId , l\_SiteType ));

IF( l\_LightLastTestedDate IS NOT NULL ) THEN

l\_LightTestedDate\_Count := l\_LightTestedDate\_Count + 1;

IF ( Upper(l\_LightingTestPeriod) = 'ANNUALLY' ) THEN

l\_lightnexttestdate := to\_char( add\_months( to\_date( l\_lightlasttesteddate, 'DD\_MON\_YY'), 12 ) );

l\_lighttestoverdue := to\_char( to\_date( l\_lightnexttestdate, 'DD-MON-YY') - trunc(SYSDATE) );

ELSE -- Quarterly

l\_lightnexttestdate := to\_char( to\_date( l\_lightlasttesteddate, 'DD-MON-YY') + 90 );

l\_lighttestoverdue := to\_char( to\_date( l\_lightnexttestdate, 'DD-MON-YY') - trunc(SYSDATE) );

END IF;

IF ( TO\_NUMBER( l\_LightTestOverDue ) <= 0 ) THEN

l\_LightTestOverDue\_Count := l\_LightTestOverDue\_Count + 1;

l\_LightNextTestDate := 'ASAP';

ELSE

l\_LightTest\_OK\_Count := l\_LightTest\_OK\_Count + 1;

END IF; -- l\_LightTestOverDue

ELSE

l\_LightTestOverDue\_Count := l\_LightTestOverDue\_Count + 1;

--added 04/21/2006 JLA

l\_LightNextTestDate := 'ASAP';

--l\_LightTestOverDue := '-90';

IF( l\_LightingTestPeriod = 'ANNUALLY' ) THEN

l\_LightTestOverDue := '-365';

ELSE

l\_LightTestOverDue := '-90';

END IF;

--

END IF; --l\_LightLastTestedDate

ELSE

l\_LightLastTestedDate := 'Light Not Reqd';

l\_LightNextTestDate := 'N/A';

END IF; -- l\_LightingRequired or l\_LightingVoluntary

ELSE

l\_AntennaVZWOwned := 'Not VZW Owned';

END IF; -- l\_AntennaVZWOwned

ELSE

l\_LightLastTestedDate := 'No Tower';

l\_LightNextTestDate := 'N/A';

END IF; -- l\_HasAntennaTower

-- Save detail record...

Insert\_NOCC\_Last\_Test\_Details( p\_DomainId,

p\_DomainName,

l\_SiteInstId,

l\_SiteName,

l\_SiteId,

l\_SiteType,

l\_SiteTechName,

l\_inventoried\_cells,

l\_NOCC\_Monitored,

l\_AntennaVZWOwned,

l\_LightingRequired,

l\_LightLastTestedDate,

l\_LightNextTestDate,

l\_LightTestOverDue,

l\_EnvLastTestedDate,

l\_EnvNextTestDate,

l\_EnvTestOverDue,

L\_Equipinstid,

l\_LightingVoluntary );

l\_saved\_at\_equipment\_level := 2;

END LOOP;

CLOSE csr\_getSiteEquipment;

if ( l\_saved\_at\_equipment\_level = 1 ) then

-- Save detail record...

Insert\_NOCC\_Last\_Test\_Details( p\_DomainId,

p\_DomainName,

l\_SiteInstId,

l\_SiteName,

l\_SiteId,

l\_SiteType,

l\_SiteTechName,

l\_inventoried\_cells,

l\_NOCC\_Monitored,

l\_AntennaVZWOwned,

l\_LightingRequired,

l\_LightLastTestedDate,

l\_LightNextTestDate,

l\_LightTestOverDue,

l\_EnvLastTestedDate,

l\_EnvNextTestDate,

l\_EnvTestOverDue,

l\_EquipInstId,

l\_LightingVoluntary );

end if;

END LOOP;

-- Save domain level test summary/count record...

Insert\_Nocc\_Last\_Test\_Summary( --p\_Area,

--p\_Region,

P\_Territory,

P\_Market\_Territory,

P\_Sub\_Market,

p\_DomainId,

p\_DomainName,

l\_Site\_Count,

l\_HasAntennaTower\_Count,

l\_AntennaVZWOwned\_Count,

l\_LightingRequired\_Count,

l\_LightTestedDate\_Count,

l\_LightTestOverDue\_Count,

l\_LightTest\_OK\_Count,

l\_NOCC\_Monitored\_Count,

l\_EnvLastTestedDate\_Count,

l\_EnvTestOverDue\_Count,

l\_EnvTestOK\_Count );

CLOSE csr\_getAllSites;

EXCEPTION

WHEN OTHERS THEN

IF( csr\_getAllSites%ISOPEN ) THEN

CLOSE csr\_getAllSites;

END IF;

IF( csr\_getSiteEquipment%ISOPEN ) THEN

CLOSE csr\_getSiteEquipment;

END IF;

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in GetAllSites(DomainId='

|| p\_DomainId || ' ) -> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

/\*

FUNCTION getSiteTechName( p\_SiteInstId IN vzwnet.Site\_attr\_settings.Site\_Inst\_Id%TYPE )

RETURN vzwnet.Site\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Site\_attr\_settings.attr\_value%TYPE;

BEGIN

SELECT sas.attr\_value

INTO retval

FROM vzwnet.Site\_attr\_settings sas,

vzwnet.val\_attr\_Name van

WHERE sas.Site\_Inst\_Id = p\_SiteInstId

AND sas.val\_attr\_Inst\_Id = van.val\_attr\_Inst\_Id

AND upper(van.attr\_Name) = 'SITE TECHNICIAN'

AND upper(van.group\_Name) = 'EMERGENCY CONTACTS';

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN null;

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in getSiteTechName('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

\*/

FUNCTION getSiteTechName( p\_SiteInstId IN vzwnet.Site\_attr\_settings.Site\_Inst\_Id%TYPE )

RETURN vzwnet.Site\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Site\_attr\_settings.attr\_value%TYPE;

BEGIN

select

min(ri.name) name into retval

from

vzwnet.resource\_inst ri

join vzwnet.resource\_definition\_inst di

on di.DEFINITION\_INST\_ID = ri.DEFINITION\_INST\_ID

join vzwnet.resource\_associations ra

on ra.resource\_inst\_id = ri.resource\_inst\_id

where

ra.target\_type\_id = 16

and di.name = 'Operations'

and ri.category = 'Technician'

and ra.TARGET\_INST\_ID=p\_SiteInstId

group by ra.target\_inst\_id

;

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN null;

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in getSiteTechName('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getCellEquipDesc( p\_SiteInstId IN vzwnet.Site\_Inst.Site\_Inst\_Id%TYPE )

RETURN NOCC\_LAST\_TEST\_DETAILS.INVENTORIED\_CELLS%TYPE

IS

l\_Cell\_Equipment VARCHAR2(3099 );

l\_Equip\_descr VARCHAR2(255 );

CURSOR cursor\_Equipment\_desc( pp\_SiteInstId IN VARCHAR2 )

IS

SELECT NVL( ei.DESCR, 'none' )

FROM vzwnet.Equip\_Inst ei

WHERE ei.Site\_Inst\_Id = pp\_SiteInstId

AND ei.TYPE IN( SELECT xp\_Equip\_Type

FROM xng\_apps.VZ\_Ntwk\_Node\_Map

WHERE end\_point\_Id = 'CELL DEMARK' );

BEGIN

OPEN cursor\_Equipment\_desc( p\_SiteInstId );

LOOP

FETCH cursor\_Equipment\_desc INTO l\_Equip\_descr;

EXIT WHEN cursor\_Equipment\_desc%NOTFOUND;

IF ( l\_Cell\_Equipment IS NULL ) THEN

l\_Cell\_Equipment := l\_Equip\_descr;

ELSE

l\_Cell\_Equipment := l\_Cell\_Equipment || ';' || l\_Equip\_descr;

END IF;

END LOOP;

CLOSE cursor\_Equipment\_desc;

Return l\_Cell\_Equipment;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in loadEquipDesc('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

IF( cursor\_Equipment\_desc%ISOPEN ) THEN

CLOSE cursor\_Equipment\_desc;

END IF;

END;

FUNCTION getIsNOCCMonitored( p\_SiteInstId IN vzwnet.Site\_Inst.Site\_Inst\_Id%TYPE )

RETURN VARCHAR2

IS

retval VARCHAR2(25);

BEGIN

SELECT NVL(sas.Attr\_value, 'NO' )

INTO retval

FROM vzwnet.Site\_Attr\_settings sas,

vzwnet.val\_Attr\_Name van

WHERE sas.Site\_Inst\_Id = p\_SiteInstId

AND van.val\_Attr\_Inst\_Id = sas.val\_Attr\_Inst\_Id

AND UPPER(van.Attr\_Name ) = 'ENVIRONMENTALS NOCC MONITORED'

AND UPPER(van.group\_Name ) = 'SITE ALARM VERIFICATION TEST';

-- added 04/21/2006 JLA

IF upper(retval) != 'NO' THEN

retval := 'YES';

ELSE

retval := 'NO';

END IF;

--

RETURN trim( retval );

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'No';

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line( SUBSTR( 'Error, in getIsNoccMonitored('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) ||': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getFailedFirstEnvAttempt( p\_SiteInstId IN vzwnet.Site\_Inst.Site\_Inst\_Id%TYPE )

RETURN VARCHAR2

IS

retval NUMBER;

BEGIN

SELECT decode(upper(sas.Attr\_value),null,1,

'NO',2,

'YES',1,1)

INTO retval

FROM vzwnet.Site\_Attr\_settings sas,

vzwnet.val\_Attr\_Name van

WHERE sas.Site\_Inst\_Id = p\_SiteInstId

AND van.val\_Attr\_Inst\_Id = sas.val\_Attr\_Inst\_Id

AND UPPER(van.Attr\_Name ) = 'FAILED FIRST ENV ATTEMPT'

AND UPPER(van.group\_Name ) = 'SITE ALARM VERIFICATION TEST';

-- retval is a multiplier for 1 or 2 year Env Testing interval for CELL Sites

-- IF upper(retval) = 'NO' and retval is not null THEN

-- retval := 2;

-- ELSE

-- retval := 1;

-- END IF;

RETURN ( retval );

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 1;

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line( SUBSTR( 'Error, in getFailedFirstEnvAttempt('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) ||': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getEnvLastTestedDate(

p\_SiteInstId IN vzwnet.Site\_attr\_settings.Site\_Inst\_Id%TYPE,

p\_SiteType IN vzwnet.val\_attr\_group.class\_category%TYPE )

RETURN vzwnet.Site\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Site\_attr\_settings.attr\_value%TYPE;

BEGIN

SELECT sas.attr\_value

INTO retval

FROM vzwnet.Site\_attr\_settings sas,

vzwnet.val\_attr\_Name van,

vzwnet.val\_attr\_group vag

WHERE sas.Site\_Inst\_Id = p\_SiteInstId

AND van.val\_attr\_Inst\_Id = sas.val\_attr\_Inst\_Id

AND UPPER(van.attr\_Name ) = 'ENVIRONMENT DATE LAST TESTED'

AND UPPER(vag.group\_Name ) = UPPER(van.group\_Name )

AND vag.class\_category = p\_SiteType

AND vag.appl\_class = 'I'

AND UPPER(vag.group\_Name ) = 'SITE ALARM VERIFICATION TEST';

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN null;

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in getENVLastTestedDate('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getHasAntennaTower(

p\_SiteInstId IN vzwnet.Site\_Inst.Site\_Inst\_Id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.Equip\_Inst\_Id%TYPE )

RETURN VARCHAR2

IS

retval VARCHAR2(20);

BEGIN

SELECT NVL2(ei.Equip\_Inst\_Id, 'Has Tower', 'No Tower' ) -- some sites have multiple antennas

INTO retval

FROM vzwnet.Equip\_Inst ei,

vzwnet.Site\_Inst si

WHERE ei.Equip\_Inst\_Id = p\_EquipInstId

AND UPPER(ei.TYPE) = 'ANTENNA SUPPORT STRUCTURE'

AND si.Site\_inst\_id = ei.site\_inst\_id

AND si.Site\_Inst\_Id = p\_SiteInstId;

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'No Tower';

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in getHasAntennaTower('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getIsAntennaVZWOwned(

p\_SiteInstId IN vzwnet.Equip\_Inst.Site\_Inst\_Id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.equip\_inst\_id%TYPE )

RETURN vzwnet.Equip\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Equip\_attr\_settings.attr\_value%TYPE;

BEGIN

SELECT UPPER( eas.attr\_value )

INTO retval

FROM vzwnet.Equip\_Inst ei,

vzwnet.Equip\_attr\_settings eas,

vzwnet.val\_attr\_Name van

WHERE ei.Site\_Inst\_Id = p\_SiteInstId

AND ei.equip\_inst\_id = p\_EquipInstId

AND ei.type = 'ANTENNA SUPPORT STRUCTURE'

AND eas.Equip\_Inst\_Id = ei.Equip\_Inst\_Id

AND van.val\_attr\_Inst\_Id = eas.val\_attr\_Inst\_Id

AND UPPER(van.attr\_Name ) = 'OWNER'

AND UPPER(van.group\_Name ) = 'ANTENNA SUPPORT STRUCTURE';

-- added 04/21/2006 JLA

IF upper(retval) = 'OTHER' THEN

retval := 'OTHER';

ELSE

retval := 'VERIZON WIRELESS';

END IF;

--

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'VERIZON WIRELESS';

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line( SUBSTR( 'Error, in getIsAntennaVZWOwned('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getIsLightingRequired(

p\_SiteInstId IN vzwnet.Equip\_Inst.Site\_Inst\_Id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.equip\_inst\_id%TYPE )

RETURN vzwnet.Equip\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Equip\_attr\_settings.attr\_value%TYPE;

BEGIN

SELECT UPPER(eas.attr\_value)

INTO retval

FROM vzwnet.Equip\_Inst ei,

vzwnet.Equip\_attr\_settings eas,

vzwnet.val\_attr\_Name van

-- Krishna 8/4/2006 ,vzwnet.val\_attr\_group vag

WHERE ei.Site\_Inst\_Id = p\_SiteInstId

AND ei.equip\_inst\_id = p\_EquipInstId

AND eas.Equip\_Inst\_Id = ei.Equip\_Inst\_Id

AND van.val\_attr\_Inst\_Id = eas.val\_attr\_Inst\_Id

AND UPPER(van.attr\_Name ) = 'LIGHTING REQUIRED'

--Krishna 8/4/2006 AND UPPER(vag.group\_Name ) = UPPER(van.group\_Name )

AND UPPER(van.group\_Name ) = 'FAA INFORMATION';

-- added 04/21/2006 JLA

IF upper(retval) = 'NO' THEN

retval := 'NO';

ELSE

retval := 'YES';

END IF;

--

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'YES'; -- if its empty then we are treating it as Yes.

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in getIsLightingRequired('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getIsLightingVoluntary(

p\_SiteInstId IN vzwnet.Equip\_Inst.Site\_Inst\_Id%TYPE,

p\_EquipInstId IN vzwnet.Equip\_Inst.equip\_inst\_id%TYPE )

RETURN vzwnet.Equip\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Equip\_attr\_settings.attr\_value%TYPE;

BEGIN

SELECT UPPER(eas.attr\_value)

INTO retval

FROM vzwnet.Equip\_Inst ei,

vzwnet.Equip\_attr\_settings eas,

vzwnet.val\_attr\_Name van

-- Krishna 8/4/2006 ,vzwnet.val\_attr\_group vag

WHERE ei.Site\_Inst\_Id = p\_SiteInstId

AND ei.equip\_inst\_id = p\_EquipInstId

AND eas.Equip\_Inst\_Id = ei.Equip\_Inst\_Id

AND van.val\_attr\_Inst\_Id = eas.val\_attr\_Inst\_Id

AND UPPER(van.attr\_Name ) = 'VOLUNTARY LIGHTING'

AND UPPER(van.group\_Name ) = 'FAA INFORMATION';

IF upper(retval) = 'NO' THEN

retval := 'NO';

ELSE

retval := 'YES';

END IF;

--

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'NO'; -- if its empty then we are treating it as No.

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in getIsLightingVoluntary('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getLightingTestPeriod(P\_SITEINSTID IN VZWNET.EQUIP\_INST.SITE\_INST\_ID%TYPE,

P\_EQUIPINSTID IN VZWNET.EQUIP\_INST.EQUIP\_INST\_ID%TYPE

)

RETURN vzwnet.Equip\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Equip\_attr\_settings.attr\_value%TYPE;

BEGIN

SELECT UPPER(eas.attr\_value)

INTO retval

FROM vzwnet.Equip\_Inst ei,

vzwnet.Equip\_attr\_settings eas,

vzwnet.val\_attr\_Name van

WHERE ei.Site\_Inst\_Id = p\_SiteInstId

AND ei.equip\_inst\_id = p\_EquipInstId

AND eas.Equip\_Inst\_Id = ei.Equip\_Inst\_Id

AND VAN.VAL\_ATTR\_INST\_ID = EAS.VAL\_ATTR\_INST\_ID

AND UPPER(van.attr\_Name ) = 'LIGHTING TEST PERIOD'

AND UPPER(van.group\_Name ) = 'FAA INFORMATION';

IF UPPER(RETVAL) = 'ANNUALLY' THEN

retval := 'ANNUALLY';

ELSE

retval := 'QUARTERLY';

END IF;

RETURN retval;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'QUARTERLY'; -- if its empty then we are treating it as Quarterly.

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line(SUBSTR( 'Error, in getLightingTestPeriod('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

FUNCTION getTowerLightLastTestedDate(

p\_SiteInstId IN vzwnet.Site\_attr\_settings.Site\_Inst\_Id%TYPE,

p\_SiteType IN vzwnet.val\_attr\_group.class\_category%TYPE )

RETURN vzwnet.Site\_attr\_settings.attr\_value%TYPE

IS

retval vzwnet.Site\_attr\_settings.attr\_value%TYPE;

BEGIN

SELECT sas.attr\_value

INTO retval

FROM vzwnet.Site\_attr\_settings sas,

vzwnet.val\_attr\_Name van,

vzwnet.val\_attr\_group vag

WHERE sas.Site\_Inst\_Id = p\_SiteInstId

AND van.val\_attr\_Inst\_Id = sas.val\_attr\_Inst\_Id

AND UPPER(van.attr\_Name ) = 'TOWER LIGHT DATE LAST TESTED'

AND UPPER(vag.group\_Name ) = UPPER(van.group\_Name )

AND UPPER(vag.class\_category ) = p\_SiteType

AND UPPER(vag.group\_Name ) = 'SITE ALARM VERIFICATION TEST';

RETURN trim(retval);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN null;

WHEN OTHERS THEN

DBMS\_OUTPUT.put\_line( SUBSTR( 'Error, in getTowerLightLastTestedDate('

|| p\_SiteInstId || ' )-> '

|| TO\_CHAR(SQLCODE ) || ': '|| SQLERRM, 1, 255 ) );

END;

PROCEDURE insert\_NOCC\_Last\_Test\_Details(

p\_DomainId IN NOCC\_LAST\_TEST\_DETAILS.domain\_Id%TYPE,

p\_DomainName IN NOCC\_LAST\_TEST\_DETAILS.domain\_name%TYPE,

p\_SiteInstId IN NOCC\_LAST\_TEST\_DETAILS.site\_Inst\_Id%TYPE,

p\_SiteName IN NOCC\_LAST\_TEST\_DETAILS.site\_name%TYPE,

p\_SiteId IN NOCC\_LAST\_TEST\_DETAILS.site\_id%TYPE,

p\_Sitetype IN NOCC\_LAST\_TEST\_DETAILS.site\_type%TYPE,

p\_SiteTechName IN NOCC\_LAST\_TEST\_DETAILS.site\_tech\_name%TYPE,

p\_inventoried\_cells IN NOCC\_LAST\_TEST\_DETAILS.inventoried\_cells%TYPE,

p\_NOCC\_Monitored IN NOCC\_LAST\_TEST\_DETAILS.nocc\_Monitored%TYPE,

p\_vzw\_owned IN NOCC\_LAST\_TEST\_DETAILS.vzw\_owned%TYPE,

p\_lights\_reqd IN NOCC\_LAST\_TEST\_DETAILS.lights\_reqd%TYPE,

p\_LightLastTested IN NOCC\_LAST\_TEST\_DETAILS.Light\_last\_tested%TYPE,

p\_LightNextTest IN NOCC\_LAST\_TEST\_DETAILS.Light\_next\_test%TYPE,

p\_LightTestOverDue IN NOCC\_LAST\_TEST\_DETAILS.light\_test\_overdue%TYPE,

p\_ENVLastTested IN NOCC\_LAST\_TEST\_DETAILS.env\_last\_tested%TYPE,

p\_ENVNextTest IN NOCC\_LAST\_TEST\_DETAILS.env\_next\_test%TYPE,

p\_ENVTestOverDue IN NOCC\_LAST\_TEST\_DETAILS.env\_test\_overdue%TYPE,

p\_EquipInstid IN NOCC\_LAST\_TEST\_DETAILS.equip\_inst\_id%TYPE,

p\_LightsVoluntary IN NOCC\_LAST\_TEST\_DETAILS.lights\_voluntary%TYPE)

IS

BEGIN

INSERT INTO NOCC\_LAST\_TEST\_DETAILS ( Domain\_Id,

Domain\_Name,

Site\_Inst\_Id,

Site\_Name,

Site\_ID,

Site\_Type,

Site\_Tech\_Name,

inventoried\_cells,

NOCC\_Monitored,

vzw\_owned,

lights\_reqd,

Light\_Last\_Tested,

Light\_Next\_Test,

Light\_Test\_OverDue,

ENV\_Last\_Tested,

ENV\_Next\_Test,

ENV\_Test\_OverDue,

Run\_Date,

Equip\_inst\_id,

Lights\_Voluntary)

VALUES(

p\_DomainId,

p\_DomainName,

p\_SiteInstId,

p\_SiteName,

p\_SiteID,

p\_SiteType,

p\_SiteTechName,

p\_inventoried\_cells,

p\_NOCC\_Monitored,

p\_vzw\_owned,

p\_lights\_reqd,

p\_LightLastTested,

p\_LightNextTest,

p\_LightTestOverDue,

p\_ENVLastTested,

p\_ENVNextTest,

p\_ENVTestOverDue,

SYSDATE,

p\_EquipInstid,

p\_LightsVoluntary);

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line('Error Insert\_NOCC\_LAST\_TEST\_DETAILS()-> '

||' - '|| p\_DomainId

||' - '|| p\_DomainName

||' - '|| p\_SiteInstId

||' - '|| p\_SiteName

||' - '|| TO\_CHAR(SQLCODE));

END;

Procedure Insert\_Nocc\_Last\_Test\_Summary(

P\_Territory In Nocc\_Last\_Test\_Summary.Territory%Type,

P\_Market\_Territory In Nocc\_Last\_Test\_Summary.Market\_Territory%Type,

P\_Sub\_Market In Nocc\_Last\_Test\_Summary.Sub\_Market%Type,

P\_Domainid In Nocc\_Last\_Test\_Summary.Domain\_Id%Type,

P\_Domainname In Nocc\_Last\_Test\_Summary.Domain\_Name%Type,

P\_Sites In Nocc\_Last\_Test\_Summary.Sites%Type,

P\_Antennas In Nocc\_Last\_Test\_Summary.Antennas%Type,

P\_Vzwowned In Nocc\_Last\_Test\_Summary.Vzw\_Owned%Type,

P\_Lightsreqd In Nocc\_Last\_Test\_Summary.Lights\_Reqd%Type,

P\_Lightstested In Nocc\_Last\_Test\_Summary.Lights\_Tested%Type,

P\_Lightsoverdue In Nocc\_Last\_Test\_Summary.Lights\_Test\_Overdue%Type,

P\_Lightsok In Nocc\_Last\_Test\_Summary.Lights\_Test\_Ok%Type,

P\_Monitored In Nocc\_Last\_Test\_Summary.Nocc\_Monitored\_Sites%Type,

P\_Envtested In Nocc\_Last\_Test\_Summary.Env\_Tested%Type,

P\_Envoverdue In Nocc\_Last\_Test\_Summary.Env\_Test\_Overdue%Type,

P\_Envok In Nocc\_Last\_Test\_Summary.Env\_Test\_Ok%Type

)

IS

BEGIN

Insert Into Nocc\_Last\_Test\_Summary( --Area,

--Region,

Territory,

Market\_Territory,

Sub\_Market,

Domain\_Id,

Domain\_Name,

Sites,

Antennas,

VZW\_Owned,

Lights\_REQD,

Lights\_Tested,

Lights\_Test\_OverDue,

Lights\_Test\_Ok,

NOCC\_Monitored\_Sites,

ENV\_Tested,

ENV\_Test\_OverDue,

ENV\_Test\_Ok,

Run\_Date )

Values( --p\_Area,

--p\_Region,

P\_Territory,

P\_Market\_Territory,

P\_Sub\_Market,

p\_DomainId,

p\_DomainName,

p\_Sites,

p\_Antennas,

p\_VZWOwned,

p\_LightsReqd,

p\_LightsTested,

p\_LightsOverDue,

p\_LightsOk,

p\_Monitored,

p\_EnvTested,

p\_EnvOverDue,

p\_EnvOK,

SYSDATE );

EXCEPTION

When Others Then

Dbms\_Output.Put\_Line(Substr( P\_Territory

||' - '|| P\_Market\_Territory

||' - '|| P\_Sub\_Market

||' - '|| p\_DomainId

||' - '|| p\_DomainName

||' - ERROR: '|| TO\_CHAR(SQLCODE ) ||': '|| SQLERRM, 1, 255));

END;

FUNCTION varchar2date( p\_StrDate IN VARCHAR2 )

RETURN DATE IS

retval DATE := NULL;

BEGIN retval := TO\_DATE( rtrim(ltrim( p\_StrDate)), 'MM-DD-YY');

RETURN (retval);

EXCEPTION WHEN OTHERS THEN

BEGIN retval := TO\_DATE( rtrim(ltrim( p\_StrDate )), 'DD-MON-YY');

RETURN (retval);

EXCEPTION WHEN OTHERS THEN

BEGIN retval := NULL;

RETURN (retval);

END;

END;

END varchar2date;

PROCEDURE removeDetails( p\_DomainId IN NOCC\_LAST\_TEST\_SUMMARY.domain\_id%TYPE ) IS

BEGIN

-- Delete data from BOTH last tested reports tables...

DELETE FROM xng\_reports.NOCC\_LAST\_TEST\_DETAILS WHERE domain\_id = p\_DomainId;

DELETE FROM xng\_reports.NOCC\_LAST\_TEST\_SUMMARY WHERE domain\_id = p\_DomainId;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line('Error when deleting DOMIAN: '|| p\_DomainId );

RAISE;

END;

END; -- END of site\_tested package...

/

--------------------------------------------------------

-- DDL for Package Body SAM\_CSR\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."SAM\_CSR\_AUDIT" AS

PROCEDURE audit\_al\_csr\_equip

IS

methodname VARCHAR2 (30) := 'audit\_AL\_csr\_equip';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

-- verified with shirley --- assumption is they wont embed all csrs within a contianer of type csr. This logic will break if they so that.

-- does it matter if the the equipment name in GI does not match Std... verified with Shirley. yes it is fatal

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

BEGIN

EXECUTE IMMEDIATE 'truncate table SAM\_NT\_AL\_CSR\_AUDIT';

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t delete records from table: SAM\_NT\_AL\_CSR\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

DELETE FROM csr\_device\_audit\_issues iss

WHERE csr\_vendor = 'ALCATEL-LUCENT';

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t delete records from table: csr\_device\_audit\_issues';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

/\* audit csr devices from SAM vs Ng. Match them on device\_name \*/

BEGIN

sqlstmt :=

'insert into SAM\_NT\_AL\_CSR\_AUDIT

(csr\_vendor, csr\_device\_name

, sam\_csr\_device\_name, sam\_reachability, sam\_resync\_status, sam\_resync\_state, sam\_last\_resync\_start, sam\_last\_resync\_end

, NT\_DEVICE\_NAME, EQP\_REFERENCE\_ID

, Live\_in\_ng/\*Live\_in\_xng\*/, eq\_status

, match\_code

, match\_status

, eh\_csr\_hostname, eh\_csr\_ipaddress

, brix\_csr\_site

)

WITH sam\_csrs AS --Granite 44,685 / Nautilus 50,912

(

SELECT DISTINCT

upper(site\_name) site\_name --, act\_mgmt\_ip sam\_csr\_device\_ip

,

reachability sam\_reachability,

resync\_status sam\_resync\_status,

resync\_state sam\_resync\_state, /\*server\_name sam\_server\_name,\*/

last\_resyn\_start sam\_last\_resync\_start,

last\_resyn\_end sam\_last\_resync\_end

FROM

sam\_ebh\_health

WHERE

REGEXP\_LIKE ( site\_name,

''([A-Z|0-9]{8})T[0-9]{1}A-P-(AL|CI)-(G|[0-9]{1}|[0-9]{3})[0-9]{3}-(S|E|H|R|W|I|D|[0-9]{1})(I|[0-9]{1})([^\\[\\]]\*)?'' )---device\_name replaced by site\_name

),ng AS --Granite 49,003 / Nautilus 54,546

(

SELECT

csr.\*,

ei.inv\_status status

FROM

ntls\_csr\_parsed\_wk csr

JOIN ng\_reports.equipment ei---vzwnet.equip\_inst

ON ei.eqp\_reference\_id = csr.eqp\_reference\_id

WHERE

csr.csr\_device\_name LIKE ''%-AL-%''

),ehealth\_csr AS --Granite 0 / Nautilus 50,533

(

SELECT

upper(ecl.device\_name) hostname,

ecl.device\_ip

FROM

ng\_reports.sevone\_csr ecl---ehealth\_csr\_list replaced by sevone\_csr

WHERE

device\_name LIKE ''%-AL-%'' -- put this is a separate column. then the query will be faster

),brix\_csr AS --Granite 45,176 / Nautilus 97,548

(

SELECT

upper(brx.cell\_site\_name) cell\_site\_name

FROM

ng\_reports.brix\_csr\_discovery\_wk /\*ng\_reports.BRIX\_CSR\_EXTRACT\*/ brx---create script for BRIX\_CSR\_EXTRACT are listed above

WHERE

cell\_site\_name LIKE ''%-AL-%''

),nt\_sam\_aud AS --Granite 51,053 / Nautilus 62,487

(

SELECT DISTINCT

CASE

WHEN sam.site\_name IS NOT NULL THEN sam.site\_name

WHEN ng.csr\_device\_name IS NOT NULL THEN ng.csr\_device\_name

ELSE ''zzzz''

|| ng.eqp\_name /\*ng.descr\*/

END

csr\_device\_name,

site\_name sam\_csr\_device\_name,

sam\_reachability,

sam\_resync\_status,

sam\_resync\_state, /\*sam\_server\_name,\*/

sam\_last\_resync\_start,

sam\_last\_resync\_end,

ng.eqp\_name /\*ng.descr\*/ nt\_device\_name,

eqp\_reference\_id,

CASE

WHEN status = ''LIVE'' THEN ''Y''

WHEN status IS NOT NULL THEN ''N''

END

live\_in\_ng,

status,

CASE

WHEN sam.site\_name IS NOT NULL

AND ng.eqp\_reference\_id IS NOT NULL THEN ''BOTH''

-- it is in both sam extracts but not in Ng then NE Only

WHEN sam.site\_name IS NOT NULL

AND ng.eqp\_reference\_id IS NULL THEN ''NE Only''

WHEN sam.site\_name IS NULL

AND ng.eqp\_reference\_id IS NOT NULL THEN ''Nautilus Only''

ELSE ''XXXXX'' -- when does the control reach here

END

match\_code,

parse\_status match\_status

FROM

sam\_csrs sam

FULL OUTER JOIN ng ON ng.csr\_device\_name = sam.site\_name

) SELECT

''ALCATEL-LUCENT'' csr\_vendor, -- Granite 52,436 / Nautilus 113,059

CASE

WHEN ecsr.hostname IS NOT NULL THEN ecsr.hostname

WHEN brx.cell\_site\_name IS NOT NULL THEN brx.cell\_site\_name

WHEN gsf.csr\_device\_name IS NOT NULL THEN gsf.csr\_device\_name

END

csr\_device\_name,

sam\_csr\_device\_name,

sam\_reachability,

sam\_resync\_status,

sam\_resync\_state, /\*sam\_server\_name,\*/

sam\_last\_resync\_start,

sam\_last\_resync\_end,

nt\_device\_name,

eqp\_reference\_id,

live\_in\_ng,

status

-- it is in all 4 systems EH, SAM and GI, BRIX

,

CASE

WHEN gsf.match\_code = ''BOTH''

AND ecsr.hostname IS NOT NULL

AND brx.cell\_site\_name IS NOT NULL THEN ''BOTH''

-- it is in gi and sam but not in EH then NE\_ONLY

WHEN gsf.match\_code = ''BOTH''

AND (

ecsr.hostname IS NULL

OR brx.cell\_site\_name IS NULL

) THEN ''NE Only''

-- it is in gi and EH but not in SAM then NE\_ONLY

WHEN gsf.match\_code = ''Nautilus Only''

AND ecsr.hostname IS NOT NULL THEN ''NE Only''

ELSE gsf.match\_code

END

match\_code,

gsf.match\_status,

ecsr.hostname eh\_csr\_hostname,

ecsr.device\_ip /\*ecsr.ip\*/ eh\_csr\_ipaddress,

brx.cell\_site\_name

FROM

nt\_sam\_aud gsf

FULL OUTER JOIN ehealth\_csr ecsr ON gsf.csr\_device\_name = ecsr.hostname

FULL OUTER JOIN brix\_csr brx ON gsf.csr\_device\_name = brx.cell\_site\_name'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t insert insert AL csr audit into table: SAM\_NT\_AL\_CSR\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

BEGIN

-- if some ne\_hostname matched > 1 granite ne\_inst\_id then mark it dup

-- does stmt query work even when there are "Dup in NE"???

UPDATE SAM\_NT\_AL\_CSR\_AUDIT aud

SET match\_code = 'NE Only'

WHERE match\_code IS NULL;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t mark NE only for match code as blank';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

-- identify all issues with the audited equipment

insert\_al\_csr\_equip\_issues ();

assign\_area\_region\_to\_csr();

END audit\_al\_csr\_equip;

PROCEDURE insert\_al\_csr\_equip\_issues

IS

methodname VARCHAR2 (30) := 'insert\_AL\_csr\_equip\_issues';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

insertSqlStart VARCHAR2(100) := 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )';

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

-- CSR name in sam CSR Extract not Std Complaint --

/\* insert\_csr\_issues (processname , methodname, insertSqlStart ||

'select distinct site\_name, 16, ''ALCATEL-LUCENT'' from sam\_ebh\_health aud where parse\_status like ''%16%''',

'Can''t insert into csr\_device\_audit\_issues issue 16 ');

-- CSR name in eHealth Extract not Std Complaint --

insert\_csr\_issues (processname , methodname, insertSqlStart ||

'select distinct hostname, 25, ''ALCATEL-LUCENT''

from ehealth\_csr\_list

where hostname like ''%-AL-%'' /\* CHANGE THIS TO CSR\_VENDOR \*/

/\* and error\_code = 25 ',

'Can''t insert into csr\_device\_audit\_issues issue ehealth 25');

-- GI name not Std Complaint --

insert\_csr\_issues (processname , methodname, insertSqlStart ||

' select distinct csr\_device\_name, 7, ''ALCATEL-LUCENT''

from XNG\_CSR\_PARSED\_WK csr

where csr.csr\_device\_name like ''%-AL-%''

and parse\_status like ''%Xng CSR Name does not match Standard%''',

'Can''t insert into csr\_device\_audit\_issues issue issue 7');

insert\_csr\_issues (processname , methodname, insertSqlStart ||

'select distinct csr\_device\_name, 8, ''ALCATEL-LUCENT''

from XNG\_CSR\_PARSED\_WK csr

where csr\_device\_name like ''%-AL-%''

and parse\_status like ''%Invalid CSR Name%''',

'Can''t insert into csr\_device\_audit\_issues issue issue 8');

-- 12-14 issues in the previous are critical ONLY IF match\_code <> 'Xng Only'

-- the scheme needs to change somehow on the csr\_issues-------???????

-- device not found in sam csr extract\*/

sqlstmt:= 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 12, ''ALCATEL-LUCENT''

from SAM\_NT\_AL\_CSR\_AUDIT aud

where SAM\_CSR\_DEVICE\_NAME is null';

EXECUTE IMMEDIATE sqlstmt;

-- device not found in eHealth extract

sqlstmt:= 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 24, ''ALCATEL-LUCENT''

from SAM\_NT\_AL\_CSR\_AUDIT aud

where eh\_csr\_hostname is null';

EXECUTE IMMEDIATE sqlstmt;

-- device not found in eHealth extract

sqlstmt:= 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 33, ''ALCATEL-LUCENT''

from SAM\_NT\_AL\_CSR\_AUDIT aud

where brix\_csr\_site is null';

EXECUTE IMMEDIATE sqlstmt;

-- device not found in Xng

sqlstmt:= 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 15, ''ALCATEL-LUCENT''

from SAM\_NT\_AL\_CSR\_AUDIT aud

where eqp\_reference\_id is null';

EXECUTE IMMEDIATE sqlstmt;

-- DUP CSR match found in Xng

sqlstmt:= 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

select distinct csr\_device\_name, 1, ''ALCATEL-LUCENT''

from SAM\_NT\_AL\_CSR\_AUDIT i

where match\_code <> ''Nautilus Only''

group by i.csr\_device\_name

having count(eqp\_reference\_id) > 1';

EXECUTE IMMEDIATE sqlstmt;

-- DUPs found in NE. is this possible ????

-- mark 6-char match-not all- use from before

-- mark Invalid CLLI

-- We might have discovered some CLLI that is not in the list of MTSO CLLI

-- mark those as bad

-- We might have discovered some CLLI where 1st 6-char match MTSO CLLI but all 8 dont

-- mark those as bad

-- some times there are 2 legitimate cllis for the same site

-- in those cases the o/p comes as followd

-- clli\_1\_csr clli\_2\_vsm CLLI Mismatch

-- clli\_2\_csr clli\_1\_vsm CLLI Mismatch

-- so reverse the order and subtract it. That will give only the ones that are

-- truly bad

/\* want to be more specific to sam/sam VLAN and HPOV \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--sam issues 18 and 20

sqlstmt:= 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT ei.csr\_device\_name,

SUBSTR (ei.csr\_device\_name, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM SAM\_NT\_AL\_CSR\_AUDIT ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.csr\_device\_name, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.csr\_device\_name, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.csr\_device\_name

FROM SAM\_NT\_AL\_CSR\_AUDIT ei

MINUS

SELECT ei.csr\_device\_name

FROM SAM\_NT\_AL\_CSR\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.csr\_device\_name, 1, 8)

MINUS

SELECT ei.csr\_device\_name

FROM SAM\_NT\_AL\_CSR\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.csr\_device\_name, 1, 6)

),

ISSUES AS

(

SELECT cm.csr\_device\_name, 20 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.csr\_device\_name, 18 issue\_id

FROM bad\_clli bc

)

SELECT distinct WK.CSR\_DEVICE\_NAME , i.issue\_id, ''ALCATEL-LUCENT''

FROM ISSUES i, SAM\_NT\_AL\_CSR\_AUDIT wk

where i.csr\_device\_name = Wk.csr\_device\_name';

EXECUTE IMMEDIATE sqlstmt;

--eHealth issues 26 and 27

sqlstmt:= 'insert into csr\_device\_audit\_issues columns (csr\_device\_name, csr\_issue\_id, csr\_vendor )

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT ei.eh\_csr\_hostname,

SUBSTR (ei.eh\_csr\_hostname, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM SAM\_NT\_AL\_CSR\_AUDIT ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.csr\_device\_name, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.eh\_csr\_hostname, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.eh\_csr\_hostname

FROM SAM\_NT\_AL\_CSR\_AUDIT ei

MINUS

SELECT ei.eh\_csr\_hostname

FROM SAM\_NT\_AL\_CSR\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.eh\_csr\_hostname, 1, 8)

MINUS

SELECT ei.eh\_csr\_hostname

FROM SAM\_NT\_AL\_CSR\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.eh\_csr\_hostname, 1, 6)

),

ISSUES AS

(

SELECT cm.eh\_csr\_hostname, 27 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.eh\_csr\_hostname, 26 issue\_id

FROM bad\_clli bc

)

SELECT distinct WK.CSR\_DEVICE\_NAME , i.issue\_id, ''ALCATEL-LUCENT''

FROM ISSUES i, SAM\_NT\_AL\_CSR\_AUDIT wk

where i.eh\_csr\_hostname = Wk.csr\_device\_name';

EXECUTE IMMEDIATE sqlstmt;

BEGIN

-- if some ne\_hostname matched > 1 granite ne\_inst\_id then mark it dup

-- does stmt query work even when there are "Dup in NE"???

UPDATE SAM\_NT\_AL\_CSR\_AUDIT aud

SET match\_code = 'NE Only'

WHERE EXISTS (

SELECT 1

FROM csr\_device\_audit\_issues i

WHERE aud.csr\_device\_name = i.csr\_device\_name

AND i.csr\_vendor = 'ALCATEL-LUCENT'

GROUP BY i.csr\_device\_name)

AND match\_code = 'BOTH';

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update NE only if there is any issue';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

END insert\_al\_csr\_equip\_issues;

PROCEDURE assign\_area\_region\_to\_csr

IS

methodname VARCHAR2 (30) := 'assign\_area\_region\_to\_CSR';

MESSAGE VARCHAR2 (300);

updstmt VARCHAR2 (32767);

--In Granite there are 48,947 distinct csr\_device\_name, nautilus has 62,319. Nautilus has alot of dups.

--Granite has no records in the sam\_NT\_al\_csr\_audit\_wk table only in the non-wk table

--sam\_NT\_al\_csr\_audit: 49,147 / SAM\_NT\_AL\_CSR\_AUDIT: 113,440

CURSOR area\_region\_dev

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS')--Granite 609 / Nautilus 608

SELECT di.territory, di.market\_territory, di.sub\_market, di.sub\_market\_leaf, di.leaf\_domain\_id,

csr\_device\_name, clli\_6

FROM SAM\_NT\_AL\_CSR\_AUDIT aud JOIN domains di

ON di.clli\_6 = SUBSTR (aud.csr\_device\_name, 1, 6)

;--Granite 49,068 non-wk / Nautilus 113,385

CURSOR eq\_area\_region\_dev

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS')--Granite 609 / Nautilus 608

SELECT di.territory, di.market\_territory, di.sub\_market, di.sub\_market\_leaf, di.leaf\_domain\_id,

csr\_device\_name, clli\_6

FROM SAM\_NT\_AL\_CSR\_AUDIT aud JOIN ng\_reports.EQUIPMENT\_DOMAIN\_MAP edm--- ref vzwnet.equip\_domain\_map

ON edm.EQP\_REFERENCE\_ID = aud.EQP\_REFERENCE\_ID---ref EQUIP\_INST\_ID

JOIN domains di ON di.leaf\_domain\_id = edm.domain\_id

WHERE aud.territory IS NULL

;--Granite 0 non-wk / Nautilus 2,031,134

CURSOR unknown\_domain

IS

SELECT dlr.territory AS territory,dlr.market\_territory AS market\_territory, dlr.sub\_market AS sub\_market, dlr.sub\_market\_leaf AS sub\_market\_leaf,

dlr.domain\_id AS domain\_id,

sgalwk.csr\_device\_name AS csr\_device\_name

FROM ng\_reports.EQUIPMENT equip,

SAM\_NT\_AL\_CSR\_AUDIT sgalwk,

ng\_reports.EQUIPMENT\_DOMAIN\_MAP equipmap,---ref vzwnet.equip\_domain\_map

ng\_reports.domains\_leaf\_reporting dlr

WHERE sgalwk.leaf\_domain\_id = 0

AND equip.eqp\_name = sgalwk.sam\_csr\_device\_name

AND equipmap.EQP\_REFERENCE\_ID = equip.EQP\_REFERENCE\_ID---EQP\_inst\_ID

AND dlr.domain\_id = equipmap.domain\_id

;--Granite 0 / Nautilus 0

BEGIN

updstmt :=

'update SAM\_NT\_AL\_CSR\_AUDIT set territory = :territory, market\_territory = :market\_territory

, sub\_market = :sub\_market, market = :sub\_market\_leaf

, leaf\_domain\_id = :leaf\_domain\_id---ref leaf\_domain\_id

, clli = :clli\_6

where csr\_device\_name = :csr\_device\_name';

FOR rec IN area\_region\_dev

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.leaf\_domain\_id,---ref leaf\_domain\_id

rec.clli\_6,

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market for '

|| rec.csr\_device\_name

|| ' in table: SAM\_NT\_AL\_CSR\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

-- area region for put xng\_only based on the area/region of ne\_inst\_id???????????

FOR rec IN eq\_area\_region\_dev

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.leaf\_domain\_id,

rec.clli\_6,

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from Ng Only for '

|| rec.csr\_device\_name

|| ' in table: SAM\_NT\_AL\_CSR\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

BEGIN

-- if there still are some devices which cant be mapped to any domain then mark those as unknown

updstmt :=

' update SAM\_NT\_AL\_CSR\_AUDIT

set territory = ''unknown''

, market\_territory = ''unknown''

, sub\_market = ''unknown''

, market = ''unknown''

, leaf\_domain\_id = 0---LEAF\_DOMAIN\_ID

, clli = ''unknown''

where territory is null ';

EXECUTE IMMEDIATE updstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from to unknown in table: SAM\_NT\_AL\_CSR\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

updstmt :=

' update SAM\_NT\_AL\_CSR\_AUDIT

set territory = :territory, market\_territory = :market\_territory

, sub\_market = :sub\_market, market = :market

, leaf\_domain\_id = :DOMAIN\_ID

where csr\_device\_name = :csr\_device\_name ';

FOR rec IN unknown\_domain

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.domain\_id,---LEAF\_DOMAIN\_ID

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update territory/market\_territory/sub\_market from Ng Only for '

|| rec.csr\_device\_name

|| ' in table: SAM\_NT\_AL\_CSR\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

BEGIN

-- if some ne\_hostname matched > 1 granite ne\_inst\_id then mark it dup

-- does stmt query work even when there are "Dup in NE"???

UPDATE SAM\_NT\_AL\_CSR\_AUDIT aud

SET clli = SUBSTR (aud.csr\_device\_name, 1, 6)

WHERE clli = 'unknown';

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname

|| '(): Can''t update clli for unknown';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

END assign\_area\_region\_to\_csr;

PROCEDURE load\_csr\_clli\_summary

IS

processname VARCHAR2 (30) := 'SAM\_NT\_CSR\_AUDIT';

-- get the all CSRs found in sam CSR Extract

CURSOR csrs\_sam

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS')

,sam AS

(

SELECT DISTINCT site\_name, SUBSTR (site\_name, 1, 6) clli\_6

FROM sam\_ebh\_health sam

WHERE REGEXP\_LIKE (SITE\_NAME, '([A-Z|0-9]{8})T[0-9]{1}A-P-(AL|CI)-(G|[0-9]{1}|[0-9]{3})[0-9]{3}-(S|E|H|R|W|I|D|[0-9]{1})(I|[0-9]{1})([^\\[\\]]\*)?'))

SELECT NVL (territory, 'unknown') territory, NVL (market\_territory, 'unknown') market\_territory, NVL (sub\_market, 'unknown') sub\_market,

NVL (sub\_market\_leaf, 'unknown') sub\_market\_leaf,

NVL (di.leaf\_domain\_id, 0) LEAF\_DOMAIN\_ID ,

sam.clli\_6, COUNT (1) sam\_csrs

FROM domains di RIGHT OUTER JOIN sam ON sam.clli\_6 = di.clli\_6

GROUP BY territory, market\_territory, sub\_market, sub\_market\_leaf, di.leaf\_domain\_id, sam.clli\_6;

-- get the all CSRs found in EH

CURSOR csrs\_eh

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS'),

ehealth AS

(SELECT DISTINCT SUBSTR (device\_name, 1, 6) clli\_6, device\_name

FROM ng\_reports.SEVONE\_CSR hp

WHERE DEVICE\_NAME LIKE '%-AL-%')------hostname REPLACED BY DEVICE\_NAME

SELECT NVL (territory, 'unknown') territory, NVL (market\_territory, 'unknown') market\_territory, NVL (sub\_market, 'unknown') sub\_market,

NVL (sub\_market\_leaf, 'unknown') sub\_market\_leaf,

NVL (di.leaf\_domain\_id, 0) LEAF\_DOMAIN\_ID ,

eh.clli\_6, COUNT (1) eh\_csrs

FROM domains di RIGHT OUTER JOIN ehealth eh ON eh.clli\_6 =

di.clli\_6

GROUP BY territory, market\_territory, sub\_market, sub\_market\_leaf, di.leaf\_domain\_id, eh.clli\_6;

-- get the all CSRs found in BRIX

CURSOR csrs\_brix

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS'),

brix AS

(SELECT DISTINCT SUBSTR (cell\_site\_name, 1, 6) clli\_6, cell\_site\_name

FROM BRIX\_CSR\_DISCOVERY\_WK

WHERE cell\_site\_name LIKE '%-AL-%')

SELECT NVL (territory, 'unknown') territory, NVL (market\_territory, 'unknown') market\_territory, NVL (sub\_market, 'unknown') sub\_market,

NVL (sub\_market\_leaf, 'unknown') sub\_market\_leaf,

NVL (di.leaf\_domain\_id, 0) LEAF\_DOMAIN\_ID ,

b.clli\_6, COUNT (1) brix\_csrs

FROM domains di RIGHT OUTER JOIN brix b ON b.clli\_6 = di.clli\_6

GROUP BY territory, market\_territory, sub\_market, sub\_market\_leaf, di.LEAF\_DOMAIN\_ID , b.clli\_6;

-- ALL issues, critical or not

CURSOR csrs\_w\_issues

IS

WITH domains AS

(SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6, territory, market\_territory, sub\_market,

sub\_market\_leaf, leaf\_domain\_id

FROM clli\_domain\_map\_v

WHERE sub\_market <> 'NNO' AND sub\_market <> 'OSS'),

csr\_w\_issues AS

(SELECT DISTINCT cai.csr\_device\_name,

SUBSTR (cai.csr\_device\_name, 1, 6) clli\_6

FROM csr\_device\_audit\_issues cai

WHERE cai.csr\_vendor = 'ALCATEL-LUCENT')

SELECT NVL (territory, 'unknown') territory, NVL (market\_territory, 'unknown') market\_territory, NVL (sub\_market, 'unknown') sub\_market,

NVL (sub\_market\_leaf, 'unknown') sub\_market\_leaf,

NVL (di.LEAF\_DOMAIN\_ID , 0) LEAF\_DOMAIN\_ID ,

sam.clli\_6, COUNT (1) sam\_eh\_nt\_issues

FROM domains di RIGHT OUTER JOIN csr\_w\_issues sam

ON sam.clli\_6 = di.clli\_6

GROUP BY territory, market\_territory, sub\_market, sub\_market\_leaf, di.LEAF\_DOMAIN\_ID , sam.clli\_6

ORDER BY territory, market\_territory, sub\_market;

-- make sure duplicate matches in GI are not considered as good match-------??????

-- DUP means > 1 LIVE ones in Ng

-- if 1 is LIVE and others are not, then that is NOT a DIP. Shirley's logic

-- this should be identifiable from the audit

-- get mismatches between sam-csr-vlan extract and gi

CURSOR sam\_NT\_match\_csrs

IS

WITH sam\_NT\_match AS

(SELECT csr\_device\_name,

CASE

WHEN live\_in\_ng = 'Y'

THEN 1

ELSE 0

END sam\_NT\_l\_match,

CASE

WHEN live\_in\_ng <> 'Y'

THEN 1

ELSE 0

END sam\_NT\_nl\_match

FROM SAM\_NT\_AL\_CSR\_AUDIT aud

WHERE aud.sam\_csr\_device\_name IS NOT NULL

AND aud.NT\_DEVICE\_NAME IS NOT NULL

MINUS

-- DUPS are NOT good

SELECT aud.csr\_device\_name,

CASE

WHEN live\_in\_ng = 'Y'

THEN 1

ELSE 0

END sam\_NT\_l\_match,

CASE

WHEN live\_in\_ng <> 'Y'

THEN 1

ELSE 0

END sam\_NT\_nl\_match

FROM SAM\_NT\_AL\_CSR\_AUDIT aud JOIN csr\_device\_audit\_issues cdai

ON cdai.csr\_device\_name = aud.csr\_device\_name

WHERE csr\_issue\_id = 1 AND match\_code <> 'Nautilus Only')

SELECT CASE

WHEN ma.csr\_device\_name IS NOT NULL

THEN SUBSTR (ma.csr\_device\_name, 1, 6)

END clli\_6,

COUNT (ma.csr\_device\_name) sam\_NT\_match,

SUM (ma.sam\_NT\_l\_match) sam\_NT\_l\_match,

SUM (ma.sam\_NT\_nl\_match) sam\_NT\_nl\_match

FROM sam\_NT\_match ma

GROUP BY CASE

WHEN ma.csr\_device\_name IS NOT NULL

THEN SUBSTR (ma.csr\_device\_name, 1, 6)

END;

CURSOR eh\_NT\_match\_csrs

IS

WITH eh\_NT\_match AS

(SELECT csr\_device\_name

,

CASE

WHEN live\_in\_ng = 'Y'

THEN 1

ELSE 0

END eh\_NT\_l\_match,

CASE

WHEN live\_in\_ng <> 'Y'

THEN 1

ELSE 0

END eh\_NT\_nl\_match

FROM SAM\_NT\_AL\_CSR\_AUDIT aud

WHERE aud.eh\_csr\_hostname IS NOT NULL

AND aud.NT\_DEVICE\_NAME IS NOT NULL

MINUS

-- DUPS are NOT good

SELECT aud.csr\_device\_name,

CASE

WHEN live\_in\_ng = 'Y'

THEN 1

ELSE 0

END sam\_NT\_l\_match,

CASE

WHEN live\_in\_ng <> 'Y'

THEN 1

ELSE 0

END sam\_NT\_nl\_match

FROM SAM\_NT\_AL\_CSR\_AUDIT aud JOIN csr\_device\_audit\_issues cdai

ON cdai.csr\_device\_name = aud.csr\_device\_name

WHERE csr\_issue\_id = 1 AND match\_code <> 'Nautilus Only')

SELECT CASE

WHEN ma.csr\_device\_name IS NOT NULL

THEN SUBSTR (ma.csr\_device\_name, 1, 6)

END clli\_6,

COUNT (ma.csr\_device\_name) eh\_NT\_match,

SUM (ma.eh\_NT\_l\_match) eh\_NT\_l\_match,

SUM (ma.eh\_NT\_nl\_match) eh\_NT\_nl\_match

FROM eh\_NT\_match ma

GROUP BY CASE

WHEN ma.csr\_device\_name IS NOT NULL

THEN SUBSTR (ma.csr\_device\_name, 1, 6)

END;

-- make sure duplicate matches in GI are not considered as good match-------??????

-- DUP means > 1 LIVE ones in Ng

-- if 1 is LIVE and others are not, then that is NOT a DUP. Shirley's logic

-- this should be identifiable from the audit

CURSOR brix\_NT\_match\_csrs

IS

WITH brix\_NT\_match AS

(SELECT csr\_device\_name --, match\_status, live\_in\_ng

,

CASE

WHEN live\_in\_ng = 'Y'

THEN 1

ELSE 0

END brix\_NT\_l\_match,

CASE

WHEN live\_in\_ng <> 'Y'

THEN 1

ELSE 0

END brix\_NT\_nl\_match

FROM SAM\_NT\_AL\_CSR\_AUDIT aud

WHERE aud.brix\_csr\_site IS NOT NULL

AND aud.NT\_DEVICE\_NAME IS NOT NULL

MINUS

-- DUPS are NOT good

SELECT aud.csr\_device\_name,

CASE

WHEN live\_in\_ng = 'Y'

THEN 1

ELSE 0

END brix\_NT\_l\_match,

CASE

WHEN live\_in\_ng <> 'Y'

THEN 1

ELSE 0

END brix\_NT\_nl\_match

FROM SAM\_NT\_AL\_CSR\_AUDIT aud JOIN csr\_device\_audit\_issues cdai

ON cdai.csr\_device\_name = aud.csr\_device\_name

WHERE csr\_issue\_id = 1 AND match\_code <> 'XnXXxg Only')

SELECT CASE

WHEN ma.csr\_device\_name IS NOT NULL

THEN SUBSTR (ma.csr\_device\_name, 1, 6)

END clli\_6,

COUNT (ma.csr\_device\_name) brix\_NT\_match,

SUM (ma.brix\_NT\_l\_match) brix\_NT\_l\_match,

SUM (ma.brix\_NT\_nl\_match) brix\_NT\_nl\_match

FROM brix\_NT\_match ma

GROUP BY CASE

WHEN ma.csr\_device\_name IS NOT NULL

THEN SUBSTR (ma.csr\_device\_name, 1, 6)

END;

CURSOR sam\_NT\_mis\_match\_csrs

IS

WITH sam\_NT\_mis\_match AS

(SELECT DISTINCT SUBSTR (csr\_device\_name, 1, 6) clli\_6, csr\_device\_name , match\_code, live\_in\_ng

FROM SAM\_NT\_AL\_CSR\_AUDIT aud

WHERE (( aud.sam\_csr\_device\_name IS NULL

AND aud.NT\_DEVICE\_NAME IS NOT NULL

)

OR ( aud.NT\_DEVICE\_NAME IS NULL

AND aud.sam\_csr\_device\_name IS NOT NULL

))

AND match\_code <> 'BOTH'

)

select clli\_6, sum(cc) sam\_NT\_mis\_match from (SELECT CASE

WHEN csr\_device\_name IS NOT NULL and match\_code = 'NE Only' or csr\_device\_name IS NOT NULL and live\_in\_ng = 'Y'

THEN 1 else 0

END cc,

csr\_device\_name, clli\_6

FROM sam\_NT\_mis\_match )

GROUP BY clli\_6;

CURSOR eh\_NT\_mis\_match\_csrs

IS

WITH eh\_NT\_mis\_match AS

(SELECT DISTINCT SUBSTR (csr\_device\_name, 1, 6) clli\_6, csr\_device\_name , match\_code, live\_in\_ng

FROM SAM\_NT\_AL\_CSR\_AUDIT aud

WHERE (( aud.eh\_csr\_hostname IS NULL

AND aud.NT\_DEVICE\_NAME IS NOT NULL

)

OR ( aud.NT\_DEVICE\_NAME IS NULL

AND aud.eh\_csr\_hostname IS NOT NULL

))

AND match\_code <> 'BOTH'

)

select clli\_6, sum(cc) eh\_NT\_mis\_match from (SELECT CASE

WHEN csr\_device\_name IS NOT NULL and match\_code = 'NE Only' or csr\_device\_name IS NOT NULL and live\_in\_ng = 'Y'

THEN 1 else 0

END cc,

csr\_device\_name, clli\_6

FROM eh\_NT\_mis\_match )

GROUP BY clli\_6;

CURSOR brix\_NT\_mis\_match\_csrs

IS

WITH brix\_NT\_mis\_match AS

(SELECT DISTINCT SUBSTR (csr\_device\_name, 1, 6) clli\_6, csr\_device\_name , match\_code, live\_in\_ng

FROM SAM\_NT\_AL\_CSR\_AUDIT aud

WHERE (( aud.brix\_csr\_site IS NULL

AND aud.NT\_DEVICE\_NAME IS NOT NULL

)

OR ( aud.NT\_DEVICE\_NAME IS NULL

AND aud.brix\_csr\_site IS NOT NULL

))

AND match\_code <> 'BOTH'

)

select clli\_6, sum(cc) brix\_NT\_mis\_match from (SELECT CASE

WHEN csr\_device\_name IS NOT NULL and match\_code = 'NE Only' or csr\_device\_name IS NOT NULL and live\_in\_ng = 'Y'

THEN 1 else 0

END cc,

csr\_device\_name, clli\_6

FROM brix\_NT\_mis\_match )

GROUP BY clli\_6;

methodname VARCHAR2 (30) := 'load\_csr\_clli\_summary';

MESSAGE VARCHAR2 (300);

updstmt VARCHAR2 (32767);

insstmt VARCHAR2 (32767);

sqlstmt VARCHAR2 (32767);

l\_cnt NUMBER;

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

EXECUTE IMMEDIATE 'TRUNCATE TABLE SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

-- from ng insert all CLLI and ng# in the table

sqlstmt :=

'insert into SAM\_EH\_NT\_CSR\_CLLI\_SUMM columns (territory, market\_territory, sub\_market, market, LEAF\_DOMAIN\_ID , clli\_6, NT\_csrs, nt\_l\_csrs, NT\_nl\_csrs)

WITH

DOMAINS AS

(

select distinct substr(clli, 1, 6) clli\_6, territory, market\_territory, sub\_market, sub\_market\_leaf, leaf\_domain\_id

from clli\_domain\_map\_v

where sub\_market <> ''NNO''

and sub\_market <> ''OSS''

)

,

-- get all devices from Ng along with LIVE and NL status

GI as

(

select distinct csr\_device\_name, substr(csr\_device\_name, 1, 6) clli\_6

, nvl(case when ei.inv\_status = ''LIVE'' then 1 else 0 end, 0) l

, nvl(case when ei.inv\_status <> ''LIVE'' then 1 else 0 end, 0) nl

from NTLS\_CSR\_PARSED\_WK csr join ng\_reports.EQUIPMENT ei---vzwnet.equip\_inst

on ei.EQP\_REFERENCE\_ID = csr.EQP\_REFERENCE\_ID

where csr.csr\_device\_name like ''%-AL-%''

)

,

NT\_CLLI\_DOMAIN as

(

-- determine the ng device domain based on CLLI

select nvl(di.territory, ''unknown'') territory, nvl(di.market\_territory, ''unknown'') market\_territory

, nvl(di.sub\_market, ''unknown'') sub\_market, nvl(di.sub\_market\_leaf, ''unknown'') sub\_market\_leaf

, nvl(di.leaf\_domain\_id, 0) LEAF\_DOMAIN\_ID

, ng.clli\_6

, ng.csr\_device\_name

, ng.l

, ng.nl

from DOMAINS di

join GI ng

on ng.clli\_6 = di.clli\_6

)

,

NT\_EQ\_NO\_CLLI\_DOMAIN as

(

-- get all devices for which domain cant be found based on CLLI

select ng.clli\_6

, ng.csr\_device\_name

, l

, nl

from GI ng

minus

-- determine the device domain based on CLLI

select ng.clli\_6

, ng.csr\_device\_name

, l

, nl

from NT\_CLLI\_DOMAIN ng

)

,

NT\_EQ\_W\_NT\_DOMAIN as

(

-- for those devices get the domain from Ng

select nvl(dlr.territory, ''unknown'') territory, nvl(dlr.market\_territory, ''unknown'') market\_territory

, nvl(dlr.sub\_market, ''unknown'') sub\_market, nvl(dlr.sub\_market\_leaf, ''unknown'') sub\_market\_leaf

, nvl(dlr.domain\_id, 0) LEAF\_DOMAIN\_ID

, ng.clli\_6

, ng.csr\_device\_name

, l

, nl

from NT\_EQ\_NO\_CLLI\_DOMAIN ng

join ntls\_csr\_parsed\_WK xcp

on xcp.CSR\_DEVICE\_NAME = ng.csr\_device\_name

join ng\_reports.EQUIPMENT\_DOMAIN\_MAP edm

on xcp.EQP\_REFERENCE\_ID = edm.EQP\_REFERENCE\_ID

join domains\_leaf\_reporting dlr

on dlr.domain\_id = edm.DOMAIN\_ID

)

,

ALL\_NT\_EQUIP as

(

select territory, market\_territory, sub\_market, sub\_market\_leaf, LEAF\_DOMAIN\_ID

, ng.clli\_6

, ng.csr\_device\_name

, l

, nl

from NT\_CLLI\_DOMAIN ng

union all

select territory, market\_territory, sub\_market, sub\_market\_leaf, LEAF\_DOMAIN\_ID

, ng.clli\_6

, ng.csr\_device\_name

, l

, nl

from NT\_EQ\_W\_NT\_DOMAIN ng

)

-- determine the device domain based on CLLI

select gi.territory, gi.market\_territory, gi.sub\_market, gi.sub\_market\_leaf, gi.LEAF\_DOMAIN\_ID

, gi.clli\_6

, count (gi.csr\_device\_name) NT\_csrs

, sum (l) NT\_l\_csr

, sum (nl) NT\_nl\_csr

from ALL\_NT\_EQUIP gi

group by gi.territory, gi.market\_territory, gi.sub\_market, gi.sub\_market\_leaf, gi.LEAF\_DOMAIN\_ID

, gi.clli\_6

order by gi.territory, gi.market\_territory, gi.sub\_market, gi.sub\_market\_leaf, gi.LEAF\_DOMAIN\_ID

, gi.clli\_6

';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

-- insert new or update metric in the same table for total\_csrs found in sam in the csr-extract

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set sam\_csrs = :sam\_csrs

where clli\_6 = :clli\_6

';

insstmt :=

'insert into SAM\_EH\_NT\_CSR\_CLLI\_SUMM columns (territory, market\_territory, sub\_market, market, LEAF\_DOMAIN\_ID , clli\_6, sam\_csrs)

values (:territory, :market\_territory, :sub\_market, :market, :LEAF\_DOMAIN\_ID , :clli\_6, :sam\_csrs)

';

FOR rec IN csrs\_sam

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.sam\_csrs, rec.clli\_6;

l\_cnt := SQL%ROWCOUNT;

IF (l\_cnt = 0)

THEN

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

DBMS\_OUTPUT.put\_line ( 'FYI: '

|| l\_cnt

|| ' matching rows for sam CLLI'

);

DBMS\_OUTPUT.put\_line ( 'FYI: should insert in sam '

|| rec.territory

|| ' '

|| rec.market\_territory

|| ' '

|| rec.sub\_market

|| ' '

|| rec.sub\_market\_leaf

|| ' '

|| rec.LEAF\_DOMAIN\_ID

|| ' '

|| rec.clli\_6

|| ' '

|| rec.sam\_csrs

);

BEGIN

EXECUTE IMMEDIATE insstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.LEAF\_DOMAIN\_ID ,

rec.clli\_6,

rec.sam\_csrs;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t insert '

|| rec.clli\_6

|| ' sam only CSR extract CLLI in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

--RAISE;

END;

END IF;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update '

|| rec.clli\_6

|| ' for sam-CSR-extract in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

COMMIT;

-- insert new or update metric in the same table for ehealth total\_csrs

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set eh\_csrs = :eh\_csrs

where clli\_6 = :clli\_6

';

insstmt :=

'insert into SAM\_EH\_NT\_CSR\_CLLI\_SUMM columns (territory, market\_territory, sub\_market, market, LEAF\_DOMAIN\_ID , clli\_6, eh\_csrs)

values (:territory, :market\_territory, :sub\_market, :market, :LEAF\_DOMAIN\_ID , :clli\_6, :eh\_csrs)

';

FOR rec IN csrs\_eh

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.eh\_csrs, rec.clli\_6;

l\_cnt := SQL%ROWCOUNT;

IF (l\_cnt = 0)

THEN

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

DBMS\_OUTPUT.put\_line ( 'FYI: '

|| l\_cnt

|| ' matching rows for sam CLLI'

);

DBMS\_OUTPUT.put\_line ( 'FYI: should insert in ehealth '

|| rec.territory

|| ' '

|| rec.market\_territory

|| ' '

|| rec.sub\_market

|| ' '

|| rec.sub\_market\_leaf

|| ' '

|| rec.LEAF\_DOMAIN\_ID

|| ' '

|| rec.clli\_6

|| ' '

|| rec.eh\_csrs

);

BEGIN

EXECUTE IMMEDIATE insstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.LEAF\_DOMAIN\_ID ,

rec.clli\_6,

rec.eh\_csrs;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t insert '

|| rec.clli\_6

|| ' sam-CSR-VLAN-extract CLLI in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

--RAISE;

END;

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update '

|| rec.clli\_6

|| ' for sam-CSR-VLAN-extract in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

END LOOP;

-- insert new or update metric in the same table for brix total\_csrs

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set brix\_csrs = :brix\_csrs

where clli\_6 = :clli\_6

';

insstmt :=

'insert into SAM\_EH\_NT\_CSR\_CLLI\_SUMM columns (territory, market\_territory, sub\_market, market, LEAF\_DOMAIN\_ID , clli\_6, brix\_csrs)

values (:territory, :market\_territory, :sub\_market, :market, :LEAF\_DOMAIN\_ID , :clli\_6, :brix\_csrs)

';

FOR rec IN csrs\_brix

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.brix\_csrs, rec.clli\_6;

l\_cnt := SQL%ROWCOUNT;

IF (l\_cnt = 0)

THEN

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

DBMS\_OUTPUT.put\_line ( 'FYI: '

|| l\_cnt

|| ' matching rows for sam CLLI'

);

DBMS\_OUTPUT.put\_line ( 'FYI: should insert in brix '

|| rec.territory

|| ' '

|| rec.market\_territory

|| ' '

|| rec.sub\_market

|| ' '

|| rec.sub\_market\_leaf

|| ' '

|| rec.LEAF\_DOMAIN\_ID

|| ' '

|| rec.clli\_6

|| ' '

|| rec.brix\_csrs

);

BEGIN

EXECUTE IMMEDIATE insstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.LEAF\_DOMAIN\_ID ,

rec.clli\_6,

rec.brix\_csrs;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t insert '

|| rec.clli\_6

|| ' sam-CSR-VLAN-extract CLLI in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

--RAISE;

END;

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update '

|| rec.clli\_6

|| ' for sam-CSR-VLAN-extract in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

-- insert new or update metric in the same table for sam-ehealth-gi csrs w issues

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set sam\_eh\_nt\_issues = :sam\_eh\_nt\_issues

where clli\_6 = :clli\_6

';

insstmt :=

'insert into SAM\_EH\_NT\_CSR\_CLLI\_SUMM columns (territory, market\_territory, sub\_market, market, LEAF\_DOMAIN\_ID , clli\_6, sam\_eh\_nt\_issues)

values (:territory, :market\_territory, :sub\_market, :market, :LEAF\_DOMAIN\_ID , :clli\_6, :sam\_eh\_nt\_issues)

';

FOR rec IN csrs\_w\_issues

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.sam\_eh\_nt\_issues, rec.clli\_6;

l\_cnt := SQL%ROWCOUNT;

IF (l\_cnt = 0)

THEN

-- there is a possibility that all CLLIs did not make into the table yet.

-- if it did not then insert the clli and the metric

DBMS\_OUTPUT.put\_line ( 'FYI: '

|| l\_cnt

|| ' matching rows for sam CLLI'

);

DBMS\_OUTPUT.put\_line ( 'FYI: should insert in sam '

|| rec.territory

|| ' '

|| rec.market\_territory

|| ' '

|| rec.sub\_market

|| ' '

|| rec.sub\_market\_leaf

|| ' '

|| rec.LEAF\_DOMAIN\_ID

|| ' '

|| rec.clli\_6

|| ' '

|| rec.sam\_eh\_nt\_issues

);

BEGIN

EXECUTE IMMEDIATE insstmt

USING rec.territory,

rec.market\_territory,

rec.sub\_market,

rec.sub\_market\_leaf,

rec.leaf\_domain\_id,

rec.clli\_6,

rec.sam\_eh\_nt\_issues;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t insert '

|| rec.clli\_6

|| ' sam-ehealth-gi-CSRs-W-ISSUES CLLI in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

--RAISE;

END;

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update '

|| rec.clli\_6

|| ' for sam-eh-gi-CSR-issues in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

-- calc matches / mismatches metric between sam and gi

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set sam\_NT\_match = :sam\_NT\_match

, sam\_NT\_l\_match = :sam\_NT\_l\_match

, sam\_NT\_nl\_match = :sam\_NT\_nl\_match

where clli\_6 = :clli\_6

';

FOR rec IN sam\_NT\_match\_csrs

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.sam\_NT\_match,

rec.sam\_NT\_l\_match,

rec.sam\_NT\_nl\_match,

rec.clli\_6;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update sam-GI mismatch in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

END LOOP;

COMMIT;

-- calc matches / mismatches metric between hpov and gi

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set eh\_NT\_match = :eh\_NT\_match

, eh\_NT\_l\_match = :eh\_NT\_l\_match

, eh\_NT\_nl\_match = :eh\_NT\_nl\_match

where clli\_6 = :clli\_6

';

FOR rec IN eh\_NT\_match\_csrs

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.eh\_NT\_match,

rec.eh\_NT\_l\_match,

rec.eh\_NT\_nl\_match,

rec.clli\_6;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update EH match in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

END LOOP;

-- calc matches / mismatches metric between brix and gi

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set brix\_NT\_match = :brix\_NT\_match

, brix\_NT\_l\_match = :brix\_NT\_l\_match

, brix\_NT\_nl\_match = :brix\_NT\_nl\_match

where clli\_6 = :clli\_6';

FOR rec IN brix\_NT\_match\_csrs

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.brix\_NT\_match,

rec.brix\_NT\_l\_match,

rec.brix\_NT\_nl\_match,

rec.clli\_6;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update BRIX match in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

END LOOP;

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set sam\_NT\_mis\_match = :sam\_NT\_mis\_match

where clli\_6 = :clli\_6

';

FOR rec IN sam\_NT\_mis\_match\_csrs

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.sam\_NT\_mis\_match,

rec.clli\_6;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update sam-GI mismatch in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

END LOOP;

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set eh\_NT\_mis\_match = :eh\_NT\_mis\_match

where clli\_6 = :clli\_6

';

FOR rec IN eh\_NT\_mis\_match\_csrs

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.eh\_NT\_mis\_match,

rec.clli\_6;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update EH mismatch in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

END LOOP;

updstmt :=

'update SAM\_EH\_NT\_CSR\_CLLI\_SUMM

set brix\_NT\_mis\_match = :brix\_NT\_mis\_match

where clli\_6 = :clli\_6

';

FOR rec IN brix\_NT\_mis\_match\_csrs

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.brix\_NT\_mis\_match,

rec.clli\_6;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update BRIX mismatch in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

END LOOP;

-- calculate the percentages

BEGIN

UPDATE SAM\_EH\_NT\_CSR\_CLLI\_SUMM

SET eh\_NT\_per =

ROUND ( NVL (eh\_NT\_l\_match, 0)

\* 2

/ DECODE ((NVL (eh\_csrs, 0) + NVL (nt\_l\_csrs, 0)),

0, 1,

NVL (eh\_csrs, 0) + NVL (nt\_l\_csrs, 0)

)

\* 100,

2

),

brix\_NT\_per =

ROUND ( NVL (brix\_NT\_l\_match, 0)

\* 2

/ DECODE ((NVL (brix\_csrs, 0) + NVL (nt\_l\_csrs, 0)),

0, 1,

NVL (brix\_csrs, 0) + NVL (nt\_l\_csrs, 0)

)

\* 100,

2

),

sam\_NT\_per =

ROUND ( NVL (sam\_NT\_l\_match, 0)

\* 2

/ DECODE ((NVL (sam\_csrs, 0) + NVL (nt\_l\_csrs, 0)),

0, 1,

NVL (sam\_csrs, 0) + NVL (nt\_l\_csrs, 0)

)

\* 100,

2

);

-- for rec in (select \* from SAM\_EH\_NT\_CSR\_CLLI\_SUMM)

-- loop

-- dbms\_output.put\_line('round(nvl('||rec.eh\_NT\_l\_match||', 0) \* 2 / decode ((nvl('||rec.eh\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)), 0, 1, nvl('||rec.eh\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)) \* 100, 2)');

-- dbms\_output.put\_line('round(nvl('||rec.sam\_NT\_l\_match||', 0) \* 2 / decode ((nvl('||rec.sam\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)), 0, 1, nvl('||rec.sam\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)) \* 100, 2)');

-- end loop;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update percentages in table: SAM\_EH\_NT\_CSR\_CLLI\_SUMM';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

load\_csr\_regional\_summary();

END load\_csr\_clli\_summary;

PROCEDURE load\_csr\_regional\_summary

IS

methodname VARCHAR2 (30) := 'load\_csr\_regional\_summary';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

err VARCHAR2 (250);

--processname VARCHAR2 (30) := 'SAM\_NT\_CSR\_AUDIT';

BEGIN

BEGIN

-- calculate the percentages

sqlstmt := 'truncate table SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE';

EXECUTE IMMEDIATE sqlstmt;

sqlstmt :=

-- rollup % from CLLI

'INSERT INTO SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE COLUMNS

(territory, market\_territory, sub\_market, nt\_csrs, nt\_l\_csrs, nt\_nl\_csrs, eh\_csrs,

eh\_nt\_match, eh\_nt\_l\_match, eh\_nt\_nl\_match,

sam\_csrs, sam\_nt\_match, SAM\_NT\_L\_MATCH, sam\_nt\_nl\_match,

sam\_eh\_nt\_issues,sam\_nt\_mis\_match,eh\_nt\_mis\_match,

brix\_csrs, brix\_nt\_match, brix\_nt\_l\_match, brix\_nt\_nl\_match, brix\_nt\_mis\_match)

select \* from (

SELECT territory, market\_territory, sub\_market, SUM (NVL (nt\_csrs, 0)) nt\_csrs,

SUM (NVL (nt\_l\_csrs, 0)) nt\_l\_csrs,

SUM (NVL (nt\_nl\_csrs, 0)) nt\_nl\_csrs,

SUM (NVL (eh\_csrs, 0)) eh\_csrs,

SUM (NVL (eh\_nt\_match, 0)) eh\_nt\_match,

SUM (NVL (eh\_nt\_l\_match, 0)) eh\_nt\_l\_match,

SUM (NVL (eh\_nt\_nl\_match, 0)) eh\_nt\_nl\_match,

SUM (NVL (sam\_csrs, 0)) sam\_csrs,

SUM (NVL (sam\_nt\_match, 0)) sam\_nt\_match,

SUM (NVL (SAM\_NT\_L\_MATCH, 0)) SAM\_NT\_L\_MATCH,

SUM (NVL (sam\_nt\_nl\_match, 0)) sam\_nt\_nl\_match,

SUM (NVL (sam\_eh\_nt\_issues, 0)) sam\_eh\_nt\_issues,

SUM (NVL (sam\_nt\_mis\_match, 0)) sam\_nt\_mis\_match,

SUM (NVL (eh\_nt\_mis\_match, 0)) eh\_nt\_mis\_match,

SUM (NVL (brix\_csrs, 0)) brix\_csrs,

SUM (NVL (brix\_nt\_match, 0)) brix\_nt\_match,

SUM (NVL (brix\_nt\_l\_match, 0)) brix\_nt\_l\_match,

SUM (NVL (brix\_nt\_nl\_match, 0)) brix\_nt\_nl\_match,

SUM (NVL (brix\_nt\_mis\_match, 0)) brix\_nt\_mis\_match

FROM SAM\_EH\_NT\_CSR\_CLLI\_SUMM

GROUP BY ROLLUP (territory, market\_territory, sub\_market)

ORDER BY territory, market\_territory, sub\_market)

where market\_territory != ''unknown'' or market\_territory is null'

;

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

sqlstmt:= 'update SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE wk set market\_territory =''unknown'',sub\_market=''unknown'' where territory=''unknown''' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE wk set TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE wk set TERRITORY\_MARKET\_SUB = market\_territory where sub\_market is null and market\_territory is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt:= 'update SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE wk set TERRITORY\_MARKET\_SUB = territory where sub\_market is null and market\_territory is null' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t rollup percentages for the region in table: SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

END;

BEGIN

UPDATE SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE

SET eh\_nt\_per =

ROUND ( NVL (eh\_nt\_l\_match, 0)

\* 2

/ DECODE ((NVL (eh\_csrs, 0) + NVL (nt\_l\_csrs, 0)),

0, 1,

NVL (eh\_csrs, 0) + NVL (nt\_l\_csrs, 0)

)

\* 100,

2

),

sam\_nt\_per =

ROUND ( NVL (SAM\_NT\_L\_MATCH, 0)

\* 2

/ DECODE ((NVL (sam\_csrs, 0) + NVL (nt\_l\_csrs, 0)),

0, 1,

NVL (sam\_csrs, 0) + NVL (nt\_l\_csrs, 0)

)

\* 100,

2

),

brix\_nt\_per =

ROUND ( NVL (brix\_nt\_l\_match, 0)

\* 2

/ DECODE ((NVL (brix\_csrs, 0) + NVL (nt\_l\_csrs, 0)),

0, 1,

NVL (brix\_csrs, 0) + NVL (nt\_l\_csrs, 0)

)

\* 100,

2

);

-- for rec in (select \* from SAM\_EH\_NT\_CSR\_CLLI\_SUMM)

-- loop

-- dbms\_output.put\_line('round(nvl('||rec.eh\_nt\_l\_match||', 0) \* 2 / decode ((nvl('||rec.eh\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)), 0, 1, nvl('||rec.eh\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)) \* 100, 2)');

-- dbms\_output.put\_line('round(nvl('||rec.SAM\_NT\_L\_MATCH||', 0) \* 2 / decode ((nvl('||rec.sam\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)), 0, 1, nvl('||rec.sam\_csrs||',0) + nvl('||rec.nt\_l\_csrs||',0)) \* 100, 2)');

-- end loop;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update percentages in table: SAM\_EH\_NT\_CSR\_REG\_SUMM\_TABLE';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

-- RAISE;

END;

-- RAISE;

END load\_csr\_regional\_summary;

PROCEDURE load\_csr\_vlan\_paths is

sqlStmt varchar2(32000);

message varchar2(250);

methodName varchar2(30) := 'load\_granite\_CSR\_vlan\_paths';

BEGIN

message := 'Error: '|| methodName ||'(): Cant truncate table: XG\_REPORTS.NTLS\_CSR\_vlan\_paths\_wk';

sqlStmt := 'truncate table NG\_REPORTS.NTLS\_CSR\_vlan\_paths\_wk';

execute immediate sqlStmt;

-- this query looks at only the 1st leg of the VLAN

-- and 1st leg of the 1Gig and 10Gig path

sqlStmt :=

'insert into /\*parallel (s,6) \*/ NTLS\_CSR\_VLAN\_PATHS columns ( EQP\_REFERENCE\_ID, CARD\_REFERENCE\_ID, CKT\_PATH\_REFERENCE\_ID, port\_BANDWIDTH, path\_TYPE, vlan\_inst\_id, vlan\_hum\_id, vlan\_number, vlan\_status, extract\_date)

with

csr\_gige\_paths as (select distinct XNE.EQP\_REFERENCE\_ID, p.CARD\_REFERENCE\_ID --, p.port\_inst\_id, pli.COMPONENT\_ID,(pli.rel\_order), PLI.LEG\_NAME

,cpi.TRAIL\_ID gige\_inst\_id, cpi.BANDWIDTH, cpi.type

from NG\_REPORTS.NTLS\_CSR\_PARSED\_WK xne,ng\_reports.PORT p,ng\_topology.TRAIL cpi,ng\_topology.TRAIL\_COMPONENT pli,ng\_topology.TRAIL\_COMPONENT\_ELEMENT plm,ng\_topology.TRAIL\_ELEMENT cpe

where p.EQP\_REFERENCE\_ID = xne.EQP\_REFERENCE\_ID and cpi.TRAIL\_ID in (p.CKT\_PATH\_REFERENCE\_ID, p.NEXT\_PATH\_REFERENCE\_ID) and cpi.BANDWIDTH like ''%Gbps'' and pli.TRAIL\_ID = cpi.TRAIL\_ID and pli.COMPONENT\_ID = plm.COMPONENT\_ID

-- and plm.sequence = cpe.sequence

and plm.ELEMENT\_ID = cpe.ELEMENT\_ID and CPE.ELEMENT\_REF\_ID = p.PORT\_REFERENCE\_ID and CPE.ELEMENT\_TYPE = ''E'' and pli.SEQUENCE = (select min(SEQUENCE) SEQUENCE from ng\_topology.TRAIL\_COMPONENT li where li.TRAIL\_ID = cpi.TRAIL\_ID)

),

gige\_vlan\_paths as (

select distinct gige.TRAIL\_ID gige\_inst\_id , VLAN.TRAIL\_ID vlan\_inst\_id

,vlan.TRAIL\_NAME vlan\_hum\_id, regexp\_substr(regexp\_substr(vlan.TRAIL\_NAME, ''VLAN(-|[[:space:]]|)\*[[:digit:]]+''), ''[[:digit:]]+'') vlan\_number, vlan.STATUS

from ng\_topology.TRAIL gige, ng\_topology.TRAIL vlan, ng\_topology.TRAIL\_COMPONENT pli, ng\_topology.TRAIL\_COMPONENT\_ELEMENT plm, ng\_topology.TRAIL\_ELEMENT cpe

where gige.BANDWIDTH like ''%Gbps'' and vlan.BANDWIDTH = ''VLAN'' and vlan.TYPE = ''EBH'' and VLAN.TRAIL\_ID = CPE.TRAIL\_ID and gige.TRAIL\_ID = CPE.ELEMENT\_REF\_ID and CPE.ELEMENT\_TYPE = ''P''

and VLAN.TRAIL\_ID = pli.TRAIL\_ID and pli.COMPONENT\_ID = plm.COMPONENT\_ID

-- and plm.sequence = cpe.sequence

and plm.ELEMENT\_ID = cpe.ELEMENT\_ID and pli.SEQUENCE=(select min(SEQUENCE) from ng\_topology.TRAIL\_COMPONENT pl where pl.TRAIL\_ID = VLAN.TRAIL\_ID)

)select ngp.EQP\_REFERENCE\_ID, ngp.CARD\_REFERENCE\_ID ,ngp.gige\_inst\_id CKT\_PATH\_REFERENCE\_ID, ngp.BANDWIDTH, ngp.type , gvp.vlan\_inst\_id, gvp.vlan\_hum\_id, gvp.vlan\_number, gvp.STATUS vlan\_status, sysdate

from csr\_gige\_paths ngp, gige\_vlan\_paths gvp where ngp.gige\_inst\_id = gvp.gige\_inst\_id';

execute immediate sqlStmt;

commit;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(SubStr(message||TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

RAISE;

END load\_csr\_vlan\_paths;

PROCEDURE audit\_al\_csr\_vlans

IS

methodname VARCHAR2 (30) := 'audit\_al\_csr\_vlans';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

sqlstmt := 'delete from SAM\_CSR\_VLAN\_AUDIT where CSR\_VENDOR = ''ALCATEL-LUCENT''';

EXECUTE IMMEDIATE sqlstmt;

BEGIN

COMMIT;

sqlstmt :=

'insert into SAM\_CSR\_VLAN\_AUDIT columns (CSR\_DEVICE\_NAME, CSR\_VENDOR, ne\_hostName, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ng\_vlan\_number -- ne\_hostName???

, vlan\_inst\_id, vlan\_status, PORT\_NAME)

WITH

al\_vlans AS(SELECT xnvp.\*, xne.CSR\_DEVICE\_NAME

FROM NTLS\_CSR\_vlan\_paths xnvp , NG\_Reports.NTLS\_CSR\_PARSED\_WK xne

where xne.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID and xne.CSR\_DEVICE\_NAME like ''%-AL-%'' AND xnvp.vlan\_number IS NOT NULL)

,matches AS(

SELECT aud.CSR\_DEVICE\_NAME, sam.OUT\_ENCAP\_VALUE ne\_vlan\_number,sam.DISPLAYED\_NAME, sam.NODE\_ID, ''BOTH'' match\_code,NULL match\_status

,xnvp.vlan\_number ng\_vlan\_number, xnvp.vlan\_inst\_id,

xnvp.vlan\_status, aud.CSR\_VENDOR, SUBSTR(sam.PORT\_NAME, 0, INSTR(sam.PORT\_NAME, '':'')-1) AS PORT\_NAME

FROM NG\_REPORTS.SAM\_EBH\_RTR sam

JOIN SAM\_NT\_AL\_CSR\_AUDIT aud ON upper(aud.CSR\_DEVICE\_NAME) = upper(sam.NODE\_NAME) -- Audit table

JOIN al\_vlans xnvp ON aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID AND sam.OUT\_ENCAP\_VALUE = xnvp.vlan\_number

WHERE sam.NODE\_ID IS NOT NULL AND sam.NODE\_NAME not like ''%7750-\_\_''),

ne\_only AS(SELECT aud.CSR\_DEVICE\_NAME, sam.DISPLAYED\_NAME, sam.NODE\_ID,sam.OUT\_ENCAP\_VALUE ne\_vlan\_number, ''NE Only'' match\_code,''Missing in Nautilus'' match\_status, null,

null ng\_vlan\_number, NULL vlan\_inst\_id, NULL vlan\_status,aud.CSR\_VENDOR, SUBSTR(sam.PORT\_NAME, 0, INSTR(sam.PORT\_NAME, '':'')-1) AS PORT\_NAME

FROM NG\_REPORTS.SAM\_EBH\_RTR sam

JOIN SAM\_NT\_AL\_CSR\_AUDIT aud ON upper(aud.CSR\_DEVICE\_NAME) = upper(sam.NODE\_NAME)

WHERE sam.OUT\_ENCAP\_VALUE IS NOT NULL AND sam.NODE\_ID IS NOT NULL AND sam.NODE\_NAME not like ''%7750-\_\_''

AND NOT EXISTS (SELECT 1

FROM al\_vlans xnvp

WHERE aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID AND upper(xnvp.CSR\_DEVICE\_NAME) = upper(aud.CSR\_DEVICE\_NAME) AND sam.OUT\_ENCAP\_VALUE = xnvp.vlan\_number)), xng\_only AS

(SELECT aud.CSR\_DEVICE\_NAME, NULL DISPLAYED\_NAME, NULL NODE\_ID, null ne\_vlan\_number, ''Nautilus Only'' match\_code, ''Missing in NE'' match\_status, xnvp.vlan\_number ng\_vlan\_number, xnvp.vlan\_inst\_id,

xnvp.vlan\_status, aud.CSR\_VENDOR, NULL PORT\_NAME

FROM al\_vlans xnvp

JOIN SAM\_NT\_AL\_CSR\_AUDIT aud ON aud.EQP\_REFERENCE\_ID = xnvp.EQP\_REFERENCE\_ID AND upper(AUD.CSR\_DEVICE\_NAME) = upper(xnvp.CSR\_DEVICE\_NAME) and not exists(select 1 from sam\_ebh\_rtr sam

where upper(aud.CSR\_DEVICE\_NAME) = upper(sam.NODE\_NAME) AND sam.OUT\_ENCAP\_VALUE = xnvp.vlan\_number and sam.OUT\_ENCAP\_VALUE IS NOT NULL AND sam.NODE\_NAME not like ''%7750-\_\_''))

SELECT CSR\_DEVICE\_NAME, CSR\_VENDOR, CSR\_DEVICE\_NAME, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ng\_vlan\_number, vlan\_inst\_id, vlan\_status, port\_name

FROM matches

UNION

SELECT CSR\_DEVICE\_NAME, CSR\_VENDOR, CSR\_DEVICE\_NAME, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ng\_vlan\_number, vlan\_inst\_id, vlan\_status, port\_name

FROM ne\_only

UNION

SELECT CSR\_DEVICE\_NAME, CSR\_VENDOR, NULL, DISPLAYED\_NAME, NODE\_ID, ne\_vlan\_number, match\_code, match\_status, ng\_vlan\_number, vlan\_inst\_id, vlan\_status, port\_name

FROM xng\_only'

;

EXECUTE IMMEDIATE (sqlstmt);

COMMIT;

insert\_al\_CSR\_vlan\_issues ();

EXCEPTION

WHEN OTHERS

THEN

MESSAGE := 'Error: ' || methodname || '(): Cant audit AL VLANs';

DBMS\_OUTPUT.put\_line (SUBSTR (MESSAGE || TO\_CHAR (SQLCODE) || ': ' || SQLERRM,1,255));

/\*watchdog.logerror (methodName, 8000, SUBSTR (message || TO\_CHAR (SQLCODE) || ': ' || SQLERRM, 1, 255), 'Y' ); \*/

RAISE;

END;

BEGIN

assign\_area\_region\_to\_csr\_vlan();

insert\_al\_CSR\_vlan\_issues();

END;

END audit\_al\_csr\_vlans;

PROCEDURE insert\_al\_CSR\_vlan\_issues

IS

methodname VARCHAR2 (30) := 'insert\_al\_CSR\_vlan\_issues';

MESSAGE VARCHAR2 (300);

sqlstmt VARCHAR2 (32767);

CURSOR dup\_l\_vlan

IS

-- if the same DEVICE\_NAME, nxa.VLAN\_number has > 1 LIVE status then

-- mark it as dup

WITH vlan\_status AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status,

match\_code, nxa.vlan\_inst\_id

FROM SAM\_CSR\_VLAN\_AUDIT nxa JOIN ng\_topology.trail cpi

ON cpi.trail\_id = nxa.vlan\_inst\_id

),

dup\_l\_vlans AS

(

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number,

COUNT (vlan\_inst\_id)

FROM vlan\_status nxa

WHERE status = 'LIVE' AND match\_code <> 'Nautilus Only'

GROUP BY nxa.csr\_device\_name, nxa.ng\_vlan\_number, status

HAVING COUNT (vlan\_inst\_id) > 1)

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status,

match\_code, nxa.vlan\_inst\_id

FROM dup\_l\_vlans dup JOIN vlan\_status nxa

ON nxa.csr\_device\_name = dup.csr\_device\_name

AND nxa.ng\_vlan\_number = dup.ng\_vlan\_number

ORDER BY nxa.csr\_device\_name, nxa.ng\_vlan\_number;

-- if the same csr\_device\_name, nxa.VLAN\_number has > 1 status then

-- and 1 is LIVE and other is not then mark NL as 'NE Only' & 'LIVE path exists'

CURSOR LIVE\_vlan\_exists

IS

WITH vlan\_status AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status,

match\_code, nxa.vlan\_inst\_id

FROM SAM\_CSR\_VLAN\_AUDIT nxa JOIN ng\_topology.trail cpi

ON cpi.trail\_id = nxa.vlan\_inst\_id

),

l\_and\_nl\_vlans AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status = 'LIVE'

INTERSECT

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status <> 'LIVE')

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number,

nxa.vlan\_inst\_id

FROM l\_and\_nl\_vlans dup JOIN vlan\_status nxa

ON nxa.csr\_device\_name = dup.csr\_device\_name

AND nxa.ng\_vlan\_number = dup.ng\_vlan\_number

WHERE nxa.status <> 'LIVE'

ORDER BY nxa.csr\_device\_name, nxa.ng\_vlan\_number;

BEGIN

DBMS\_OUTPUT.put\_line ('FYI: In ' || methodname);

BEGIN

EXECUTE IMMEDIATE ('delete CSR\_VLAN\_AUDIT\_ISSUES where csr\_vendor = ''ALCATEL-LUCENT''');

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t delete rows in table: CSR\_VLAN\_AUDIT\_ISSUES';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

-- mark duplicate in Granite as NE Only, or LIVE Path Exists

-- if there is a path revision then the same VLAN will be present twice.

-- BUT that is NOT a duplicate

-- if NOT a revision path and VLAN# is present multiple times THEN it is dup

-- mark duplicate in Granite as NE Only, or LIVE Path Exists

-- else 2 DUP vlans on a csr then Duplicate

sqlstmt :=

'update SAM\_CSR\_VLAN\_AUDIT

set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

, match\_status = nvl2(match\_status, match\_status|| '', Dup in NT'', ''Dup in NT'')

where vlan\_inst\_id = :p\_vlan\_inst\_id

and ng\_vlan\_number = :p\_ng\_vlan\_number

and CSR\_DEVICE\_NAME = :p\_csr\_device\_name

';

FOR rec IN dup\_l\_vlan

LOOP

BEGIN

EXECUTE IMMEDIATE sqlstmt

USING rec.vlan\_inst\_id,

rec.ng\_vlan\_number,

rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update ''Dup in NT'' in table: SAM\_CSR\_VLAN\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

COMMIT;

BEGIN

sqlstmt :=

'insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

select distinct W.CSR\_DEVICE\_NAME

, case when ne\_vlan\_number is null then ng\_vlan\_number

else ne\_vlan\_number

end vlan\_number,1,''ALCATEL-LUCENT''

from SAM\_CSR\_VLAN\_AUDIT w

where W.MATCH\_STATUS like ''%Dup in NT%''';

EXECUTE IMMEDIATE sqlstmt;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t insert into CSR\_VLAN\_AUDIT\_ISSUES issue Duplicate in NT ';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

COMMIT;

BEGIN

-- check the ne\_hostname matches MTCE std, else update match\_code = ''NE Only''

-- and match\_status = ''ne\_hostname not MTCE Stds compliant''

/\* sqlstmt :=

' update SAM\_CSR\_VLAN\_AUDIT aud set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

where exists (select 1

from SAM\_EBH\_RTR i

where DEVICE\_TYPE = ''CSR''

and AUD.CSR\_DEVICE\_NAME = I.NODE\_NAME

and parse\_status like ''%11%''

)';

EXECUTE IMMEDIATE sqlstmt;\*/

COMMIT;

sqlstmt :=

'insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

select distinct W.CSR\_DEVICE\_NAME

, case when ne\_vlan\_number is null then ng\_vlan\_number

else ne\_vlan\_number

end vlan\_number,17,''ALCATEL-LUCENT''

from SAM\_CSR\_VLAN\_AUDIT w

where

W.CSR\_VENDOR = ''ALCATEL-LUCENT''

and NOT REGEXP\_LIKE ( CSR\_DEVICE\_NAME,

''([A-Z|0-9]{8})T[0-9]{1}A-P-(AL|CI)-(G|[0-9]{1}|[0-9]{3})[0-9]{3}-(S|E|H|R|W|I|D|[0-9]{1})(I|[0-9]{1})([^\\[\\]]\*)?'' )';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update name does not match MTCE Std';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- check if ne\_hostnames' don't match first 6-char match some MTSO CLLI

-- then match\_code = ''NE Only'' and match\_status = ''1st 6-chars of don''''t match any MTSO CLLIs''

sqlstmt :=

'insert into CSR\_VLAN\_AUDIT\_ISSUES columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

WITH

CLLI\_6\_MATCH AS

(

SELECT DISTINCT SUBSTR (clli, 1, 6) clli\_6

FROM clli\_domain\_map cdm

where cdm.LEAF\_DOMAIN\_ID not in (1021, 1016)

)

,

CLLI\_MISMATCH AS

(

-- get devices where 6-chars

SELECT ei.CSR\_device\_name,

SUBSTR (ei.csr\_device\_name, 1, 8) csr\_clli, cdm6.clli\_6 vsm\_clli

FROM SAM\_CSR\_VLAN\_AUDIT ei JOIN clli\_6\_match cdm6

ON cdm6.clli\_6 = SUBSTR (ei.csr\_device\_name, 1, 6)

WHERE NOT EXISTS (SELECT cdm.clli

FROM clli\_domain\_map cdm

WHERE SUBSTR (ei.csr\_device\_name, 1, 8) = cdm.clli)

),

BAD\_CLLI AS

(

SELECT ei.csr\_device\_name

FROM SAM\_CSR\_VLAN\_AUDIT ei

MINUS

SELECT ei.csr\_device\_name

FROM SAM\_CSR\_VLAN\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 8) = SUBSTR (ei.csr\_device\_name, 1, 8)

MINUS

SELECT ei.csr\_device\_name

FROM SAM\_CSR\_VLAN\_AUDIT ei JOIN clli\_domain\_map cdm

ON SUBSTR (cdm.clli, 1, 6) = SUBSTR (ei.csr\_device\_name, 1, 6)

),

ISSUES AS

(

SELECT cm.csr\_device\_name, 22 issue\_id

FROM clli\_mismatch cm

UNION

SELECT bc.csr\_device\_name, 2 issue\_id

FROM bad\_clli bc

)

SELECT distinct i.csr\_device\_name, W.NE\_VLAN\_NUMBER, i.issue\_id, ''ALCATEL-LUCENT''

FROM ISSUES i, SAM\_CSR\_VLAN\_AUDIT w

where i.csr\_device\_name = W.CSR\_DEVICE\_NAME

and W.NE\_VLAN\_NUMBER is not null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND

THEN

NULL;

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t mark 1st 6-char don''t match MTSO CLLIs';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- device not found in sam vlan extract

sqlstmt :=

' insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

select distinct aud.CSR\_DEVICE\_NAME, ng\_vlan\_number, 13, ''ALCATEL-LUCENT''

from SAM\_CSR\_VLAN\_AUDIT aud

where aud.ne\_hostname is null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant insert issue ''Not in SAM VLAN Extract''';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- device not found in NT

sqlstmt :=

' insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

select distinct aud.CSR\_DEVICE\_NAME, ne\_vlan\_number, 15, ''ALCATEL-LUCENT''

from SAM\_CSR\_VLAN\_AUDIT aud

where aud.vlan\_inst\_id is null';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname

|| '(): Cant insert issue ''Not in NT''';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

-- if there is a path revision then the same VLAN will be present twice.

-- BUT that is NOT a duplicate

-- if NOT a revision path and VLAN# is present multiple times THEN it is dup

-- if 1 == LIVE and the other <> LIVE then mark <> LIVE as 'LIVE in Xng'

sqlstmt :=

'update SAM\_CSR\_VLAN\_AUDIT

set match\_code = case when match\_code <> ''Nautilus Only'' then ''NE Only'' else ''Nautilus Only'' end

, match\_status = nvl2(match\_status, ''LIVE Path Exists,'' ||match\_status , ''LIVE Path Exists'')

where vlan\_inst\_id = :p\_vlan\_inst\_id

and ng\_vlan\_number = :p\_ng\_vlan\_number

and CSR\_DEVICE\_NAME = :p\_csr\_device\_name

--and csr\_vendor = :p\_csr\_vendor

';

FOR rec IN LIVE\_vlan\_exists

LOOP

BEGIN

EXECUTE IMMEDIATE sqlstmt

USING rec.vlan\_inst\_id,

rec.ng\_vlan\_number,

rec.csr\_device\_name; --, p\_csr\_vendor;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Can''t update ''LIVE Path Exists'' in table: SAM\_CSR\_VLAN\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

ROLLBACK;

RAISE;

END;

END LOOP;

COMMIT;

BEGIN

-- IF BOTH and has issues then mark it NE

sqlstmt :=

' update SAM\_CSR\_VLAN\_AUDIT aud

set match\_code = ''NE Only''

where exists (select 1

from

CSR\_VLAN\_AUDIT\_ISSUES vi,

CSR\_issue n

where aud.MATCH\_CODE = ''BOTH''

and aud.csr\_vendor = ''ALCATEL-LUCENT''

and VI.CSR\_VENDOR=''ALCATEL-LUCENT''

and AUD.CSR\_DEVICE\_NAME = VI.CSR\_DEVICE\_NAME

and AUD.NE\_VLAN\_NUMBER = VI.VLAN\_NUMBER

and N.IS\_CRITICAL=''Y''

and N.CSR\_ISSUE\_ID = vi.CSR\_ISSUE\_ID

)';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant update match\_status to ''NE Only''';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

-- device not found in sam vlan extract

sqlstmt :=

' insert into CSR\_VLAN\_AUDIT\_ISSUES

columns (csr\_device\_name, vlan\_number, csr\_issue\_id, csr\_vendor)

WITH vlan\_status AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, status, match\_code,

nxa.vlan\_inst\_id

FROM SAM\_CSR\_VLAN\_AUDIT nxa JOIN ng\_topology.trail cpi

ON cpi.trail\_id = nxa.vlan\_inst\_id

),

l\_and\_nl\_vlans AS

(SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status = ''LIVE''

INTERSECT

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number

FROM vlan\_status nxa

WHERE status <> ''LIVE''),

LIVE\_vlan\_exists AS

(

SELECT nxa.csr\_device\_name, nxa.ng\_vlan\_number, nxa.vlan\_inst\_id

FROM l\_and\_nl\_vlans dup JOIN vlan\_status nxa

ON nxa.csr\_device\_name = dup.csr\_device\_name

AND nxa.ng\_vlan\_number = dup.ng\_vlan\_number

WHERE nxa.status <> ''LIVE'')

SELECT DISTINCT aud.csr\_device\_name, ng\_vlan\_number, 31, ''ALCATEL-LUCENT''

FROM SAM\_CSR\_VLAN\_AUDIT aud

WHERE vlan\_inst\_id IS NOT NULL

AND match\_code = ''NE Only''

AND NOT EXISTS (

SELECT 1

FROM csr\_vlan\_audit\_issues nvai

WHERE nvai.vlan\_number = aud.ng\_vlan\_number

AND aud.csr\_device\_name = nvai.csr\_device\_name

AND nvai.csr\_vendor = ''ALCATEL-LUCENT'')

AND EXISTS (

SELECT 1

FROM LIVE\_vlan\_exists lle

WHERE lle.csr\_device\_name = aud.csr\_device\_name

AND aud.ng\_vlan\_number = lle.ng\_vlan\_number)';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: '

|| methodname

|| '(): Cant insert issue ''LIVE VLAN Exists''';

DBMS\_OUTPUT.put\_line (SUBSTR ( MESSAGE

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

END insert\_al\_CSR\_vlan\_issues;

PROCEDURE assign\_area\_region\_to\_csr\_vlan

IS

methodname VARCHAR2 (30) := 'assign\_area\_region\_to\_csr\_vlan';

MESSAGE VARCHAR2 (300);

updstmt VARCHAR2 (32767);

processname VARCHAR2 (30) := 'SAM\_GI\_CSR\_AUDIT';

CURSOR area\_region\_dev IS WITH domains AS

(SELECT DISTINCT SUBSTR (CLLI, 1, 6) clli\_6, TERRITORY, SUB\_MARKET, SUB\_MARKET\_LEAF , LEAF\_DOMAIN\_ID

FROM NG\_Reports.clli\_domain\_map\_v

WHERE SUB\_MARKET <> 'NNO' AND SUB\_MARKET <> 'OSS')

SELECT di.TERRITORY, di.SUB\_MARKET, di.SUB\_MARKET\_LEAF MARKET, di.LEAF\_DOMAIN\_ID, aud.csr\_device\_name, clli\_6

FROM SAM\_CSR\_VLAN\_AUDIT aud

JOIN domains di ON di.clli\_6 = SUBSTR (aud.csr\_device\_name, 1, 6);

CURSOR area\_region\_SAM IS

SELECT di.TERRITORY, di.SUB\_MARKET, di.market, di.LEAF\_DOMAIN\_ID, di.csr\_device\_name, di.CLLI

FROM SAM\_CSR\_VLAN\_AUDIT aud

JOIN SAM\_NT\_AL\_CSR\_AUDIT di ON di.csr\_device\_name = aud.csr\_device\_name -- Has to be Created

where aud.TERRITORY is null;

BEGIN

updstmt :=

'update SAM\_CSR\_VLAN\_AUDIT set TERRITORY = :TERRITORY, SUB\_MARKET = :SUB\_MARKET, market = :market, LEAF\_DOMAIN\_ID = :LEAF\_DOMAIN\_ID, CLLI = :clli\_6

where csr\_device\_name = :csr\_device\_name';

FOR rec IN area\_region\_dev

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt

USING rec.TERRITORY,rec.SUB\_MARKET,rec.MARKET,rec.LEAF\_DOMAIN\_ID,rec.clli\_6,rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE := 'Error: ' || methodname || '(): Cant update clli for ' || rec.csr\_device\_name || ' in table: SAM\_CSR\_VLAN\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR (MESSAGE || TO\_CHAR (SQLCODE) || ': ' || SQLERRM, 1, 255));

ROLLBACK;

RAISE;

END;

END LOOP;

FOR rec IN area\_region\_SAM

LOOP

BEGIN

EXECUTE IMMEDIATE updstmt USING rec.TERRITORY, rec.SUB\_MARKET, rec.market, rec.LEAF\_DOMAIN\_ID, rec.CLLI, rec.csr\_device\_name;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE := 'Error: ' || methodname || '(): Cant update clli for ' || rec.csr\_device\_name || ' in table: SAM\_CSR\_VLAN\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR (MESSAGE || TO\_CHAR (SQLCODE) || ': ' || SQLERRM, 1, 255));

ROLLBACK;

RAISE;

END;

END LOOP;

-- area region for put xng\_only based on the area/region of ne\_inst\_id???????????

BEGIN

-- if there still are some devices which cant be mapped to any domain then mark those as unknown

updstmt :=

'update SAM\_CSR\_VLAN\_AUDIT set TERRITORY = ''unknown'' , SUB\_MARKET = ''unknown'' , market = ''unknown'' , LEAF\_DOMAIN\_ID = 0 , CLLI = SUBSTR (csr\_device\_name, 1, 6) where TERRITORY is null' ;

EXECUTE IMMEDIATE updstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE := 'Error: ' || methodname || '(): Cant update TERRITORY/market\_treritory/SUB\_MARKET from to unknown in table: SAM\_CSR\_VLAN\_AUDIT';

DBMS\_OUTPUT.put\_line (SUBSTR (MESSAGE || TO\_CHAR (SQLCODE) || ': ' || SQLERRM,1, 255 ) );

ROLLBACK;

RAISE;

END;

END assign\_area\_region\_to\_csr\_vlan;

PROCEDURE load\_vlan\_csr\_clli\_summary

IS

methodname VARCHAR2 (30) := 'load\_vlan\_csr\_clli\_summary';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

EXECUTE IMMEDIATE 'truncate table SAM\_CSR\_VLAN\_CLLI\_SUMM';

-- generate device summary

BEGIN

sqlstmt :=

'insert into SAM\_CSR\_VLAN\_CLLI\_SUMM columns (territory, sub\_market, market, clli, ne\_total\_csr\_vlan, matched\_l\_csr\_vlan, matched\_nl\_csr\_vlan, mismatched\_csr\_vlan

, nt\_nl\_2\_total\_vlan\_pct, vlan\_COMPLIANCE)

WITH csr as(

select distinct territory, sub\_market, market, leaf\_domain\_id, clli, csr\_device\_name, csr\_vendor

from SAM\_CSR\_VLAN\_AUDIT ng

where ng.CSR\_DEVICE\_NAME is not null),csr\_VLAN as(

select distinct NODE\_NAME CSR\_DEVICE\_NAME, OUT\_ENCAP\_VALUE vlan\_number

from NG\_REPORTS.SAM\_EBH\_RTR ng

where ng.OUT\_ENCAP\_VALUE > 0 and ng.NODE\_NAME not like ''%7750-\_\_''),CNT\_csr\_VLANs as(

select distinct ng.CSR\_DEVICE\_NAME, count(vlan\_number) total\_ne\_csr\_vlan

from csr\_VLAN nv

join csr ng on ng.CSR\_DEVICE\_NAME = nv.CSR\_DEVICE\_NAME

group by ng.CSR\_DEVICE\_NAME),live\_match as(

select distinct aud.CSR\_DEVICE\_NAME, aud.ne\_vlan\_number, vlan\_status

from SAM\_CSR\_VLAN\_AUDIT aud

where match\_code = ''BOTH'' and aud.csr\_vendor = ''ALCATEL-LUCENT'' and aud.ne\_vlan\_number > 0 and aud.vlan\_status = ''Live''),nonlive\_match as(

select distinct aud.csr\_DEVICE\_NAME, aud.ne\_vlan\_number

from SAM\_CSR\_VLAN\_AUDIT aud

where match\_code = ''BOTH'' and aud.csr\_vendor = ''ALCATEL-LUCENT'' and aud.ne\_vlan\_number > 0 and aud.vlan\_status <> ''Live''

minus

select distinct aud.csr\_DEVICE\_NAME, aud.ne\_vlan\_number

from SAM\_CSR\_VLAN\_AUDIT aud

where match\_code = ''BOTH'' and aud.csr\_vendor = ''ALCATEL-LUCENT'' and aud.ne\_vlan\_number > 0 and aud.vlan\_status = ''Live'') ,CNT\_BOTH\_LIVE\_VLANs as( -- good matches

select csr\_DEVICE\_NAME, nvl (count(csr\_DEVICE\_NAME), 0) matched\_l\_csr\_vlan

from live\_match aud

group by csr\_DEVICE\_NAME), CNT\_BOTH\_NONLIVE\_VLANs as( -- good matches

select csr\_DEVICE\_NAME, nvl( count(csr\_DEVICE\_NAME), 0) matched\_nl\_csr\_vlan

-- , nvl(sum(case when vlan\_status = ''Live'' then 1 else 0 end), 0) matched\_l\_csr\_vlan

-- , nvl(sum(case when vlan\_status <> ''Live'' then 1 else 0 end), 0) matched\_nl\_csr\_vlan

from nonlive\_match aud

group by aud.csr\_DEVICE\_NAME) ,MISMATCH\_VLANs as( -- bad matches

select distinct aud.csr\_DEVICE\_NAME, aud.ne\_vlan\_number, aud.ng\_vlan\_number, vlan\_status

from SAM\_CSR\_VLAN\_AUDIT aud

where match\_code <> ''BOTH'' and aud.csr\_vendor = ''ALCATEL-LUCENT'' and (aud.ne\_vlan\_number > 0 or aud.ng\_vlan\_number > 0)),CNT\_MISMATCH\_VLANs as( -- bad match cnts

select aud.CSR\_DEVICE\_NAME, count(1) mismatch\_cnt

from MISMATCH\_VLANs aud

group by aud.csr\_DEVICE\_NAME

)

select territory, sub\_market, market,clli, sum(nvl(nv.total\_ne\_csr\_vlan, 0)) ne\_total\_csr\_vlan, sum(nvl(matched\_l\_csr\_vlan, 0)) matched\_l\_csr\_vlan

, sum(nvl(matched\_nl\_csr\_vlan, 0)) matched\_nl\_csr\_vlan, sum(nvl(mismatch\_cnt, 0)) mismatched\_csr\_vlan, round(sum(NVL(matched\_nl\_csr\_vlan, 0)) / decode(sum(NVL(matched\_l\_csr\_vlan, 0)), 0, 1, sum(nvl(matched\_l\_csr\_vlan, 0)))

\* 100, 2) gi\_nl\_2\_total\_vlan\_pct, round (sum(nvl(matched\_l\_csr\_vlan,0)) / decode (sum(nvl( nv.total\_ne\_csr\_vlan, 0)) - sum(nvl(matched\_nl\_csr\_vlan, 0)), 0 , 1, sum(nvl(nv.total\_ne\_csr\_vlan,0)) - sum(nvl(matched\_nl\_csr\_vlan,0)))

\* 100 , 2) vlan\_COMPLIANCE

from csr ng

left outer join CNT\_csr\_VLANs nv on nv.csr\_DEVICE\_NAME = ng.csr\_DEVICE\_NAME

left outer join CNT\_BOTH\_LIVE\_VLANs lmv on lmv.csr\_DEVICE\_NAME = ng.csr\_DEVICE\_NAME

left outer join CNT\_BOTH\_NONLIVE\_VLANs nlmv on nlmv.csr\_DEVICE\_NAME = ng.csr\_DEVICE\_NAME

left outer join CNT\_MISMATCH\_VLANs mv on mv.csr\_DEVICE\_NAME = ng.csr\_DEVICE\_NAME

group by ng.territory, ng.sub\_market, ng.market, csr\_Vendor, clli';

EXECUTE IMMEDIATE sqlstmt;

COMMIT;

EXCEPTION

WHEN OTHERS

THEN

MESSAGE :=

'Error: ' || methodname || '(): Cant generate cllil summary';

DBMS\_OUTPUT.put\_line (SUBSTR (MESSAGE || TO\_CHAR (SQLCODE) || ': ' || SQLERRM,1,255));

RAISE;

END;

load\_csr\_vlan\_regional\_summ();

END load\_vlan\_csr\_clli\_summary;

PROCEDURE load\_csr\_vlan\_regional\_summ

IS

methodname VARCHAR2 (30) := 'load\_vlan\_regional\_summary';

MESSAGE VARCHAR2 (500);

sqlstmt VARCHAR2 (32767);

BEGIN

EXECUTE IMMEDIATE 'truncate table SAM\_CSR\_VLAN\_REGION\_SUMM';

-- generate device summary

insert into SAM\_CSR\_VLAN\_REGION\_SUMM

columns (territory, sub\_market, ne\_total\_csr\_vlan, matched\_l\_csr\_vlan, matched\_nl\_csr\_vlan, mismatched\_csr\_vlan, gi\_nl\_2\_total\_vlan\_pct, vlan\_compliance)

select \* from (

select dlr.territory, dlr.sub\_market, sum(ne\_total\_csr\_vlan) ne\_total\_csr\_vlan, sum(matched\_l\_csr\_vlan) matched\_l\_csr\_vlan, sum(matched\_nl\_csr\_vlan) matched\_nl\_csr\_vlan

, sum(mismatched\_csr\_vlan) mismatched\_csr\_vlan, round(sum(NVL(matched\_nl\_csr\_vlan, 0)) / decode(sum(NVL(matched\_l\_csr\_vlan, 0)), 0, 1, sum(nvl(matched\_l\_csr\_vlan, 0)))

\* 100, 2) gi\_nl\_2\_total\_vlan\_pct, round (sum(nvl(matched\_l\_csr\_vlan,0)) / decode (sum(nvl( ne\_total\_csr\_vlan, 0)) - sum(nvl(matched\_nl\_csr\_vlan, 0)), 0 , 1

, sum(nvl(ne\_total\_csr\_vlan,0)) - sum(nvl(matched\_nl\_csr\_vlan,0)))\* 100 , 2) vlan\_compliance

from

SAM\_CSR\_VLAN\_CLLI\_SUMM dlr

group by rollup (dlr.territory, dlr.sub\_market)

ORDER BY territory, sub\_market)

where sub\_market != 'unknown' or sub\_market is null;

COMMIT;

sqlstmt := 'update SAM\_CSR\_VLAN\_REGION\_SUMM wk set sub\_market=''unknown'' where territory=''unknown''' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

sqlstmt := 'update SAM\_CSR\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = sub\_market where sub\_market is not null';

EXECUTE IMMEDIATE sqlstmt;

commit;

--sqlstmt := 'update SAM\_CSR\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = market\_territory where sub\_market is null and market\_territory is not null';

--EXECUTE IMMEDIATE sqlstmt;

--commit;

sqlstmt := 'update SAM\_CSR\_VLAN\_REGION\_SUMM wk set TERRITORY\_MARKET\_SUB = territory where sub\_market is null' ;

EXECUTE IMMEDIATE sqlstmt;

commit;

END load\_csr\_vlan\_regional\_summ;

END SAM\_CSR\_AUDIT;

/

--------------------------------------------------------

-- DDL for Package Body SITE\_PORTAL\_AUDIT

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."SITE\_PORTAL\_AUDIT" AS

PROCEDURE updatesiteportalaudit IS

sql\_stmt VARCHAR2(32000);

BEGIN

sql\_stmt := 'truncate table ' || audit\_table\_name;

--DBMS\_OUTPUT.put\_line (sql\_stmt);

BEGIN

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS

THEN

DBMS\_OUTPUT.put\_line

(SUBSTR ( 'Error: Could not truncate table: '

|| audit\_table\_name

|| 'Error: '

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

END;

BEGIN

sql\_stmt := 'insert into '

|| audit\_table\_name

|| ' (database\_name, server\_ip, group\_name,

site\_reference\_id, site\_name, address, city, state

, post\_code, site\_type, match\_code, match\_status, DOMAIN\_ID) '

|| ' WITH SITE\_PORTAL\_TMP AS

(select

distinct sp.region, sp.datbase\_name, sp.server\_ip, upper(sp.group\_name) as group\_name

, si.site\_reference\_id, si.site\_name, si.ADDRESS,

si.CITY, si.state, si.zip\_code as post\_code,

si.BASE\_NUM as SITE\_TYPE,

case when si.site\_reference\_id is not null and not regexp\_like(sp.group\_name, ''.\*-.\*'') and not regexp\_like(sp.group\_name, ''.\*\_.\*'') then ''BOTH''

ELSE ''NE Only''

end match\_code,

case

when si.site\_reference\_id is null and regexp\_like(sp.group\_name, ''.\*-.\*'') then ''Not in GI (Hyphen in Group Name Not Allowed)''

when si.site\_reference\_id is null and regexp\_like(sp.group\_name, ''.\*\_.\*'') then ''Not in GI (Underscore in Group Name Not Allowed)''

when si.site\_reference\_id is null then ''Not in GI''

when regexp\_like(sp.group\_name, ''.\*-.\*'') then ''Hyphen(-) in Group Name Not Allowed''

when regexp\_like(sp.group\_name, ''.\*\_.\*'') then ''Underscore(\_) in Group Name Not Allowed''

end match\_status,

sd.DOMAIN\_ID as DOMAIN\_ID

from SITE\_PORTAL\_WK sp

left outer join site si

on upper(sp.GROUP\_NAME) = upper(si.site\_name)

left join site\_portal\_domain\_mapping sd on

sd.SITE\_PORTAL\_DOMAIN\_CODE = sp.region where sp.group\_name is not null)

(select datbase\_name, server\_ip, group\_name,

site\_reference\_id, site\_name, address, city, state,

post\_code, site\_type, match\_code, match\_status,DOMAIN\_ID

from SITE\_PORTAL\_TMP)

'

;

dbms\_output.put\_line(sql\_stmt);

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error: Could not truncate table: '

|| audit\_table\_name

|| 'Error: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

COMMIT;

BEGIN

sql\_stmt := 'update '

|| audit\_table\_name

|| ' aud set DUPLICATE = ''Y'' '

|| 'where exists(select 1 from '

|| audit\_table\_name

|| ' i where i.group\_name = aud.group\_name and '

|| '(i.database\_name != aud.database\_name))';

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error: Could not update table: '

|| audit\_table\_name

|| 'Error: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

COMMIT;

END;

PROCEDURE updatesiteportalauditsummary IS

sql\_stmt VARCHAR2(32000);

BEGIN

sql\_stmt := 'truncate table ' || audit\_summary\_table\_name;

DBMS\_OUTPUT.put\_line (sql\_stmt);

BEGIN

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS

THEN

DBMS\_OUTPUT.put\_line

(SUBSTR ( 'Error: Could not truncate table: '

|| audit\_summary\_table\_name

|| 'Error: '

|| TO\_CHAR (SQLCODE)

|| ': '

|| SQLERRM,

1,

255

)

);

RAISE;

END;

BEGIN

sql\_stmt := 'insert into '

|| audit\_summary\_table\_name

|| ' ( TERRITORY, MARKET,SUB\_MARKET, MATCHED\_NT, NO\_MATCH\_IN\_NT, PER\_MATCHED, TOTAL)

select replace(TERRITORY, ''\_Grand'', ''Grand'') TERRITORY, MARKET\_TERRITORY,SUB\_MARKET,BOTH, NEONLY,PER\_MATCHED, count from (

select nvl(D.TERRITORY, ''\_Grand Total'') TERRITORY, D.MARKET\_TERRITORY ,SUB\_MARKET,

Sum(CASE WHEN MATCH\_CODE =''BOTH'' Then 1 else 0 end) both,

Sum(CASE WHEN MATCH\_CODE =''NE Only'' Then 1 else 0 end) NEOnly, 0 PER\_MATCHED, count(1) count from '

|| audit\_table\_name

|| ' aud

join domains\_regional\_reporting d

on d.domain\_id = aud.domain\_id

and MARKET\_TERRITORY != ''OSS''

group by rollup ( TERRITORY,MARKET\_TERRITORY,SUB\_MARKET) order by TERRITORY,MARKET\_TERRITORY,SUB\_MARKET

)

where MARKET\_TERRITORY != ''NNO'' or MARKET\_TERRITORY is null'

;

--dbms\_output.put\_line(sql\_stmt);

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error: Could not insert into table: '

|| audit\_summary\_table\_name

|| 'Error: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

BEGIN

sql\_stmt := 'update '

|| audit\_summary\_table\_name

|| ' set PER\_MATCHED = (MATCHED\_NT \* 100.00 / TOTAL)';

-- dbms\_output.put\_line(sql\_stmt);

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

sql\_stmt := 'update '

|| audit\_summary\_table\_name

|| ' wk set wk.TERRITORY\_MARKET\_SUB = SUB\_MARKET where sub\_market is not null';

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

sql\_stmt := 'update '

|| audit\_summary\_table\_name

|| ' wk set wk.TERRITORY\_MARKET\_SUB = MARKET where sub\_market is null and wk.market is not null';

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

sql\_stmt := 'update '

|| audit\_summary\_table\_name

|| ' wk set wk.TERRITORY\_MARKET\_SUB = TERRITORY where sub\_market is null and wk.market is null';

EXECUTE IMMEDIATE sql\_stmt;

COMMIT;

EXCEPTION

-- When NO\_DATA\_FOUND Then

-- dbms\_output.put\_line(SubStr('Error: No data found to update table: '|| AUDIT\_SUMMARY\_TABLE\_NAME ||

-- 'Error: '|| TO\_CHAR(SQLCODE)||': '||SQLERRM, 1, 255));

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error: Could not update table: '

|| audit\_summary\_table\_name

|| 'Error: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

END;

END;

PROCEDURE restore\_siteportalaudit IS

sql\_stmt VARCHAR2(32000);

BEGIN

sql\_stmt := 'insert into '

|| audit\_table\_name

|| ' (select \* from '

|| prod\_audit\_table\_name

|| ')';

EXECUTE IMMEDIATE sql\_stmt;

sql\_stmt := 'insert into '

|| audit\_summary\_table\_name

|| ' (select \* from '

|| prod\_audit\_summary\_table\_name

|| ')';

EXECUTE IMMEDIATE sql\_stmt;

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line(substr('Error: Could not restore SP audit table: '

|| audit\_table\_name

|| 'Error: '

|| TO\_CHAR(sqlcode)

|| ': '

|| sqlerrm,1,255) );

RAISE;

-- dbms\_output.put\_line();

END;

END site\_portal\_audit;

/

--------------------------------------------------------

-- DDL for Package Body XPERWEB

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."XPERWEB" is

procedure home (userName in varchar2 default NULL ) is

roUser varchar2(5);

begin

roUser := 'N';

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line('Not Found. ' || userName || ' is a not read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end;

if ( roUser = 'Y' ) then

-- read only user. Redirect to Nautilus Home.

dbms\_output.put\_line(userName || ' is a read only user. Redirect to Nautilus');

owa\_util.redirect\_url(curl => 'https://coavzw.nss.vzwnet.com/uuigui/' );

end if;

-- Default. Direct to Granite

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end if;

end home;

procedure path\_def (iPathInstId in varchar2, userName in varchar2 default NULL ) is

roUser varchar2(5);

trailId number;

pathInstId varchar2(100);

begin

pathInstId := iPathInstId;

roUser:= 'N';

trailId := 0;

dbms\_output.put\_line ('pathInstId=' || pathInstId || '; userName=' ||userName);

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line('Not Found. ' || userName || ' is a not read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.path\_def?iPathInstId='||pathInstId );

end;

if ( roUser = 'N' ) then

-- Not a Nautilus read only user. Redirect to Granite.

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.path\_def?iPathInstId='||pathInstId );

end if;

-- Nautilus read only user. Attempt to redirect to Nautilus

begin

select trail\_id into trailId from NG\_TOPOLOGY.TRAIL tr where TR.FR\_REF\_KEY\_NAME='GRANITE' and TR.FR\_REF\_KEY\_VALUE = pathInstId;

exception

WHEN NO\_DATA\_FOUND then

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/' );

end;

if ( trailId > 0 ) then

-- Redirect to Nautilus Archive Trail

dbms\_output.put\_line(userName || ' is read only user. Identified trail\_id=' || trailId || ' for corresponding Granite pathInstId=' || pathInstId );

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/uui/eapp/uiam/circuitDetailsv2?header=true&trailId=' || trailId ); --923207

end if;

else

--UserName not provided. Redirect to Granite.

dbms\_output.put\_line('userName is NULL');

end if;

--Default. Redirect to Granite.

dbms\_output.put\_line('Default. Nautilus object not found. Directing to Granite pathInstId=' || pathInstId );

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.path\_def?iPathInstId='||pathInstId ); -- 2465733');

end path\_def;

FUNCTION authorize return boolean is

begin

return true;

end authorize;

end xperweb;

/

--------------------------------------------------------

-- DDL for Package Body XWEB\_EQUIP

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."XWEB\_EQUIP" is

procedure home (userName in varchar2 default NULL ) is

roUser varchar2(5);

begin

roUser := 'N';

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line('Not Found. ' || userName || ' is a not read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end;

if ( roUser = 'Y' ) then

-- read only user. Redirect to Nautilus Home.

dbms\_output.put\_line(userName || ' is a read only user. Redirect to Nautilus');

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/' );

end if;

-- Default. Direct to Granite

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end if;

end home;

procedure equip\_def (ieqInstId in varchar2, userName in varchar2 default NULL ) is

roUser varchar2(5);

siteId number;

equipId number;

equipInstId number;

cntnr varchar2(50);

-- req utl\_http.req;

-- res utl\_http.resp;

-- url varchar2(4000);

-- buffer varchar2(4000);

-- content varchar2(4000);

begin

siteId := 0;

equipId := 0;

roUser := 'N';

cntnr := '';

equipInstId := ieqInstId;

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_equip.equip\_def?iEqInstId=' || equipInstId );

end;

if ( roUser = 'Y' ) then

-- read only user. Redirect to Nautilus Home.

dbms\_output.put\_line(userName || ' is a read only user. Redirect to Nautilus');

-- url:= 'https://nautilus-dev.nss.vzwnet.com/service-sidecar-server/location/search/v3'; -- use side car URL.

-- content := '{ "frRefKeyName": "GRANITE", "frRefKeyValues": [ "82442" ] }';

-- dbms\_output.put\_line(content);

-- dbms\_output.put\_line(url);

-- req := utl\_http.begin\_request(url, 'POST',' HTTP/1.1');

-- utl\_http.set\_header(req, 'user-agent', 'mozilla/4.0');

-- utl\_http.set\_header(req, 'content-type', 'application/json');

-- utl\_http.set\_header(req, 'Content-Length', length(content));

-- utl\_http.write\_text(req, content);

-- res := utl\_http.get\_response(req);

-- begin

-- loop

-- utl\_http.read\_line(res, buffer);

-- dbms\_output.put\_line(buffer);

-- end loop;

-- utl\_http.end\_response(res);

-- exception

-- when utl\_http.end\_of\_body

-- then

-- utl\_http.end\_response(res);

-- end;

begin

select site\_reference\_id, eqp\_reference\_id, CONTAINER into siteId, equipId, cntnr from XNG\_MIGRATION.EQUIPMENT xs where XS.FR\_REF\_KEY\_NAME='GRANITE' and XS.FR\_REF\_KEY\_VALUE = equipInstId;

exception

when NO\_DATA\_FOUND then

dbms\_output.put\_line('Nautilus Equipment not found. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/' );

end;

if ( siteId > 0 ) then

--equipDetailsv2?header=true&siteId=29270&equipContain=RACK&equipId=343004

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/uui/eapp/uiam/equipDetailsv2?header=true&siteId=' || siteId || '&equipContain=' || cntnr || '&equipId=' || equipId);

end if;

else

dbms\_output.put\_line( userName || ' is not a read only user.');

end if;

end if;

-- Default. Redirect to Granite

dbms\_output.put\_line('Default. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_equip.equip\_def?iEqInstId=' || equipInstId );

end equip\_def;

FUNCTION authorize return boolean is

begin

return true;

end authorize;

end XWEB\_EQUIP;

/

--------------------------------------------------------

-- DDL for Package Body XWEB\_SEG

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."XWEB\_SEG" is

procedure home (userName in varchar2 default NULL ) is

roUser varchar2(5);

begin

roUser := 'N';

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line('Not Found. ' || userName || ' is a not read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end;

if ( roUser = 'Y' ) then

-- read only user. Redirect to Nautilus Home.

dbms\_output.put\_line(userName || ' is a read only user. Redirect to Nautilus');

owa\_util.redirect\_url(curl => 'https://coavzw.nss.vzwnet.com/uuigui/' );

end if;

-- Default. Direct to Granite

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end if;

end home;

procedure segment\_def (iSegInstId in varchar2, userName in varchar2 default NULL ) is

roUser varchar2(5);

segId number;

segInstId number;

begin

segInstId := iSegInstId;

roUser:= 'N';

segId := 0;

dbms\_output.put\_line ('segInstId=' || segInstId || '; userName=' ||userName);

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line('Not Found. ' || userName || ' is a not read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_seg.segment\_def?iSegInstId='||segInstId );

end;

if ( roUser = 'N' ) then

-- Not a Nautilus read only user. Redirect to Granite.

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_seg.segment\_def?iSegInstId='||segInstId );

end if;

-- Nautilus read only user. Attempt to redirect to Nautilus

begin

select segment\_id into segId from NG\_TOPOLOGY.TRAIL\_SEGMENT tr where TR.FR\_REF\_KEY\_NAME='GRANITE' and TR.FR\_REF\_KEY\_VALUE = segInstId;

exception

WHEN NO\_DATA\_FOUND then

--owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_seg.segment\_def?isegInstId='||segInstId );

begin

select segment\_id into segId from NG\_TOPOLOGY.AR\_TRAIL\_SEGMENT tr where TR.FR\_REF\_KEY\_NAME='GRANITE' and TR.FR\_REF\_KEY\_VALUE = segInstId;

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/uui/eapp/uiam/segmentDetails?header=true&archiveSearch=true&segmentId=' || segId );

exception

WHEN NO\_DATA\_FOUND then

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/' );

end;

end;

if ( segId > 0 ) then

-- Redirect to Nautilus Archive Trail

dbms\_output.put\_line(userName || ' is read only user. Identified trail\_id=' || segId || ' for corresponding Granite segInstId=' || segInstId );

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/uui/eapp/uiam/segmentDetails?header=true&archiveSearch=false&segmentId=' || segId ); --923207

end if;

else

--UserName not provided. Redirect to Granite.

dbms\_output.put\_line('userName is NULL');

end if;

--Default. Redirect to Granite.

dbms\_output.put\_line('Default. Nautilus object not found. Directing to Granite segInstId=' || segInstId );

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_seg.segment\_def?iSegInstId='||segInstId ); -- 2465733');

end segment\_def;

FUNCTION authorize return boolean is

begin

return true;

end authorize;

end xweb\_seg;

/

--------------------------------------------------------

-- DDL for Package Body XWEB\_SITE

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE PACKAGE BODY "NG\_REPORTS"."XWEB\_SITE" is

procedure home (userName in varchar2 default NULL ) is

roUser varchar2(5);

begin

roUser := 'N';

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line('Not Found. ' || userName || ' is a not read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end;

if ( roUser = 'Y' ) then

-- read only user. Redirect to Nautilus Home.

dbms\_output.put\_line(userName || ' is a read only user. Redirect to Nautilus');

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/' );

end if;

-- Default. Direct to Granite

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xperweb.home' );

end if;

end home;

procedure site\_def (siteInstId in varchar2, userName in varchar2 default NULL ) is

roUser varchar2(5);

siteId number;

-- req utl\_http.req;

-- res utl\_http.resp;

-- url varchar2(4000);

-- buffer varchar2(4000);

-- content varchar2(4000);

begin

siteId := 0;

roUser := 'N';

if ( userName is not NULL ) then

begin

select RO into roUser from UUIAPP.GRANITE\_USERS where VZID=userName;

exception

WHEN NO\_DATA\_FOUND then

dbms\_output.put\_line(userName || ' is not a read only user. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_site.site\_def?siteInstId=' || siteInstId );

end;

if ( roUser = 'Y' ) then

-- read only user. Redirect to Nautilus Home.

dbms\_output.put\_line(userName || ' is a read only user. Redirect to Nautilus');

-- url:= 'https://nautilus-dev.nss.vzwnet.com/service-sidecar-server/location/search/v3'; -- use side car URL.

-- content := '{ "frRefKeyName": "GRANITE", "frRefKeyValues": [ "82442" ] }';

-- dbms\_output.put\_line(content);

-- dbms\_output.put\_line(url);

-- req := utl\_http.begin\_request(url, 'POST',' HTTP/1.1');

-- utl\_http.set\_header(req, 'user-agent', 'mozilla/4.0');

-- utl\_http.set\_header(req, 'content-type', 'application/json');

-- utl\_http.set\_header(req, 'Content-Length', length(content));

-- utl\_http.write\_text(req, content);

-- res := utl\_http.get\_response(req);

-- begin

-- loop

-- utl\_http.read\_line(res, buffer);

-- dbms\_output.put\_line(buffer);

-- end loop;

-- utl\_http.end\_response(res);

-- exception

-- when utl\_http.end\_of\_body

-- then

-- utl\_http.end\_response(res);

-- end;

begin

select site\_reference\_id into siteId from XNG\_MIGRATION.SITE xs where XS.FR\_REF\_KEY\_NAME='GRANITE' and XS.FR\_REF\_KEY\_VALUE = siteInstId;

exception

when NO\_DATA\_FOUND then

dbms\_output.put\_line('Nautilus Site not found. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_site.site\_def?siteInstId=' || siteInstId );

end;

if ( siteId > 0 ) then

owa\_util.redirect\_url(curl => 'https://coavzw-dev.nss.vzwnet.com/uuigui/uui/eapp/uiam/siteDetails?header=true&siteId=' || siteId );

end if;

else

dbms\_output.put\_line( userName || ' is not a read only user.');

end if;

end if;

-- Default. Redirect to Granite

dbms\_output.put\_line('Default. Redirect to Granite');

owa\_util.redirect\_url(curl => 'https://txslxngda1/pls/vzwnet\_stg/xweb\_site.site\_def?siteInstId=' || siteInstId );

end site\_def;

FUNCTION authorize return boolean is

begin

return true;

end authorize;

end xweb\_site;

/

--------------------------------------------------------

-- DDL for Function STRIP\_SEGMENT\_NAME

--------------------------------------------------------

CREATE OR REPLACE EDITIONABLE FUNCTION "NG\_REPORTS"."STRIP\_SEGMENT\_NAME" (SEGMENT\_NAME VARCHAR2)

RETURN VARCHAR2 IS

nob\_seg\_name varchar2(200);

new\_seg\_name varchar2(200);

BEGIN

--remove anything left of squiggly bracket

if (SEGMENT\_NAME like '%{%') then

nob\_seg\_name := rtrim(ltrim(substr(SEGMENT\_NAME,1,instr(SEGMENT\_NAME, '{',-1,1) - 1)));

else

nob\_seg\_name := SEGMENT\_NAME;

end if;

if nob\_seg\_name is null then

nob\_seg\_name := SEGMENT\_NAME;

end if;

if length(nob\_seg\_name) > 60 then

nob\_seg\_name := SEGMENT\_NAME;

end if;

-- translate various characters to blank. Then strip blanks

SELECT replace(translate(replace(nob\_seg\_name,'[ALLTEL]'), '-/,.()[]\*$;?#&~:=`<>\^@!"'||chr(160), ' '), ' ', '')

INTO new\_seg\_name

FROM dual;

RETURN new\_seg\_name;

END STRIP\_SEGMENT\_NAME;

/

--------------------------------------------------------

-- Constraints for Table NIS\_EXTRACTOR\_PARAM

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" ADD CONSTRAINT "NIS\_EXTRACTOR\_PARAM\_PK" PRIMARY KEY ("RULE\_ID", "PARAM\_INST\_ID", "PARAM\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" MODIFY ("PARAM\_NAME" CONSTRAINT "XP\_PARAM\_NAME\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" MODIFY ("PARAM\_TYPE" CONSTRAINT "XP\_PARAM\_TYPE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" MODIFY ("SYS\_CREATION\_DATE" CONSTRAINT "XP\_SYS\_CRE\_DATE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" MODIFY ("PARAM\_ID" CONSTRAINT "XP\_PARAM\_ID\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" MODIFY ("PARAM\_INST\_ID" CONSTRAINT "XP\_PARAM\_INST\_ID\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" MODIFY ("RULE\_ID" CONSTRAINT "XP\_RULE\_ID\_NN" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CRS\_DEVICE\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CRS\_DEVICE\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CRS\_DEVICE\_WK" MODIFY ("DEVICE\_IP" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CRS\_DEVICE\_WK" MODIFY ("DEVICE\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SEGMENT\_TERM\_EXP\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SEGMENT\_TERM\_EXP\_WK" MODIFY ("IS\_TERM\_EXPIRED" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEGMENT\_TERM\_EXP\_WK" MODIFY ("SEGMENT\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEGMENT\_TERM\_EXP\_WK" MODIFY ("SEGMENT\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_XCONN\_AUDIT\_SUMMARY

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_SUMMARY" MODIFY ("NG\_SITE\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_SUMMARY" MODIFY ("NG\_EQUIP\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_SUMMARY" MODIFY ("TID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_XCONN\_AUDIT\_SUMMARY" ADD CONSTRAINT "INC\_MSPP\_XCONN\_AUDIT\_SUMMA\_U01" UNIQUE ("TID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table NIS\_EXTRACTOR\_CONTROL

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("RUN\_STATUS" CONSTRAINT "XTC\_RUN\_STATUS\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("RUN\_DATE" CONSTRAINT "XTC\_RUN\_DATE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("REQUESTED\_EXEC\_DATE" CONSTRAINT "XTC\_REQ\_EXEC\_DATE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("SUBMIT\_DATE" CONSTRAINT "XTC\_REQ\_SUB\_DATE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("SYS\_CREATION\_DATE" CONSTRAINT "XTC\_SYS\_CRE\_DATE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("NODE\_ID" CONSTRAINT "XTC\_NODE\_ID\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("RULE\_ID" CONSTRAINT "XTC\_STATEMENT\_ID\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("CONTROL\_ID" CONSTRAINT "XTC\_RUN\_NO\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_CONTROL" MODIFY ("PARTITION\_ID" CONSTRAINT "XTC\_PARTITION\_ID\_NN" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CSR\_VLAN\_AUDIT\_ISSUES

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CSR\_VLAN\_AUDIT\_ISSUES" MODIFY ("CSR\_ISSUE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NCM\_CSR\_VLAN\_AUDITS

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_AUDITS" MODIFY ("CSR\_VENDOR" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_AUDITS" MODIFY ("MATCH\_CODE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SITE\_PORTAL\_DOMAIN\_MAPPING

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SITE\_PORTAL\_DOMAIN\_MAPPING" ADD CONSTRAINT "SITE\_PORTAL\_DOMAIN\_MAPPING\_PK" PRIMARY KEY ("SITE\_PORTAL\_DOMAIN\_MAPPING\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."SITE\_PORTAL\_DOMAIN\_MAPPING" MODIFY ("SITE\_PORTAL\_DOMAIN\_CODE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SITE\_PORTAL\_DOMAIN\_MAPPING" MODIFY ("DOMAIN\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SITE\_PORTAL\_DOMAIN\_MAPPING" MODIFY ("SITE\_PORTAL\_DOMAIN\_MAPPING\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table META\_DATA

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."META\_DATA" MODIFY ("UPDATE\_TYPE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NTLS\_CSR\_GIGE\_DELIVERY\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_GIGE\_DELIVERY\_WK" MODIFY ("SEG\_ORDER" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_GIGE\_DELIVERY\_WK" MODIFY ("SEGMENT\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_GIGE\_DELIVERY\_WK" MODIFY ("TRAIL\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table ERICSSON\_RRU

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."ERICSSON\_RRU" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."ERICSSON\_RRU" MODIFY ("RRU" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NOCC\_LAST\_TEST\_DETAILS

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_DETAILS" MODIFY ("DOMAIN\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_DETAILS" MODIFY ("DOMAIN\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_DETAILS" MODIFY ("SITE\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_DETAILS" MODIFY ("SITE\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NIS\_EXTRACTOR\_GRAPH

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" ADD CONSTRAINT "NIS\_EXTRACTOR\_GRAPH\_PK" PRIMARY KEY ("NODE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" MODIFY ("IS\_RECUR" CONSTRAINT "XG\_IS\_RECUR\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" MODIFY ("SYS\_CREATION\_DATE" CONSTRAINT "XG\_SYS\_CRE\_DATE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" MODIFY ("RULE\_ID" CONSTRAINT "XG\_RULE\_ID\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" MODIFY ("NODE\_ID" CONSTRAINT "XG\_NODE\_ID\_NN" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CRS\_PORT\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CRS\_PORT\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CRS\_PORT\_WK" MODIFY ("DEVICE\_IP" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CRS\_PORT\_WK" MODIFY ("HOST\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table PORT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."PORT" MODIFY ("PORT\_STATUS" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."PORT" MODIFY ("PORT\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."PORT" MODIFY ("TP\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."PORT" MODIFY ("PORT\_REFERENCE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SEVONE\_CSR

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SEVONE\_CSR" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEVONE\_CSR" MODIFY ("SOURCE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEVONE\_CSR" MODIFY ("DEVICE\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NCM\_NGMLS\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NCM\_NGMLS\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NCM\_NGMLS\_WK" MODIFY ("HOSTNAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table VZW\_NETWORK\_ORG

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."VZW\_NETWORK\_ORG" MODIFY ("TERRITORY" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."VZW\_NETWORK\_ORG" MODIFY ("MARKET\_TERRITORY" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."VZW\_NETWORK\_ORG" MODIFY ("MARKET" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."VZW\_NETWORK\_ORG" MODIFY ("REGION" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."VZW\_NETWORK\_ORG" MODIFY ("AREA" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."VZW\_NETWORK\_ORG" MODIFY ("INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_ADM\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_ADM\_NE\_VS\_NG\_AUDIT" MODIFY ("MATCH\_CODE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_EQPT\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_EQPT\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_SLOT\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_SLOT\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_SLOT\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CIENA\_6500\_NE\_EQUIP\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_EQUIP\_WK" ADD CONSTRAINT "CIENA\_6500\_NE\_EQUIP\_WK\_PK" PRIMARY KEY ("ID") DISABLE;

--------------------------------------------------------

-- Constraints for Table EQUIPMENT\_ATTRIBUTES

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."EQUIPMENT\_ATTRIBUTES" ADD CONSTRAINT "IDX\_EQ\_ATTRIBUTE\_ID" PRIMARY KEY ("EQP\_ATTRIBUTES\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table NCM\_CSR\_VLAN\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NCM\_CSR\_VLAN\_WK" MODIFY ("HOSTNAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table VSM\_DEVICES\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."VSM\_DEVICES\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NT\_SITERRA\_ANT\_EXTRACT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" ADD CONSTRAINT "NT\_SITERRA\_ANT\_EXTRACT\_PK" PRIMARY KEY ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "EQUIP\_INST\_ID", "NAURU\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" MODIFY ("NAURU\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" MODIFY ("ANTENNA\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" MODIFY ("EQUIP\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" MODIFY ("SITE\_DESCRIPTION" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" MODIFY ("SITE\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NTLS\_CSR\_PARSED\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_PARSED\_WK" ADD CONSTRAINT "NTLS\_CSR\_PARSED\_WK\_PK" PRIMARY KEY ("EQP\_REFERENCE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_PARSED\_WK" MODIFY ("EQP\_REFERENCE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NTLS\_CSR\_VLAN\_PATHS\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS\_WK" MODIFY ("EQP\_REFERENCE\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS\_WK" MODIFY ("PORT\_BANDWIDTH" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS\_WK" MODIFY ("PATH\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS\_WK" MODIFY ("VLAN\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_NE\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_NE\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table GEOPLAN\_DISCOVERY\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."GEOPLAN\_DISCOVERY\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."GEOPLAN\_DISCOVERY\_WK" MODIFY ("CELL\_NUMBER" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."GEOPLAN\_DISCOVERY\_WK" MODIFY ("CELL\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NG\_FW\_EQUIP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NG\_FW\_EQUIP" MODIFY ("EQP\_VENDOR" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_FW\_EQUIP" MODIFY ("EQP\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_FW\_EQUIP" MODIFY ("EQP\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_FW\_EQUIP" MODIFY ("EQ\_CLASS\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_FW\_EQUIP" MODIFY ("NE\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_IPADDR\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_IPADDR\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table VSM\_DEVICES\_DOMAIN\_MAP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."VSM\_DEVICES\_DOMAIN\_MAP" ADD CONSTRAINT "VSM\_DEVICES\_DOMAIN\_MAP\_PK" PRIMARY KEY ("VSM\_DEVICE\_NAME")

USING INDEX (CREATE INDEX "NG\_REPORTS"."VVDM\_UNQ\_IDX" ON "NG\_REPORTS"."VSM\_DEVICES\_DOMAIN\_MAP" ("VSM\_DEVICE\_NAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ) ENABLE;

--------------------------------------------------------

-- Constraints for Table ERICSSON\_ENM\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."ERICSSON\_ENM\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."ERICSSON\_ENM\_WK" MODIFY ("IP\_ADDRESS" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."ERICSSON\_ENM\_WK" MODIFY ("NODE\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK" MODIFY ("NG\_SITE\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK" MODIFY ("NG\_EQUIP\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INCMSPP\_XCONN\_AUDIT\_SUMMARY\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CIRC\_SEG\_NEW\_DECOM\_DETAIL

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("RPT\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("DOMAIN\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("DOMAIN\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("SEGMENT\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("SEGMENT\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("SEGMENT\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("Z\_SITE\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("SITE\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("ACTION\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_DETAIL" MODIFY ("COMMIT\_DATE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table HIGH\_CAPACITY\_PATH\_CHAN\_PARENT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_PARENT" MODIFY ("PATHPARENT\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_PARENT" MODIFY ("PATHPARENT\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_PROTECT\_PORT\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_PROTECT\_PORT\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_PROTECT\_PORT\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_LOGICAL\_PORT\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_LOGICAL\_PORT\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_LOGICAL\_PORT\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NORTEL\_DOM\_INVENTORY\_WRK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NORTEL\_DOM\_INVENTORY\_WRK" ADD PRIMARY KEY ("SEQ\_NUM")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table NT\_SITERRA\_SEG\_EXTRACT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" MODIFY ("SITE\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" ADD CONSTRAINT "NT\_SITERRA\_SEG\_EXTRACT\_PK" PRIMARY KEY ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "CIRC\_INST\_ID", "NAURU\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" MODIFY ("CIRCUIT\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" MODIFY ("NAURU\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" MODIFY ("CIRC\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" MODIFY ("SITE\_DESCRIPTION" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CLLI\_DOMAIN\_MAP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CLLI\_DOMAIN\_MAP" MODIFY ("GI\_LEAF\_DOMAIN\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CLLI\_DOMAIN\_MAP" MODIFY ("LEAF\_DOMAIN\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table JUNIPER\_DEVICES\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."JUNIPER\_DEVICES\_WK" ADD CONSTRAINT "JUNIPER\_DEVICES\_WK\_PK" PRIMARY KEY ("HOST\_NAME", "DEVICE\_IP", "DEVICE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."JUNIPER\_DEVICES\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."JUNIPER\_DEVICES\_WK" MODIFY ("DEVICE\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."JUNIPER\_DEVICES\_WK" MODIFY ("DEVICE\_IP" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."JUNIPER\_DEVICES\_WK" MODIFY ("HOST\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SITE

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SITE" ADD PRIMARY KEY ("SITE\_REFERENCE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."SITE" MODIFY ("SITE\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SITE" MODIFY ("SITE\_REFERENCE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN" MODIFY ("RPT\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_RPT\_GEN" MODIFY ("PROC\_START\_TS" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SEGMENTS\_MISSING\_EST\_COST

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST" ADD CONSTRAINT "SEGMENTS\_MISSING\_EST\_COST\_PK" PRIMARY KEY ("CIRC\_INST\_ID", "DOMAIN")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE( INITIAL 8388608 NEXT 8388608 MINEXTENTS 1

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST" MODIFY ("IS\_COST\_MISSING" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST" MODIFY ("CIRC\_HUM\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEGMENTS\_MISSING\_EST\_COST" MODIFY ("CIRC\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_OPTICAL\_LINK\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_OPTICAL\_LINK\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_OPTICAL\_LINK\_WK" MODIFY ("A\_TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NG\_ADM\_FW\_EQUIP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NG\_ADM\_FW\_EQUIP" MODIFY ("EQP\_VENDOR" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_ADM\_FW\_EQUIP" MODIFY ("EQP\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_ADM\_FW\_EQUIP" MODIFY ("EQP\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_ADM\_FW\_EQUIP" MODIFY ("EQ\_CLASS\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_ADM\_FW\_EQUIP" MODIFY ("NE\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_SNCNE\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_SNCNE\_WK" ADD CONSTRAINT "NETSMART\_SNCNE\_WK\_PK" PRIMARY KEY ("TID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NETSMART\_SNCNE\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_ADM\_NE\_VS\_NG\_ADT\_TMP" MODIFY ("MATCH\_CODE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table JV\_GLOBAL\_ID

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."JV\_GLOBAL\_ID" MODIFY ("GLOBAL\_ID\_PK" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table MSPP\_SUMMARY\_REGION\_MAP\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."MSPP\_SUMMARY\_REGION\_MAP\_WK" MODIFY ("NG\_EQUIP\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."MSPP\_SUMMARY\_REGION\_MAP\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table VDDM

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."VDDM" MODIFY ("LEAF\_DOMAIN\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NT\_SITERRA\_SITE\_EXTRACT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT" ADD CONSTRAINT "NT\_SITERRA\_SITE\_EXTRACT\_PK" PRIMARY KEY ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "NAURU\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT" MODIFY ("NAURU\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT" MODIFY ("SITE\_DESCRIPTION" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT" MODIFY ("SITE\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NOCC\_LAST\_TEST\_SUMMARY

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY" MODIFY ("TERRITORY" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY" MODIFY ("MARKET\_TERRITORY" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY" MODIFY ("SUB\_MARKET" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY" MODIFY ("DOMAIN\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY" MODIFY ("DOMAIN\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NOCC\_LAST\_TEST\_SUMMARY" MODIFY ("RUN\_DATE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table LUCENT\_AUDIT\_SWITCH\_SUMM

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."LUCENT\_AUDIT\_SWITCH\_SUMM" MODIFY ("LEAF\_DOMAIN\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."LUCENT\_AUDIT\_SWITCH\_SUMM" MODIFY ("MARKET\_TERRITORY" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table JUNIPER\_VLANS\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."JUNIPER\_VLANS\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."JUNIPER\_VLANS\_WK" MODIFY ("VLAN\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."JUNIPER\_VLANS\_WK" MODIFY ("DEVICE\_IP" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."JUNIPER\_VLANS\_WK" MODIFY ("HOST\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_CONNECTION\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_CONNECTION\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_CONNECTION\_WK" MODIFY ("CIRCUIT\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM" MODIFY ("CLLI\_6" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM" MODIFY ("LEAF\_DOMAIN\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NCM\_HPOV\_EH\_NT\_CSR\_CLLI\_SUM" MODIFY ("MARKET" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NCM\_CSR\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NCM\_CSR\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NCM\_CSR\_WK" MODIFY ("HOSTNAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SEVONE\_PERFORMANCE\_DATA

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SEVONE\_PERFORMANCE\_DATA" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEVONE\_PERFORMANCE\_DATA" MODIFY ("DEVICE\_IP" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEVONE\_PERFORMANCE\_DATA" MODIFY ("DEVICE\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."SEVONE\_PERFORMANCE\_DATA" MODIFY ("DEVICE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table GEOPLAN\_SMALLCELL\_DISCOVERY\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."GEOPLAN\_SMALLCELL\_DISCOVERY\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table VZW\_ERROR\_DETAILS

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."VZW\_ERROR\_DETAILS" MODIFY ("LOG\_DATE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table EBH\_DISCONNECT\_DETAILS

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."EBH\_DISCONNECT\_DETAILS" MODIFY ("DF\_SEG\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_SYS\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_SYS\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table MSPP\_SUMMARY\_REGION\_MAP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."MSPP\_SUMMARY\_REGION\_MAP" MODIFY ("NG\_SITE\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."MSPP\_SUMMARY\_REGION\_MAP" MODIFY ("NG\_EQUIP\_INST\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."MSPP\_SUMMARY\_REGION\_MAP" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NTLS\_UPDATER

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NTLS\_UPDATER" MODIFY ("UPDATE\_TYPE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NORTEL\_DOM\_INVENTORY

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NORTEL\_DOM\_INVENTORY" ADD PRIMARY KEY ("SEQ\_NUM")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table NGMLS\_VLAN\_AUDIT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NGMLS\_VLAN\_AUDIT" MODIFY ("NGMLS\_VENDOR" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NGMLS\_VLAN\_AUDIT" MODIFY ("MATCH\_CODE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_NE\_VS\_NG\_AUDIT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_NE\_VS\_NG\_AUDIT" MODIFY ("MATCH\_CODE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_NE

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_NE" ADD CONSTRAINT "INC\_MSPP\_NE\_U01" UNIQUE ("TID", "GNE\_TID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_NE" ADD CONSTRAINT "INC\_MSPP\_NE\_C01" CHECK (TID is not null) ENABLE;

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_NE" ADD CONSTRAINT "INC\_MSPP\_NE\_PK" PRIMARY KEY ("NEID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table CSR\_ISSUE

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CSR\_ISSUE" MODIFY ("CSR\_ISSUE\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CSR\_ISSUE" MODIFY ("DESCRIPTION" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CSR\_ISSUE" MODIFY ("IS\_CRITICAL" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CSR\_ISSUE" ADD PRIMARY KEY ("CSR\_ISSUE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."CSR\_ISSUE" ADD UNIQUE ("DESCRIPTION")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table RRH\_CPRI

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."RRH\_CPRI" MODIFY ("REQUEST\_TIME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."RRH\_CPRI" MODIFY ("REQUEST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NIS\_EXTRACTOR

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR" ADD CONSTRAINT "NIS\_EXTRACTOR\_PK" PRIMARY KEY ("RULE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR" MODIFY ("RULE\_DEFINITION" CONSTRAINT "IE\_RULE\_DEF\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR" MODIFY ("SYS\_CREATION\_DATE" CONSTRAINT "IE\_SYS\_CRE\_DATE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR" MODIFY ("RULE\_TYPE" CONSTRAINT "IE\_RULE\_TYPE\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR" MODIFY ("RULE\_NAME" CONSTRAINT "IE\_RULE\_NAME\_NN" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR" MODIFY ("RULE\_ID" CONSTRAINT "IE\_RULE\_ID\_NN" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_XCONNECT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_XCONNECT" ADD CONSTRAINT "NETSMART\_XCONNECT\_PK" PRIMARY KEY ("INST\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NETSMART\_XCONNECT" MODIFY ("SNCNE\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NETSMART\_XCONNECT" MODIFY ("INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table DOMAIN\_PARENTS

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."DOMAIN\_PARENTS" MODIFY ("DOMAIN\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CLLI\_DOMAINS\_MAP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CLLI\_DOMAINS\_MAP" ADD CONSTRAINT "CLLI\_DOMAIN\_MAP\_PK" PRIMARY KEY ("CLLI", "LEAF\_DOMAIN\_INST\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."CLLI\_DOMAINS\_MAP" MODIFY ("LEAF\_DOMAIN\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table TEMP\_NT\_SITERRA\_ENB

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."TEMP\_NT\_SITERRA\_ENB" MODIFY ("SITE\_DESCRIPTION" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."TEMP\_NT\_SITERRA\_ENB" MODIFY ("SITE\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table EQUIPMENT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."EQUIPMENT" ADD CONSTRAINT "EQP\_REFERENCE\_ID" PRIMARY KEY ("EQP\_REFERENCE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table NTLS\_CSR\_GIGE\_TRAILS\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_GIGE\_TRAILS\_WK" MODIFY ("PORT\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table PORT\_OLD

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."PORT\_OLD" MODIFY ("PORT\_STATUS" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."PORT\_OLD" MODIFY ("PORT\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."PORT\_OLD" MODIFY ("TP\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."PORT\_OLD" MODIFY ("PORT\_REFERENCE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NCM\_NGMLS\_VLAN\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NCM\_NGMLS\_VLAN\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NCM\_NGMLS\_VLAN\_WK" MODIFY ("HOSTNAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_EQPT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_EQPT" ADD CONSTRAINT "NETSMART\_EQPT\_PK" PRIMARY KEY ("INST\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NETSMART\_EQPT" MODIFY ("SNCNE\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NETSMART\_EQPT" MODIFY ("INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CRS\_BANDWIDTH\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CRS\_BANDWIDTH\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CRS\_BANDWIDTH\_WK" MODIFY ("DEVICE\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NTLS\_CSR\_VLAN\_PATHS

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS" MODIFY ("VLAN\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS" MODIFY ("PATH\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS" MODIFY ("PORT\_BANDWIDTH" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NTLS\_CSR\_VLAN\_PATHS" MODIFY ("EQP\_REFERENCE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table MOTO\_CDMA\_T1\_INV

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."MOTO\_CDMA\_T1\_INV" MODIFY ("VSM\_DEVICE\_NAME\_OMCR" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table TEST\_HEAD\_XREF

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."TEST\_HEAD\_XREF" MODIFY ("TEST\_HEAD\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."TEST\_HEAD\_XREF" MODIFY ("STATUS" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."TEST\_HEAD\_XREF" ADD CONSTRAINT "TEST\_HEAD\_XREF\_PK" PRIMARY KEY ("TEST\_HEAD\_NAME")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table NG\_EQUIPMENT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NG\_EQUIPMENT" ADD CONSTRAINT "NG\_EQUIPMENT\_PK" PRIMARY KEY ("EQP\_REFERENCE\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NG\_EQUIPMENT" MODIFY ("SITE\_REFERENCE\_ID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_EQUIPMENT" MODIFY ("INV\_STATUS" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_EQUIPMENT" MODIFY ("EQP\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_EQUIPMENT" MODIFY ("CONTAINER" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_EQUIPMENT" MODIFY ("EQP\_NAME" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NG\_EQUIPMENT" MODIFY ("EQP\_REFERENCE\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CIENA\_6500\_DISC\_METADATA

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CIENA\_6500\_DISC\_METADATA" ADD CONSTRAINT "CIENA\_6500\_DISC\_METADATA\_PK" PRIMARY KEY ("COMMAND\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table JV\_CDO\_CLASS

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."JV\_CDO\_CLASS" MODIFY ("CDO\_CLASS\_PK" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SAM\_CSR\_VLAN\_AUDIT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SAM\_CSR\_VLAN\_AUDIT" MODIFY ("CSR\_VENDOR" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table HIGH\_CAPACITY\_PATH\_CHAN\_CHILD

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_CHILD" MODIFY ("PATH\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN\_CHILD" MODIFY ("PATH\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table DIR\_INV\_DEF\_ATTRIBUTE

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."DIR\_INV\_DEF\_ATTRIBUTE" ADD CONSTRAINT "DIR\_INV\_DEF\_ATTRIBUTE\_PK" PRIMARY KEY ("INV\_DEF\_ATTR\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."DIR\_INV\_DEF\_ATTRIBUTE" MODIFY ("DATA\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."DIR\_INV\_DEF\_ATTRIBUTE" MODIFY ("INV\_DEF\_ATTR\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SITE\_ATTRIBUTES

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."SITE\_ATTRIBUTES" ADD CONSTRAINT "IDX\_SITE\_ATTRIBUTE\_ID" PRIMARY KEY ("SITE\_ATTRIBUTES\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

--------------------------------------------------------

-- Constraints for Table CIRC\_SEG\_NEW\_DECOM\_SUMMARY

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("PERIOD" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("PERIOD\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("COUNT\_NEW" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("COUNT\_DECOM" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("COUNT\_NEW\_TOTAL" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("COUNT\_NEW\_BY\_TESS" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("COUNT\_NEW\_NOT\_TESS\_CONTRACT" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("COUNT\_NEW\_NOT\_TESS\_NO\_CONTRACT" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."CIRC\_SEG\_NEW\_DECOM\_SUMMARY" MODIFY ("PERCENTAGE\_NEW\_BY\_TESS" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table HIGH\_CAPACITY\_PATH\_CHAN

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN" MODIFY ("PATHPARENT\_BANDWIDTH" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN" MODIFY ("PATHPARENT\_TYPE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."HIGH\_CAPACITY\_PATH\_CHAN" MODIFY ("PATHPARENT\_NAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table TEMP\_NT\_SITERRA\_CSR

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."TEMP\_NT\_SITERRA\_CSR" MODIFY ("SITE\_INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table JV\_COMMIT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."JV\_COMMIT" MODIFY ("COMMIT\_PK" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_SNCNE

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_SNCNE" ADD CONSTRAINT "NETSMART\_SNCNE\_PK" PRIMARY KEY ("INST\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 8388608 NEXT 8388608 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TS\_NGREPORTS" ENABLE;

ALTER TABLE "NG\_REPORTS"."NETSMART\_SNCNE" MODIFY ("TID" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."NETSMART\_SNCNE" MODIFY ("INST\_ID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table BRIX\_CSR\_DISCOVERY\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."BRIX\_CSR\_DISCOVERY\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_RACK\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_RACK\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_RACK\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NETSMART\_NE\_VS\_NG\_AUDIT\_TEMP" MODIFY ("MATCH\_CODE" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table INC\_MSPP\_PHYSICAL\_PORT\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_PHYSICAL\_PORT\_WK" MODIFY ("SYS\_CREATION\_DATE" NOT NULL ENABLE);

ALTER TABLE "NG\_REPORTS"."INC\_MSPP\_PHYSICAL\_PORT\_WK" MODIFY ("TID" NOT NULL ENABLE);

--------------------------------------------------------

-- Ref Constraints for Table CIENA\_6500\_NE\_PARSED\_DATA\_WK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."CIENA\_6500\_NE\_PARSED\_DATA\_WK" ADD CONSTRAINT "FK\_CIENA\_6500\_CMD\_ID" FOREIGN KEY ("COMMAND\_ID")

REFERENCES "NG\_REPORTS"."CIENA\_6500\_DISC\_METADATA" ("COMMAND\_ID") ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table NIS\_EXTRACTOR\_GRAPH

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_GRAPH" ADD CONSTRAINT "NIS\_EXTRACTOR\_GRAPH\_1FK" FOREIGN KEY ("RULE\_ID")

REFERENCES "NG\_REPORTS"."NIS\_EXTRACTOR" ("RULE\_ID") ON DELETE CASCADE ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table NIS\_EXTRACTOR\_PARAM

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NIS\_EXTRACTOR\_PARAM" ADD CONSTRAINT "NIS\_EXTRACTOR\_PARAM\_1FK" FOREIGN KEY ("RULE\_ID")

REFERENCES "NG\_REPORTS"."NIS\_EXTRACTOR" ("RULE\_ID") ON DELETE CASCADE ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table NORTEL\_PORT\_INVENTORY

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NORTEL\_PORT\_INVENTORY" ADD CONSTRAINT "DOM\_REF" FOREIGN KEY ("DOM\_SEQ\_NUM")

REFERENCES "NG\_REPORTS"."NORTEL\_DOM\_INVENTORY" ("SEQ\_NUM") ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table NORTEL\_PORT\_INVENTORY\_WRK

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NORTEL\_PORT\_INVENTORY\_WRK" ADD CONSTRAINT "DOM\_SEQ\_REF" FOREIGN KEY ("DOM\_SEQ\_NUM")

REFERENCES "NG\_REPORTS"."NORTEL\_DOM\_INVENTORY\_WRK" ("SEQ\_NUM") ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table NT\_SITERRA\_ANT\_EXTRACT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_ANT\_EXTRACT" ADD CONSTRAINT "NT\_SITERRA\_ANT\_EXTRACT\_FK" FOREIGN KEY ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "NAURU\_ID")

REFERENCES "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT" ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "NAURU\_ID") ON DELETE CASCADE ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table NT\_SITERRA\_SEG\_EXTRACT

--------------------------------------------------------

ALTER TABLE "NG\_REPORTS"."NT\_SITERRA\_SEG\_EXTRACT" ADD CONSTRAINT "NT\_SITERRA\_SEG\_EXTRACT\_FK" FOREIGN KEY ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "NAURU\_ID")

REFERENCES "NG\_REPORTS"."NT\_SITERRA\_SITE\_EXTRACT" ("SITE\_INST\_ID", "SITE\_DESCRIPTION", "NAURU\_ID") ON DELETE CASCADE ENABLE;