Overview

If your district uses the Geocode Map method and address validation, before you run Year-End Processing, make sure geocodes are set up correctly for your district and all grades in the district are mapped to a school for each geocode.

Use the SQL queries in this technical bulletin to identify possible issues with the geocode map.

- Identifying Geocode/Grade Combinations not Mapped. See page 1.
- Identifying Students with Projected School Set to “None”. See page 2.
- Identifying Students Set to Project to Requested School but not Projected to Any School. See page 2.
- Identifying Students Set to Project to Zoned School but not Projected to Any School. See page 3.

NOTE

Before running these queries, run Next-Year Preparation at least once.

For further information about address validation and geocodes, see the Chancery SMS Address Validation Guide. For further information about Year-End Processing, see the Chancery SMS Next-Year Preparation and Year-End Processing Guide.

Identifying Geocode/Grade Combinations not Mapped

If this query returns any results, then the geocode/grade levels returned do not have an associated school map and should be corrected.

To identify geocode/grade combinations that are not mapped to any school:

1. From the Start menu, navigate to All Programs > Microsoft SQL Server > Query Analyzer.
2. Connect to the database.
3. Enter the following script:

```
CREATE TABLE #DISTRICT_GRADES
(
ID_SET_GRADE_LEVEL INT,
GRADE_ORDER SMALLINT
)

-- this temporary table makes a list of only the grades that are in use
-- in at least one school in the district

insert into #DISTRICT_GRADES
select distinct GL.ID_SET_GRADE_LEVEL, GL.GRADE_ORDER from CSL_SET_GRADE_LEVEL GL
inner join CSL_SET_GRADE G on
GL.ID_SET_GRADE_LEVEL = G.ID_SET_GRADE_LEVEL
where G.GRADE_USED_IN_SCHOOL = 'Y'
order by GL.ID_SET_GRADE_LEVEL

-- the CSL_VW_GEOCODE_RANGE view shows the mapping between geocodes, grade levels
-- and schools. This select will find any district used grade that is not mapped to
-- a school for any of the Geocodes.

select G.GEOCODE_CODE, SGL.GRADE_LEVEL_CODE,SGL.GRADE_LEVEL_DESC from
CSL_SET_GEOCODE G, #DISTRICT_GRADES DG
inner join CSL_SET_GRADE_LEVEL SGL (nolock)
on SGL.ID_SET_GRADE_LEVEL = DG.ID_SET_GRADE_LEVEL
where not exists
```

```
( select 1 from CSL_VW_GEOCODE_RANGE GR
inner join CSL_SET_GRADE_LEVEL GL_MIN on GL_MIN.ID_SET_GRADE_LEVEL =
GR.ID_SET_GRADE_LEVEL_MIN
inner join CSL_SET_GRADE_LEVEL GL_MAX on GL_MAX.ID_SET_GRADE_LEVEL =
GR.ID_SET_GRADE_LEVEL_MAX
where GR.ID_SET_GEOCODE = G.ID_SET_GEOCODE
and GL_MIN.GRADE_ORDER <= DG.GRADE_ORDER
and GL_MAX.GRADE_ORDER >= DG.GRADE_ORDER
)
order by GEOCODE_CODE, SGL.GRADE_ORDER

drop table #DISTRICT_GRADES
```

4 Execute the query.

Identifying Students with Projected School Set to “None”

Although some students should not be projected to a school for the next year, verify that all the students in the query results are correctly set up to not project to a school.

To identify students whose projected school is set to “None”:

1 From the Start menu, navigate to All Programs > Microsoft SQL Server > Query Analyzer.

2 Connect to the database.

3 Enter the following script:

```
select SSI.SCHOOL_DESC,DD.STD_NUMBER,DS.ID_STD_DEMO_SCHOOL from CSL_STD_DEMO_SCHOOL
DS (NOLOCK)
inner join TMP_NYP_STUDENT_INFO SI (nolock) on DS.ID_STD_DEMO_SCHOOL =
SI.ID_STD_DEMO_SCHOOL
inner join CSL_STD_DEMO_DISTRICT DD (nolock) on DS.ID_STD_DEMO_DISTRICT =
DD.ID_STD_DEMO_DISTRICT
inner join CSL_DIS_DISTRICT_INFO DI (nolock) on ID_DIS_DISTRICT_INFO = 0
inner join CSL_SET_SCHOOL_INFO SSI (nolock) on SSI.ID_SET_SCHOOL_INFO =
DS.ID_SET_SCHOOL_INFO

where
    SI.NEXT_SCHOOL_INFO_ID is null AND
    ((SI.GRADUATION_YEAR =   DI.SCHOOL_YEAR  ) OR (SI.GRADUATION_YEAR IS NULL)) and
    DS.PROJECTED_SCHOOL_CHOICE = 2
    order by SSI.SCHOOL_DESC
```

4 Execute the query.

Identifying Students Set to Project to Requested School but not Projected to Any School

This query identifies students whose projected school is “Use requested school” but who are not being projected to a school in the planning calendar. Examine the projections for the students in the query results to determine the reason they are not being projected to a school.

To identify students who have a requested school but are not projected to any school:

1 From the Start menu, navigate to All Programs > Microsoft SQL Server > Query Analyzer.

2 Connect to the database.

3 Enter the following script:

```
select SSI.SCHOOL_DESC, DD.STD_NUMBER, DS.ID_STD_DEMO_SCHOOL from
CSL_STD_DEMO_SCHOOL DS (NOLOCK)
inner join TMP_NYP_STUDENT_INFO SI (nolock) on DS.ID_STD_DEMO_SCHOOL =
SI.ID_STD_DEMO_SCHOOL
inner join CSL_STD_DEMO_DISTRICT DD (nolock) on DS.ID_STD_DEMO_DISTRICT =
DD.ID_STD_DEMO_DISTRICT
inner join CSL_DIS_DISTRICT_INFO DI (nolock) on ID_DIS_DISTRICT_INFO = 0
inner join CSL_SET_SCHOOL_INFO SSI (nolock) on SSI.ID_SET_SCHOOL_INFO =
DS.ID_SET_SCHOOL_INFO
where
    SI.NEXT_SCHOOL_INFO_ID is null AND
    ((SI.GRADUATION_YEAR = DI.SCHOOL_YEAR ) OR (SI.GRADUATION_YEAR IS NULL))
and
    DS.PROJECTED_SCHOOL_CHOICE = 3
    order by SSI.SCHOOL_DESC
```

4 Execute the query.

Identifying Students Set to Project to Zoned School but not Projected to Any School

This query identifies students whose projected school choice is “User zoned school” but who are not projected to the planning calendar. Typically, this is because the students do not have a geocode for the planning calendar, their geocode is invalid, or their geocode is not mapped to a school for their grade level.

To identify students who are set to project to their zoned school but are not projected to any school:

1 From the Start menu, navigate to All Programs > Microsoft SQL Server > Query Analyzer.

2 Connect to the database.

3 Enter the following script:

```
select SSI.SCHOOL_DESC, DD.STD_NUMBER, DS.ID_STD_DEMO_SCHOOL from
CSL_STD_DEMO_SCHOOL DS (NOLOCK)
inner join TMP_NYP_STUDENT_INFO SI (nolock) on DS.ID_STD_DEMO_SCHOOL =
SI.ID_STD_DEMO_SCHOOL
inner join CSL_STD_DEMO_DISTRICT DD (nolock) on DS.ID_STD_DEMO_DISTRICT =
DD.ID_STD_DEMO_DISTRICT
inner join CSL_DIS_DISTRICT_INFO DI (nolock) on ID_DIS_DISTRICT_INFO = 0
inner join CSL_SET_SCHOOL_INFO SSI (nolock) on SSI.ID_SET_SCHOOL_INFO =
DS.ID_SET_SCHOOL_INFO

where
    SI.NEXT_SCHOOL_INFO_ID is null AND
    ((SI.GRADUATION_YEAR = DI.SCHOOL_YEAR ) OR (SI.GRADUATION_YEAR IS NULL))
and
    DS.PROJECTED_SCHOOL_CHOICE = 1
    order by SSI.SCHOOL_DESC
```

4 Execute the query.

If the above query identifies any students, use the following subqueries to determine if the students’ geocodes are mapped correctly for their grade level.

Identifying Students' Next-Year Grade Levels

1. From the Start menu, navigate to All Programs > Microsoft SQL Server > Query Analyzer.
2. Connect to the database.
3. Enter the following script:

```
select DS.PROJECTED_SCHOOL_CHOICE, DD.ID_SET_GEOCODE_PLANNING,
NGL.ID_SET_GRADE_LEVEL_NEXT
from CSL_STD_DEMO_SCHOOL DS
inner join CSL_STD_DEMO_DISTRICT DD on
       DS.ID_STD_DEMO_DISTRICT = DD.ID_STD_DEMO_DISTRICT
inner join CSL_SET_GRADE G on
       G.ID_SET_GRADE = DS.ID_SET_GRADE
inner join CSL_SET_GRADE_LEVEL GL on
       G.ID_SET_GRADE_LEVEL = GL.ID_SET_GRADE_LEVEL
inner join CSL_VW_GET_NEXT_GRADE_LEVEL NGL on
       GL.ID_SET_GRADE_LEVEL = NGL.ID_SET_GRADE_LEVEL
   WHERE DS.ID_STD_DEMO_SCHOOL = <xxxx>

-- where <xxxx> = a student demo school ID from the list of students returned in
the previous query.
```

4. Execute the query.

Identifying Students' Assigned Geocodes

1. From the Start menu, navigate to All Programs > Microsoft SQL Server > Query Analyzer.
2. Connect to the database.
3. Enter the following script:

```
select GRADE_ORDER from CSL_SET_GRADE_LEVEL where CSL_SET_GRADE_LEVEL=
<ID_SET_GRADE_LEVEL_NEXT>

--where <ID_SET_GRADE_LEVEL_NEXT> is the student's next-year grade level
returned in the previous query.
```

4. Execute the query.

Determining if Students' Geocodes for their Next-Year Grade Level are Mapped Correctly

1. From the Start menu, navigate to All Programs > Microsoft SQL Server > Query Analyzer.
2. Connect to the database.
3. Enter the following script:

```
select * from CSL_VW_GEOCODE_RANGE GR
inner join CSL_SET_GRADE_LEVEL GL_MIN on  GL_MIN.ID_SET_GRADE_LEVEL =
GR.ID_SET_GRADE_LEVEL_MIN
inner join CSL_SET_GRADE_LEVEL GL_MAX on  GL_MAX.ID_SET_GRADE_LEVEL =
GR.ID_SET_GRADE_LEVEL_MAX
where GR.ID_SET_GEOCODE = <ID_SET_GEOCODE_PLANNING>
```

```
and GL_MIN.GRADE_ORDER <= <GRADE_ORDER>
and GL_MAX.GRADE_ORDER >= <GRADE_ORDER>


-- where <ID_SET_GEOCODE_PLANNING> is the student's assigned geocode identified
-- in the previous query.
-- and where <GRADE_ORDER> is the student's grade order identified in the
-- previous query.
```

4 Execute the query.