Introduction

There are numerous tests applied to your submission. The testing starts with the files submitted to the system in a zip file. These generic tests will usually result in the submitting person receiving an email indicating the nature of the error(s). Some error conditions result in the system stopping, such as file naming errors and further testing cannot continue until a submission is received which resolves the error. Preliminary error tests include:

1. The structure of the control file is incorrect, this happens only on a batch submission. As it may not be possible to determine who submitted the file only the system administrator is notified (stops process, notify administrator).
2. There are folders in the zip file (stops process, notify submitter, status to error). The notification will state that the structure of the zip file is incorrect. The zip file contains folders and the shapefiles need to be placed in the root of the zip file with no folders present.
3. The files are named correctly (stops process, notify submitter, status to error). The notification will list the file or files that are not named according to the specification. If version 2 naming is used the process continues but the files will need to be converted at a later point in the process. A parameter is passed to the future steps to note that the files are version 2. If the files are being submitted as version 2, they must be submitted using the road_name field as the master road name field. If this is not the case the process stops, notify the submitter that the files must be submitted using the road_name values. The road_name values must be populated for road classes that require road names.
4. All the required files are included in the submission. A minimum of 12 are required (4 per shapefile including shp, shx, dbf and prj) (stops process, notify submitter, status to error). The notification will list the missing files which are needed to do a proper submission. This process will need to account for version 2 filenames.
5. The files are all in the same map projection which is a valid projection (stops process, notify submitter, status to error). The notification will let the user know that all the submitted files must be in the same projection. If one or more of the files are in an invalid map projection the notice will list the files with invalid projections and list the valid projections that can be used. This process will have to account for version 2 filenames.
6. The mandatory fields are missing in one or more shapefiles (stops process, notify submitter, status to error). The notice shall include the details of which fields are missing from which files. This process will have to check for different fields if the files are version 2.
7. One or more of the fields submitted has the wrong data type or field length (stops process, notify submitter, status to error). The notice will include the details of the fields that do not conform to the specification and the files that they are in. This process will have to check for different fields if the files are version 2.
8. One or more of the shapefiles are in a multipoint or multilinestring format (stops process, notify submitter, status to error). The notice will let the user know which files are structured in a multipoint or multilinestring format. The files must be created in simple point or linestring format. This process will have to account for version 2 filenames.

The user should note that each of the above tests will immediately halt the further processing of the data. If multiple errors of these types exist within a submission it will be a painful process for the user as

they correct one error, resubmit, and find they have a second error. It is wise therefore to thoroughly check the submitted data to ensure that it meets the requirements of all these preliminary tests.

If submitting data using the automated submission process these errors should be returned to you in an email. In the event of a serious error in the submission the review grid will not indicate a successful or failed submission. Contact the administrator. It is recommended that your during the development of your automated process that you manually submit the files you intend for automated submission to detect if you have any of these preliminary errors in your automated process.

The following sections in the document list of all of the tests applied to your data during processing. Each test, particularly attribute tests, may apply to hundreds of values and fields but the test is considered to be one test. For example, the test that checks the attribute field names will check all the attribute names in all the files but this is considered a single test.

Addition of all v3 fields.

Unlike the submission format in version 2, the new submission format does not require the inclusion of all fields. Only mandatory fields for each dataset must be included. The remainder of the fields will be added by a workspace in the submission process, and some will be populated with standard values based on the geoadministrative area that the data falls in. The files returned to the user will include a complete set of fields. The values in the street name parsed fields will be populated using the street name dictionary. Other fields to be populated are noted in the schema table for each dataset.

Submitting version 2 files.

The system has a routine that runs after the preliminary tests to detect if the submitted data is in the old version 2 format. If it is the system will attempt to create a set of version 3 formatted files and submit those to the system for processing. Version 3 files will be returned to the user. The conversion is likely to create many errors that need to be reviewed and repaired in the version 3 file format. There are attributes that are not available in the version 2 format and thus further editing of the version 2 data will never fix the issues. This is simply a means of making the conversion easier for the user.

The conversion process is the first step after the preliminary tests. If the preliminary tests all pass the data is in a consistent enough format to do the conversion. The field names in the documentation below are in lower case but may be represented in the data in a different case. Following are the steps for the process:

1. The submitted files are checked for any user data fields and those are noted.
2. A new set of shapefiles or feature classes are created including any user data fields.
3. The data is copied to the new shapefiles or feature classes. Any features that contain attributes that do not conform to the rules applied will result in creating an error record in the appropriate error shapefile noting the attribute which contains data that cannot be converted. In this case the v3 attributes are to be left blank for that attribute on that feature. The mapping of certain fields uses the following logic:
    a. In the address file, if the house_id and access_num fields both contain data the access_num value goes into the add_number field and house_id goes into the building field. If the address number is alpha numeric and the number proceeds alpha

characters, the number goes in add_number and the suffix goes into addnum_suf. If the alpha is a prefix, generate a conversion error. If either the house_id or the access_num are blank or null use the populated value for the add_number dealing with any alpha characters that may be present. If both house_id and access_num are blank no address_num value is populated and this will be picked up and an error created by a subsequent test for missing attributes.

b. In both address and road files, if the road_name field contains abbreviations for directional prefixes or suffixes such as N, S, NW, NE etc. these are to be translated to North, South, Northwest, Northeast etc. The resulting string is placed in the a_st_name field. This will subsequently be checked against the street name dictionary and will either populate other field values or result in an error.
c. The address field a_add_type will be populated with Unknown. This will require the user to revisit the values in this field and populate it with something meaningful. Typically, the address points represent the access to the property, the building entrance or the centroid of the parcel.
d. The address field a_mast_add is to be populated with 1 for all address point features.
e. The address field a_security is to be populated with 0 which will make all address data available to the public. **The user is responsible to mark any address points that are to be removed from public distribution with a value of 1.**
f. If the civic_id field contains data is to be placed in the uninccom_l and uninccom_r fields and a conversion error created. The user will have to verify that the name represents an unincorporated community name. If the value is for a neighborhood the user should move the values to the nbrhdcom_l and nbrhdcom_r fields. If the values are for an incorporated municipality, they should be removed from the uninccom_l and uninccom_r fields as the incorporated municipality name will be automatically written to the incmuni_l and incmuni_r fields by a subsequent process.
g. In all the datasets the value in the source field is to be included in the v3 file as a user defined field.
h. In the mplaces data the values for sub_cat, address, phone, zoom and t_scale are to be included in the v3 file as user defined fields.
i. In the road file the values for t_scale and e_speed are to be included in the v3 file as user defined fields.
j. In the road data the oneway value for any one-way roads is to be set to FT and Both for any two-way roads.
k. In the road data if all the values of the from and to address range fields are 0 the valid_l and valid_r fields are to be sent to N and the parity_l and parity_r values are set to Z. If road address ranges are included in the data, the user should review and appropriately populate the parity left and parity right fields indicating if the ranges represent odd or even values. **This is the responsibility of the user and there is no test to ensure this is done correctly**.
l. In the road data the values in the following fields translate directly into values in the v3 data: surf_type -> a_surftype, meridian -> a_mer, num_lane -> a_numlane, rd_width -> a_rd_width, road_sign -> a_rd_sign, owner -> a_owner, maintain -> a_maint, alias_1 ->

a_alias_1, alias_2 -> a_alias_2, overpass -> a_overpass, fr_left -> fromadd_l, to_left -> toaddr_l, fr_right -> fromaddr_r, to_right -> toaddr_r.

m. In the road data the value in the rd_desc field will be placed in the roadclass field. The value HWY 1-216 will be translated to PRIMARY HWY and the value HWY 500-986 will be translated to SECONDARY HWY. The topology tests and street name test for the road data will be based on these new values.

n. In the places data the name data will be placed in the landmkname field, the category data will be placed in the place_type field which will subsequently generate errors that have to be manually populated by the user. The placement field in the v3 data will be left blank which will subsequently generate errors as this is a mandatory field for NENA. The user should be aware that the new v3 format accepts the input of mileposts. The user should consider adding mile post points for any roads in their area that have them.

Structure of error shapefiles

Below is an example of an error shapefile. There will be an error file created for rd (roads) add (addresses) and lmark (landmarks). The error shapefile for any submitted file that does not have any errors will be empty.

error_rd_twn_xxxx

FID	Shape *	error_id	error_code	description	sub_date	record_id	col_name
0	Polyline	1	20	One way road direction error	202303301020	3723	geom
1	Polyline	2	20	One way road direction error	202303301020	3723	geom
2	Polyline	3	20	One way road direction error	202303301020	3732	geom
3	Polyline	4	20	One way road direction error	202303301020	3738	geom
4	Polyline	5	20	One way road direction error	202303301020	3738	geom
5	Polyline	6	20	One way road direction error	202303301020	3739	geom
6	Polyline	7	20	One way road direction error	202303301020	3746	geom
7	Polyline	8	20	One way road direction error	202303301020	13	geom
8	Polyline	9	20	One way road direction error	202303301020	19	geom
9	Polyline	10	20	One way road direction error	202303301020	44	geom
10	Polyline	11	20	One way road direction error	202303301020	26	geom
11	Polyline	12	20	One way road direction error	202303301020	28	geom
12	Polyline	13	20	One way road direction error	202303301020	31	geom
13	Polyline	14	20	One way road direction error	202303301020	177	geom
14	Polyline	15	20	One way road direction error	202303301020	51	geom

The structure will be the same for both the error and exception shapefiles. The only differences are that the road error and exception files are polyline shapefiles, and the others are point shapefiles.

The col_name (column name) field in the error attributes is used to indicate missing columns in submitted data, column names that do not have the correct data type, the attribute name of attributes that do not match a lookup value, and attribute columns that are missing that are required.

Tie point snapping and vertex creation

This is the only routine in the validation process which modifies the geometry of submitted data. The submitted data is converted to 10TM in the testing workspace and then the end points of the roads

within 10cm of a tie point are snapped to the tie point. This forces the coordinates of the end of the line to be the same as the tie point. The coordinates of all tie points are rounded to 2 decimal places in the 10tm projection. The coordinates of these end points in the projection that the data is submitted in will likely contain more decimals due to the projection conversion. The tie points and the geoadmin boundaries share vertices at the tie points. This keeps everything topologically in sync.

Another part of this routine is to snap the minor roads to the major roads within 0.01 meters and introduce a vertex in the major road. This forces a clean topological relationship into the road network. If the user has concern about their road network being modified note that loading the submission data back into their system.

A check will be performed at the end of this process to find any features with invalid geometry. These features are to be inserted into the road error file with code 210.

Tests applied to all data

Multipart geometry (code 101) – Any features that are detected as multipart features will be flagged in error. All multipart features will have to be dropped to be single part features.

Change in size of data over limit (code 102) – This is a warning to the user that a shapefile has increased or decreased more than 20%. It could indicate an inadvertent deletion of data or some other error. This will require the contractor configuring the database to have a storage size recorded for each submission file. The test will also require the creation of a fictional feature to attach the attribute to which will either be a point or line inside the geoadministrative boundary. The line should be 6 meters long pointing due East.

The user does not need to do anything to resolve this issue but to resubmit the data unless the dramatic reduction or increase in data size resulted from a blunder on the part of the user. **The next submission will automatically eliminate this error as, at that point, the file will not have significantly changed in size**.

Attribute missing (code 103) – Any attribute that is marked for testing in the schema that must contain a value that is not governed by a lookup table and is null or empty will be flagged. The field that requires a value will be indicated in the column name field.

Outside municipal boundary (code 104) – The data submitted must be within the geographic boundary of a distinct geoadministrative area. Road linework must stop at the boundary at a tie point. These polygons are defined by the province of Alberta and maintained by Altalis. There are slight modifications made to the geoadministrative boundaries for the AMDSP system. A current copy of the boundary file can be downloaded from the AMOS web site. At times the data may not be in sync with the boundaries used by the member. Should a member change their boundary please contact AMDSP support to get the information updated.

A particular member may be able to submit data for one or more geographic areas, but each must be a separate submission. For example, Frog Lake first nation can submit data for the PUSKIAKIWENIN INDIAN RESERVE #122 and UNIPOUHEOS INDIAN RESERVE #121. The two areas belong to the same first nation but are independent areas and need to be submitted separately. Data which is outside the geoadministrative area or extends beyond the boundary will be flagged as an error and must be

corrected. If the data is relevant to the member but is in another jurisdiction the member should request to have the data included in the neighboring geoadministrative area.

Lookup value validation failed (code 105) – The value in a field that is tested using a lookup table is in error. The field containing the data that is in error is indicated in the column name field in the error shapefile.

Non-printable characters (Code 106) – This test will look for non-printable ASCII or non utf-8 characters within each shapefile. If these characters are detected an error will be placed in the error file for that shapefile. The error will record the fieldname that the invalid character is in.

Excessive use of value UNKNOWN (code 107) – The system will scan through the attributes and check for the use of the word UNKNOWN in any of the text fields. There are several attributes where this value can be used but to fill in a value without determining what the attribute should be set to is discouraged. For example: the road surface type should be set to one of the common surface types (gravel, dirt, concrete, asphalt etc.) setting all or most of the roads to be UNKNOWN is not acceptable. The error will indicate the field that has exceeded the quota which is 2% of the records in the dataset.

Numbered street using alpha street name (Code 108) – Street names that are numbered need to be input with the numeric value of the street not the word for the number. For example: First Street and Second Avenue need to be input as 1 Street and 2 Avenue. This error will also result in street name dictionary issue being flagged in the error file pertaining to the submitted feature. This error will be generated by finding the word First, Second, Third, Fourth, Fifth, Sixth, Seventh, Eighth … up to Twentieth in any part of the street name.

Street name not in dictionary (Code 109) – The street name submitted for a street name or alias name is not listed in the street name dictionary for the member. The street name must be added to the dictionary. Street names will be reviewed by the system administrator and removed if they are invalid. The test should include the field name that contains the incorrect value.

If the street name is a simple, seemingly correct value, the system should add a record to the street name dictionary with the approval fields left blank. Simple street names would include such names that have a single alpha name part and a standard street type, number and the word street or avenue, standard highway, township road and range road types. The portions of the street name will be placed in the appropriate fields in the new street name record. If the name is more complicated no street name value will be created and the user will have to manually create the entry.

V2 to v3 conversion error (Code 110) – The automated routine that tries to convert the format of the submission data from v2 to v3 will create errors that need to be reviewed and fixed by the user. The conversion process creates situations where values are not available for the new format or old values are ambiguous and cannot be translated directly.

Tests applied to Roads

Street name missing (code 201) – There is no street name for road that should have a street name. Normally this indicates a street name that is null or blank. The lookup table for the Road Classification indicates which road classes require a street name. UNKNOWN or UNNAMED can be used where the street name value has not been determined but is to be used sparingly.

Orphan road segment (code 202) – All road segments within the road data must connect to each other or to a tie point.

End Point Connection (Code 203) – This is one of the most difficult errors to understand. The system will flag one or more features that are not topologically correctly connected to each other. In most cases this is due to one of the features having been given the wrong road classification or caused by the road segments not being broken so that they meet end to end. Some investigation is required to determine the cause of the issue as not all features involved in the problem may be flagged as being in error. The error may be caused by a feature that doesn't have an error.

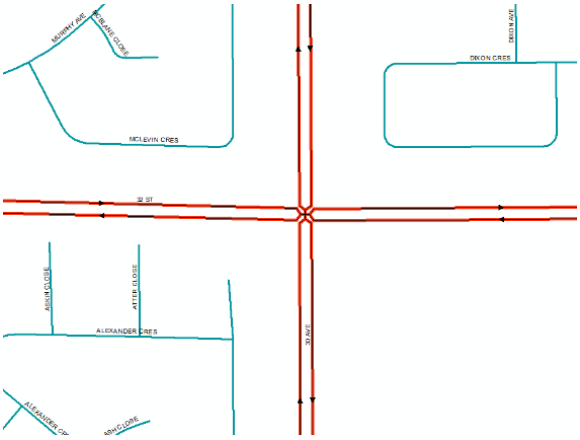
The road classification determines if the road segment is a major or minor road. All major roads must connect to all other major roads end to end. All minor roads must connect to all other minor roads end to end. A minor road may connect to a major road on a vertex of the major road. If no vertex is present and the roads mathematically touch within 0.01 meters the system should introduce a vertex unless the road segment is an overpass structure. Any roads that do not connect properly to other road segments according to these rules are to be flagged with this error.

Overpass error (Code 204) – This test looks for roads that are flagged as overpass segments and then looks at any segments that pass over or under the feature. If there are vertexes in either the overpass segment and or crossing lines that touch the overpass or crossing line, the overpass feature will be flagged in error.

Road segment duplicates (code 205) – If a road segment is duplicated (there are two exact copies of the same road segment) or if a portion of one segment is coincident with another segment the features will be flagged with this error and may also receive other error types such as end point errors.

Road shorter than 2 meters (Code 206) – The indicates road segments that are shorter than 2 meters in length. In many cases these are segments near the edge of the municipality where a short segment is required to connect from the intersection in the member data to the tie point on the edge of the member area. In most cases it is advisable to modify the geometry to "exaggerate" the short segment so that it is slightly longer than 2 meters. In cases where these segments are valid, and modification is not possible, the segment can be left as is but an exception record will have to be created in the road exception shapefile.

One-way road direction error (code 207) – The one-way road test looks for road segments that are flagged as one-way roads that have the same street name that meet head-to-head. Essentially this indicates where vehicles travelling in the direction of travel indicated in the mapping would run into each other head on. There are cases where this is valid. One example is where the road network has been built so that intersections meet at a point. These errors will have to be included in the exception file so that they are ignored.



Dangle errors (Code 208) – Roads that have an end point within 3 meters of another road will be flagged with a dangle error.

Missing tie point (Code 209) – This error indicates where a road touches the geoadministrative boundary for the member and does not have a coincident tie point on the boundary. This often will reflect a situation where the road has not been snapped to an existing tie point.

Invalid geometry (Code 210) – This error indicates road segments that cannot be snapped to a tie point where doing so would create an invalid line feature such as a zero length line or a line that crosses itself. See snapping workspace near the beginning of the process. The routine also looks for other line features that have invalid geometry.

Tie point not connected to any road (Code 211) – This error indicates where a tie point exists for the member but there is no road snapped to it. The system will generate a 6 meter long line that points due East from the tie point. The purpose of the line segment is to give a feature in the error shapefile that can store the attribute data for the error. The line segment has no purpose other than to be a geographic place marker of the issue.

Duplicate address in address ranges (Code 212) – This indicates that the same address can be found on multiple road segments. This excludes the value 0. It could be an indication that a segment has been split and the address range information is duplicated or some other issue. All affected road segments will be flagged. This test can be made an exception. **The test is optional.** If you choose not to run the test the exceptions are not required.

Tests applied to Addresses

Address point outside acceptable distance (Code 301) – Address point 100 meters beyond any road or driveway that would take it to a road. . **This is an optional test on the submission dialog.** This is a relatively simple test which checks if there is a road or driveway within 100 meters of the address point. It does not compare the road names. That check is done by the street name mismatch test.

Duplicate addresses (Code 302) – Duplicate addresses are address points that contain exactly the same address field values including the AMDSP Master Address field. Fields compared include A_St_Name, A_Mast_Add, Uninc_Comm, Nbrhd_Comm, AddNum_pre, Add_Number, AddNum_Suf, Building, and Unit. Multiple copies of an address may exist in the data but only one can be flagged as the AMDSP Master Address in the A_Mast_Add field.

Street name mismatch (Code 303) – The street name for the address point must match a street name of a road within 100 meters or the road that the driveway nearest the address point leads to. Exceptions are allowed. This test requires that the workspace performing the test do the following steps:

- Starting at the address point find any roads or driveways within 100 meters. If there are no driveways check if the street name of the point matches the street name of one of the roads found.
- If a driveway Is found and is closer to the address point than the nearest road, follow the driveway down all of the paths that the contiguous driveway network leads to until each either results in a dead end or touches a road.
- Check all the street names of the roads that the driveways lead to and see if one matches.

- If none of these tests produce a match create the error record for the address point.

Street name missing in roads layer (Code 304) – The street name of an address point must have a matching street name in the road data. Exceptions are allowed. It is possible that the address refers to a road in another jurisdiction.

Address near hamlet (Code 305) – Indicates an address point which is near a registered hamlet location or subdivision and contains address values consistent with an urban type of address. The unincorporated community field should be filled in with the hamlet name. **This is an optional test.** Exception records are allowed. The test uses the hamlet point feature class and checks that address points with 1000 meters of the hamlet point that have an street name with a value in the St_PosTyp field has the value of the hamlet name in the Uninc_Comm field.

Excessive use of Unit field (Code 306) – This error indicates that too many address points make use of the unit field which may indicate that the unit field is being used to store building numbers or address suffix values. Exceptions are allowed. The test will check for the percentage of addresses that have unit values above 5%. If the percentage is exceeded all the addresses will be flagged with this error.

Invalid characters in attributes (code 307) – The characters which are not numbers and letters are going to be flagged in error that are found in the Unit and Building fields in the address data. Characters allowed in street names are controlled by the entries in the street name dictionary. The field containing the data that is in error is indicated in the column name field in the error shapefile. An exception can be used but the exceptions will be scrutinized by the system administrator. If the user is using the dash character to represent a range of values (for example 1 - 10 in the unit field representing units 1 to 10) the user should replace this by creating 10 address points, each with a distinct unit number.

Tests applied to Landmarks

There are no particular tests that pertain to landmarks. The domain values are checked in the attribute testing routines. The name of the landmark must be provided as well as the classification of the landmark type using the Place Type field and lookup table. The address and street name would be useful but are not mandatory as some place names refer to physical locations.

Creation of NENA data

A workspace will run at the end of the completion of all the testing if there are no errors found and no corrections or approvals are required in the street name dictionary. The workspace will take the successful submission data and translate and load the information into an empty NENA geodatabase and then zip the file up. This zip file will be available for download by the user through a link on the review grid but will not be available to the public. Only the most recent successful submission will have the link.

Global tests for system

These tests are to be run each day for system administrators and are not triggered by or reported directly to members submitting data. The tests are included in this document just to make the users aware of the testing that goes on in the background. If errors are found by these tests and require data changes the applicable members will be notified.

Globally orphaned roads – This test checks for roads which are connected to a tie point and have a road in an adjoining member area, but the roads do not ultimately connect to the rest of the road network. The topology of the entire road network needs to be examined.

Tie points are all in pairs – All tie points in the system need to be in pairs. A single tie point would indicate an error in the tie point data.

Tie points missing a geoadmin vertex – Each tie point must coincide with a vertex along the boundary of the geoadmin polygon it pertains to.

Tie point coordinate has more than two decimal places – All the tie point coordinate data must be rounded to the nearest two decimal places. The coordinates may only be to two decimal places in 10TM which is the projection that the system uses. If the tie points are reprojected, the coordinates will have more decimal places. If possible, the contractor setting up the feature class should limit the number of decimals in the feature class to two decimals of precision. If this is done the test will become redundant.

Geoadmin vertex has more than two decimal places – All coordinates of geoadmin boundaries vertices must be rounded to the nearest two decimal places. The coordinates may only be to two decimal places in 10TM which is the projection that the system uses. If the boundaries are reprojected the vertex coordinates will have more decimal place values. If possible, the contractor setting up the feature class should limit the number of decimals in the feature class to two decimals of precision. If this is done the test will become redundant.

The distributed data must snap to all the tie points – All of the distributed road data will be checked if it connects to the tie points. Tie points each have an associated geoadmin area and must have a connecting road ending from the same geoadministrative road dataset. There may be new tie points affecting municipalities that are in a failed state. The id numbers of orphan tie points that are unconnected are to be reported to the administrator.

Error code list

A list of error codes is included in the Appendix B spreadsheet.