

February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 1 of 11

The Fatal Accident application has two objectives:

1. Develop a spatial database for current Fatal Accident Analysis and maps.
2. Develop a report to assist NDOR in maintaining the Nebraska component of the Federal Analysis Reporting System (FARS).

The Business process to accomplish these objectives is as follows:

1. Accident reports are received from involved drivers and law enforcement officials.
2. Accident reports are scanned into the NDOR Accident Report System (ARS).
3. Accidents are "location marked" and the accident data from the report entered into the NDOR Highway Safety Information (HSI) system.
4. An accident analyst specialist reviews fatal accidents. Data and reports are extracted from HSI and ARS respectfully for Quality Control (QC).
5. Upon completion of a month's data, the information is transferred to the Safety GIS Unit.
6. The Safety GIS Unit, extracts the related accident data from HSI, geographically locates the fatal accidents and creates a continuing, spatial environment for fatal accident analysis.
7. A report is generated, including latitude and longitude values, to assist Safety staff in maintaining the Nebraska component of the Federal Analysis Reporting System (FARS).

February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 2 of 11

NAD 83 OFF Accident Conversion:

1. Extract fatal accidents from NAD 27 StateYRoff.dgn
  - a. Create ULF based on severity code = 1
  - b. Use RIS Interactive to query severity\_code 1 count
  - c. The two counts above MUST match
  - d. Create DGN file (fatalYRoff.dgn)
2. Project fatalYRoff to NAD 83 fatalYRoff.d83
3. Blank attributes *longitude* and *latitude* to NULL
  - a. Use RIS Interactive
  - b. Update accidents\_YR\_off Set longitude = null, latitude = null where severity\_code = '1'
4. Re-compute *longitude* and *latitude*
  - a. Enter fatalYRoff.d83
  - b. Create ULF of fatal accidents
  - c. Load longitude and latitude values using *point\_loader*
5. Insert *accidents\_YR\_off* into *accidents\_fatal*
  - a. Use RIS Interactive
  - b. Insert into *accidents\_fatal* (*mslink*, *longitude*, *latitude*, *accident\_key*) select 0, *longitude*, *latitude*, *accident\_key* from *accidents\_YR\_off* where severity\_code = '1'
  - c. Check *fatal\_accidents* - Select count(\*) from *accidents\_fatal* where *mslink* = 0
  - d. If totals do no match, correct prior errors prior to Step e.
  - e. When correct run *mslnkldr* (MGE Prompt: *mslnkldr - v - t accidents\_fatal*)
6. Populate fatal accident attributes
  - a. Extract HIS data using fatal update query
  - b. Execute *up\_fatal.exe* to populate fatal accident attributes
7. Populate coordinate values
  - a. Longitude/Latitude populated
  - b. Populate *longitude\_dms* and *latitude\_dms* using *point\_loader*
  - c. Populate *stateplane\_x* and *stateplane\_y* using *point\_loader*

February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 3 of 11

The status of the above tasks is as follows:

1. Develop Procedures: The procedures are generally known from past experience. The detail procedures will be developed and documented after the known tasks are developed and tested.
2. Locate ON Fatal Accidents: A C++ routine has been developed to populate the fatal accident database table that will use Segmentation to locate the accidents. The C++ routine is similar to that presently used in populating the GIS Accident database tables with ON accidents.
3. Spot OFF Fatal Accidents: As fatal accidents become available, they are being spotted using MGE digitizing tools in a state DGN file. The spotted accidents are identified by the *accident key* attribute. This procedure is identical to that used to spot county OFF accidents.
4. Update OFF Fatal Accidents: A C++ routine has been developed to update the OFF accident attributes in the fatal accident database table. The *accident key* attribute

NEBRASKA DEPARTMENT OF ROADS  
 Fatal Accident Application

Page 4 of 11

is the link between the spotted accident and the update record. The C++ routine is similar to that presently used in updating the GIS Accident database tables with OFF accident attributes.

5. Update ALL Fatal Accidents: A C++ routine will be developed to maintain the data integrity of the fatal accident table.
6. FARS Report: Items needed by Jan were identified and included in version 1 of the fatal accident database table. A report will be generated for Jan that shows the data to enter into the FED system. This report includes the latitude and longitude values.

Procedure Steps:

- A. Request HSI accident updates that have occurred since the date of the last QC update. This data set may include both OFF and ON accidents for any year beginning in 1996.
- B. Perform the Stage 1 QC process by executing *qcoffon.exe*. The Stage 1 process identifies in the GIS accident database, HSI accidents that have been updated. This first process provides the tools to rectify differences in the existence of accidents in HSI and the GIS accident database. This is accomplished by updating the *status\_2* and *status\_3* attributes in the GIS accident tables for accident key matches. The *status\_2* and *status\_3* attributes are updated with an ascending sequence number. The *status\_2* and *status\_3* attribute updates are defined as follows:

Accident Key	* HSI Criteria *		GIS Action			
	* Accident Type	* Accident Type	* Accident Table	Status_2	Status_3	
196011902	OFF	OFF	OFF	-	100000x	
			ON	50000x	-	
197033177	ON	ON	OFF	100000x+1	-	-
			ON	-	50000x+1	
198019061	OFF	OFF	OFF	-	-	
			ON	-	-	
200018052	ON	ON	OFF	-	-	
			ON	-	-	

February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 5 of 11

The tools provided for the Stage 1 QC are a summary of HSI accident processing information, displayed on the screen prompt and a RIS batch file to update the GIS accidents.

- C. Establish a current GIS accident database by identifying those HSI accidents that are not in the GIS accident database and/or located in the incorrect GIS table.
  1. Initiate Stage 1 QC Document (qcstage1.doc)
    - a. Verify data files
    - b. Document the accident and record counts.
    - c. Document the GIS accident tables that are to be updated
  2. Determine the reliability of values in the *status\_2* and *status\_3* attributes. If values above 500000 remain from old QC testing, the QC process will be more difficult.
  3. Update the GIS accidents
  4. Analyze the Stage 1 QC Document

Procedure Step Descriptions:

- A. HSI Request:
  1. Request Sam Prieb to extract the data set described by the query *Updatecadd\_Q*.
  2. The data should eventually exist in the directory *C:\users\safety\data* as an \*.dat extension.
  3. Example is *C:\users\safety\data\hsiupdate.dat*. EXHIBIT 1 defines the extracted file as *update.dat*.
- B. Perform Stage 1 QC
  1. Execute *qcoffon.exe*, Option *qckey*:
    - a. Execute from operating directory *C:\users\safety*.
    - b. Execute *qcoffon.exe OD hsidatafile gisdatafile Option* where the parameter definitions are:  
*OD* is operating directory  
*hsidatafile* is *hsiupdate.dat* (Enter without extension)  
*gisdatafile* is *hsiupdate.dat* (Enter without extension)  
*Option* is *qckey*  
Example of the executable statement is:  
*qcoffon.exe C:\users\safety hsiupdate hsiupdate qckey*
    - c. Special Note - The option *qckey* does not use the *gisdatafile* but to maintain conformity and simplicity the parameter is included. The requirement for the third parameter is that it exists in the directory *C:\users\safety\data\*. The example above uses the same data file for the second and third parameters as a convenience that meets parameter requirements. Other options used in subsequent process tasks require a third parameter data file.

February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 6 of 11

- d. The executable statement for EXHIBIT 1 is *qcoffon.exe*  
*c:\users\safety update update qckey.*
2. The Stage 1 tools generated are:
  - a. A file located in directory *C:\users\safety\bulk* with an *\*.ris* extension. This is a RIS batch update file.
  - b. Prompt screen output
- C. Establish current GIS accident database.
  1. Initiate Stage 1 QC Document (*qcstage1.doc*)
    - a. The initial QC information is to be extracted from the prompt screen.
    - b. The Stage 1 QC Document can not be completed until the RIS Batch update is processed.
    - c. See attached example EXHIBIT 1
  2. Establish *status\_2* and *status\_3* value reliability.
    - a. Use RIS Interactive to verify *status\_2* and *status\_3* value reliability. The Stage 1 QC document identifies all tables that need to be checked.
    - b. If the *status\_2* and *status\_3* attributes are populated with values greater than 500000, rectifying differences will be more difficult. In this situation it is probably best to update the *status\_2* and *status\_3* attributes with the "null" value.
    - c. The best SQL methodology is to update each *status* attribute for each table separately. An example is as follows:

```
Update accidents_98 Set status_2 = null where status_2
>= 500000
Update accidents_98 Set status_3 = null where status_3
>= 500000
```
  3. Update the GIS accidents to identify HSI matches.
    - a. Update the GIS accidents by executing the RIS Batch file from the operating directory, *C:\users\safety*. An example of the executable statement is:

```
risbatch < C:\users\safety\bulk\hsiupdate.ris
```

Correspondingly the EXHIBIT 1 statement is:

```
risbatch < C:\users\safety\bulk\update.ris
```
    - b. Complete the Stage 1 QC Document.
      - (1) The end of the *risbatch* process displays on the screen a group of text that appears as follows:

```
ris>
expression: xxx
End of Data Found ...
```

February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 7 of 11

- (2) This group of text exists for each accident table identified in the Stage 1 QC document. The accident tables are in the same order displayed when *qcoffon* was executed. The "expression" indicates the number of accidents updated in that table and is entered in the "RIS Batch Count" column of the Stage 1 QC document. See EXHIBIT 1.

4. Analyze the Stage 1 QC Document

- a. Analysis 1 of the Stage 1 QC document is to sum the total number of OFF accidents and total number of ON accidents for all the Accident tables in the Stage 1 QC document. Compare these totals with the Accident Count items 3.c and 3.d in the same document. If both totals match, then it is assumed that there are no accidents missing and the next process is to update the GIS database. Proceed to the XXXXX process. See attached example EXHIBIT 2.
- b. If one or both of the totals do not match, the next step is to determine if a HSI OFF accident is a GIS ON accident or visa versa. Analysis 2 is accomplished by evaluating the *status\_2* attribute using RIS Interactive. The following demonstrates how this is accomplished for EXHIBIT 2.

- (1) EXHIBIT 2 shows that the ON accident total matches, thus there are no ON accident discrepancies. However, EXHIBIT 2 shows the OFF accident totals differ by 2.

- (2) To assess an OFF/ON reverse situation, it is necessary to analyze the *opposite* accident type. To resolve the OFF accident discrepancies in EXHIBIT 2, RIS Interactive is used to analyze all ON accident tables that are a part of the Stage 1 QC document. The SQL statement for EXHIBIT 2 is as follows:

```
Select accident_key, accident_date,  
       accident_seq_no, status_2 from accidents_98 where  
       status_2 >= 500000 order by status_2  
Select accident_key, accident_date,  
       accident_seq_no, status_2 from accidents_99 where  
       status_2 >= 500000 order by status_2
```

- (3) The result for the EXHIBIT 2 is that there are no ON GIS accidents that matched HSI OFF accident keys for years 98 or 99. EXHIBIT 2 will require Analysis 3.

- (4) If a discrepancy is found, the SQL statement would yield sufficient data to research the accident in error and HSI or the GIS database corrected accordingly.

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 8 of 11

- c. Analysis 3 is to determine the accident key of the HSI accidents that did not find a GIS match. The methodology used is to analyze the *status\_3* attribute of all the GIS database updates to determine the missing sequence numbers. Since the updated GIS accidents populate several tables, the following steps are used in Analysis 3.
- (1) Use RIS Interactive.
  - (2) Delete the records from the table *accident\_sort*. The SQL statement is *Delete from accident\_sort*.
  - (3) Open the query *c:\users\safety\query\accsort.qry*.
  - (4) Modify the *accident\_96* table referenced in the query as required and execute for each accident table in the Stage 1 QC document.
  - (5) After completing the "Insert Into" SQL statements for all of the Stage 1 QC document tables, execute the SQL statement, *Select count(\*) from accident\_sort*. Compare this count with *Total OFF/ON Accidents* item in the Stage 1 QC document. These two totals MUST match or there is a process procedure error.
  - (6) The sequence numbers in *status\_3* are of 2 different magnitudes depending on whether they are OFF or ON accidents. Sequence numbers greater than 1000000 are OFF accidents and less than 1000000 ON accidents. To rectify this magnitude difference, execute the following SQL statements:  

```
Update accident_sort Set status_1 = status_3
- 1000000 where status_3 >= 1000000
Update accident_sort Set status_1 = status_3
- 500000 where status_3 <= 1000000
```
  - (7) To locate the missing sequence numbers, generate a report from the *accident\_sort* table. The SQL to generate this report is:  

```
Select accident_key, accident_date, status_3,
status_1 from accident_sort order by status_1
```

Review the report and ascertain the missing sequence numbers from the *status\_1* column.
  - (8) Using the missing sequence numbers, analyze the RIS Batch update file to identify the accident key associated with missing sequence number.
- d. The Analysis 3 as applied to the example defined by EXHIBIT 2 is as follows:
- (1) Beginning at Step 4.c.5, where the "Insert Into" SQL statements have been completed, the "Select Count(\*)" SQL statement from the *accident\_sort* table yields an



February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 9 of 11

expression of "422" records. This matches the *Total OFF/ON Accidents* in the Stage 1 QC document and confirms there are 2 missing accidents.

- (2) After populating the *status\_1* attribute as defined in Step 4.c.6, Step 3.4.7 generates a report from the *accident\_sort* table. The pertinent part of the report is shown below:

accident_key	accident_date	status_3	status_1
198040089	1998-05-22	500001	1
198057682	1998-08-20	500002	2
*	*	*	*
*	*	*	*
*	*	*	*
200013152	1999-11-26	500019	19
198045973	1998-06-20	500020	20
199051522	1999-07-14	500023	23
200004859	1999-10-11	500024	24

The 2 missing sequence numbers in the *status\_1* field are 21 and 22.

- (3) Armed with the missing sequence numbers, the accident key can be located from analyzing the RIS Batch update file, *update.ris*. Again, the pertinent part of the update file is shown below:

```
UPDATE accidents_99 SET status_3 = 500000 + 19 where
accident_key = 200013152;
UPDATE accidents_99_off SET status_2 = 1000000 + 19
where accident_key = 200013152;
UPDATE accidents_98 SET status_3 = 500000 + 20 where
accident_key = 198045973;
UPDATE accidents_98_off SET status_2 = 1000000 + 20
where accident_key = 198045973;
UPDATE accidents_99_off SET status_3 = 1000000 + 21
where accident_key = 199044450;
UPDATE accidents_99 SET status_2 = 500000 + 21 where
accident_key = 199044450;
```

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 10 of 11

```
UPDATE accidents_99_off SET status_3 = 1000000 + 22
where accident_key = 199057151;
UPDATE accidents_99 SET status_2 = 500000 + 22 where
accident_key = 199057151;
UPDATE accidents_99 SET status_3 = 500000 + 23 where
accident_key = 199051522;
UPDATE accidents_99_off SET status_2 = 1000000 + 23
where accident_key = 199051522;
UPDATE accidents_99 SET status_3 = 500000 + 24 where
accident_key = 200004859;
UPDATE accidents_99_off SET status_2 = 1000000 + 24
where accident_key = 200004859;
```

- (4) The missing accident information extracted from the batch update file is summarized as follows:
- | Sequence Number | Accident Key |
|-----------------|--------------|
| 21              | 199044450    |
| 22              | 199057151    |
- (5) The batch update file includes update statements for both OFF and ON accidents. However, the Stage 1 QC document has already identified the missing accidents as OFF accidents. Research of the source accident documents will identify the appropriate action to be taken.
- (6) Completion of Stage 1 is defined as follows:
- (a) A modified HSI accident input data file that reflects the accidents included in the GIS Accident database.
  - (b) A GIS Accident database that includes an accident record for every accident in the HSI accident input data set.

D. Stage 2 - Prepare GIS matching input data files.

1. Verify HIS accident input data file

- a. Edit the *hsidatafile*, *hsiupdate.dat*, to include only valid accidents. In our example this is *update.dat*.
- b. Clear values from *status\_2* and *status\_3* for all accidents included in the HSI input data file. An acceptable SQL methodology is as follows:

```
Update accidents_98 Set status_2 = null, status_3 =
null where status_3 >= 500000
Update accidents_98_off Set status_2 = null, status_3 =
null where status_3 >= 500000
```

- c. Execute *qcoffon.exe OD hsidatafile gisdatafile Option*

February 2, 2002

NEBRASKA DEPARTMENT OF ROADS  
Fatal Accident Application

Page 11 of 11

*Example of the executable statement is:*

*qcoffon.exe C:\users\safety update update qckey*

See Item B of this document for information defining the *qcoffon.exe* process.

- d. Identify Stage 2 QC totals. Use the screen output from executing *qcoffon.exe* to verify the accident totals.
  - e. Execute the \*.ris batch file (Example: *update.ris*) to update the *status\_3* attribute for sequencing.
2. Create the GIS matching input file
- a. Use RIS Interactive.
  - b. Delete the records from the table *accident\_sort*. The SQL statement is *Delete from accident\_sort*.
  - c. Open the query *c:\users\safety\query\accsort.qry*.
  - d. Modify the *accident\_96* table referenced in the query as required and execute for each accident table included in Stage 2.
  - e. After completing the "Insert Into" SQL statements for all of the Stage 2 accident tables, execute the SQL statement, *Select count(\*) from accident\_sort*. Compare this count with Stage 2 total count. These two totals MUST match or there is a process procedure error.
  - f. Update *status\_1* with *accident\_sort* sequence number. Recall that the sequence numbers in *status\_3* are of 2 different magnitudes depending on whether they are OFF or ON accidents. This is resolved by executing the following SQL statements:  

```
Update accident_sort Set status_1 = status_3 - 1000000
where status_3 >= 1000000
Update accident_sort Set status_1 = status_3 - 500000
where status_3 <= 1000000
```
3. Create the GIS matching report
- a. Load
4. Execute *qcoffon.exe* - option *qcacc*
- THE ACCIDENT\_SORT TABLE IS NOT CORRECT