Please note that the section numbers are the numbers form the User guild and not the design document.

Section 3.1

3.1.1

This is a view of the system rolled up breaks at distance breaks.

You can test this from toad/sql developer by running:

Select \* from xaa\_route where route\_name = ‘MY\_ROUTE’

I would spot check some tricky areas that you know has district breaks and ensure you see the breaks were you would expect to see them.

3.1.2

This is a view that contains the geometry of the route network broken up by Road Name.

You can view the data but doing:

Select \* from xaa\_route\_sdo

I would expect this to be correct since it is based directly on a core view. I can’t think of a good way to test this individually. Perhaps making a GIS theme of it and seeing if it is exact to your route network.

3.1.2

This table shows spatial related changes. It shows 2 operations. ADD and DELETE. If something was modified it should show both and add and delete.

To Test:

Run:

Select \* from xaa\_spatial\_audit where route\_id = 00000000

The 00000000 would be the NE\_ID of the route you are going to change.

Once you have a baseline of the data for that ne\_id, then change the spatial representation of that route, either with Spatial Manager or with the forms.

Rerun the query and see if your items show up. You might want to run the report in 3.4 to check the lengths of the segments if that you help you to confirm. I’ll go over that a bit more in when I talk about section 3.4

3.2

Event Information.

This is the part that I believe needs the most amount of testing. The premise is that you fill in an input table (xaa\_loc\_ident) and then run a procedure and see the results in the output table (xaa\_net\_ref).

The output table is wiped every time the procedure is run.

Ideally you would put in an input, run the function and receive one of the following results:

1. Old Route Location Not Found
   1. This can occur if the input offsets never match historic Exor values or if the input offsets were never continuous, i.e. there was a gap in the system in between those points.
2. Route Location Not Changed
   1. If the route location offsets and name have remained the same, this message will be generated. It is possible that the date has changed though.
3. Route Location Changed
   1. This message is generated if the name changes or if the mile points of the offsets are no longer the same as the input offsets.
4. Route Location Closed
   1. If the route location has been end dated in the Exor system then this process message is displayed.

If the route length changed from the historic date the assets that use to be there would need to be changed/move in the AgileAssets system for example.

Testing: I would Approach testing this in two different ways.

1. Getting the program to display each message
   1. Old Route Location insert into XAA\_LOC\_IDENT some bogus information for:
      1. Historic\_date Some real date in the past
      2. Loc\_Ident for testing real items I have been putting the NE\_ID here for my own reference, in reality this would be a number form the Agile system, you can put any number for testing if you would like.
      3. Route\_Name Some false route name
      4. Offset\_from Some false Begin Mileage
      5. Offset\_to Some false End Mileage
      6. Source\_table Some false Agile table name
   2. Old Route Location insert into XAA\_LOC\_IDENT some real information for:
      1. Historic\_date Some real date in the past
      2. Loc\_Ident for testing real items I have been putting the NE\_ID.
      3. Route\_Name Some real route name
      4. Offset\_from Some real Begin Mileage
      5. Offset\_to Some End Mileage that would not have existed
      6. Source\_table Some fake Agile table name
2. Route Location Not Changed
   1. For this item I would put information that you know hasn’t change
3. Route Location Changed
   1. This is the item that should require the most testing. I would put in the information for an area you plan to change:
      1. Historic\_date Some real date in the past
      2. Loc\_Ident for testing real items I have been putting the NE\_ID.
      3. Route\_Name The route name you plan on changing
      4. Offset\_from Begin Mileage as it is before the change
      5. Offset\_to End Mileage as it is before the change
      6. Source\_table Some fake Agile table name
      7. ---
      8. Now change the data either using forms or spatial manager so that there is an INCREASE of length between the mile points you gave.
   2. For Decreasing. I would put in the information for an area you plan to change:
      1. Historic\_date Some real date in the past
      2. Loc\_Ident for testing real items I have been putting the NE\_ID.
      3. Route\_Name The route name you plan on changing
      4. Offset\_from Begin Mileage as it is before the change
      5. Offset\_to End Mileage as it is before the change
      6. Source\_table Some fake Agile table name
      7. ---
      8. Now change the data either using forms or spatial manager so that there is an DECREASE of length between the mile points you gave.
4. Route Location Closed
   1. This item should be reported if the route was closed since the historic date. If you know of a route where this occurred then use that else, fill in the XAA\_LOC\_IDENT Table with a route and then close the route to see if it well show up when ran.

Now that some data is inserted, it’s time to run it and check the output.

In toad or sql devolper

Type in:

exec xky\_hig\_to\_aa.process\_route\_events;

And execute it. Once this is done, you should be able to run the query:

Select \* from XAA\_NET\_REF;

Looking in the following fields and see if you have the output you were expecting. If a route location changed and was spilt via distance break, you may see more than one entry for the item.

|  |  |
| --- | --- |
| NEW\_DATE | Date of the locations coming from the Exor System  (filled by Exor) |
| NEW\_ROUTE\_NAME | New Route Name (filled by EXOR)  If the route is closed or invalid this value can be null |
| NEW\_OFFSET\_FROM | New From Milepoint (filled by EXOR)  If the route is closed or invalid this value can be null |
| NEW\_OFFSET\_TO | New To Milepoint (filled by EXOR)  If the route is closed or invalid this value can be null |
| PROCESS\_MSG | Messages and Errors produced during the process. Identified possible errors are:   * Old Route Location Not Found * Route Location Not Changed * Route Location Changed * Route Location Closed |

Now that you verified that date it would be good to randomly check some routes that you know have been lengthened or shortened by using the same method.

3.3 Asset Information

This item works by filling in 2 tables, run the procedure and then getting a new table with the desired asset information in it.

There would some asset information filled in these tables that were listed in appendix A of the design document. I would start by verifying that there is data in xaa\_asset\_attrib for:

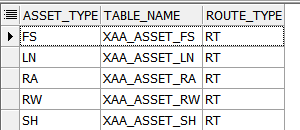
AL, FS, LN, RA, RW, SH, SL, SS, TF

If they exist then check for the items xaa\_asset\_type I believe there should be 29 rows in that table.

Then run:

exec xky\_hig\_to\_aa.generate\_asset\_info;

Now check the table defined in xaa\_asset\_attrib Notice that one of the fields is labeled Table name that is your output table For example querying back xaa\_asset\_attrib in toad gives:

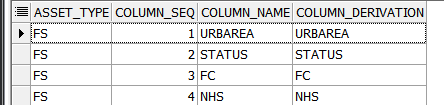


We can see that the output table was xaa\_asset\_fs for the FS asset type entry.

If I query:

select \* from XAA\_ASSET\_ATTRIB where asset\_type = 'FS'

You can see what extra columns you should have in the table xaa\_asset\_fs



Once you verified the extra columns, go ahead and run

Select \* from xaa\_asset\_fs order by route, from\_point;

Browse the list or pull up a particular route with a where clause to check the results.

You should be able to spot check 1 or 2 more for the initial testing; the code is the same so if you find something broken I would expect to find it broken for all.

I created the output table from my understanding of the needs that were listed in the design document, if it is in the wrong format please let me know.

3.4 reports

The report should list all Length change operations since the report function was last ran. Now for initial testing purposes I have included a parameter that allows us to manually assign a last ran date. Once the procedure is ran the results are stored in the table: XAA\_LENGTH\_CHANGE

These changes should be:

On a datum:

* Close
* Newly created
* Reclassify
* Rescale
  + Only if a new segment get made or closed, this includes distance breaks

Using forms or Spatial Manager do an operation that would make each of those things happen. I would like you to use your normal workflows so I can’t say exact the best way to do that is. I want to see if your workflows might uncovers something I didn’t take into account.

Once some changes are made we can run the process to update the report:

Normally you would just run: exec xky\_hig\_to\_aa. generate\_report\_info;

But since we are testing we are going to us an option parameter to tell the procedure that the last run date was yesterday for example.

exec xky\_hig\_to\_aa.generate\_report\_info(to\_date('04-MAR-2014'));

Once this is ran we can access that output form the last run but running the following query:

select \* from xaa\_length\_change

where report\_run\_date= (select max(report\_run\_date) from xaa\_length\_change);

The where clause will only bring back the results form the last run.



One thing that is missing from this sample are the spatial changes, they well show up in yours; the last thing that I had tested was the install and that cleared out my tables.

If you have any questions contact me or Jeremy and one of us can set up a meeting to help you.