SOFTWARE RELEASE NOTICE

01. SRN Number: PA-SRN-020	
02. Project Title: C14H, Gaseous Transport of Radionuclides, subroutine for TPA, CNWRA Version 1.1	Project No. 20-5702-723
03. SRN Title: C14H	
04. Originator/Requester: Thomas J. Ratchford	Date: 03/09/94

05. Summary of Actions

- ■ Release of new code admitted to CM System (R.Janetzke)
- ☐ Release of modified code:
  - ☐ Enhancements made
  - ☐ Corrections made
- ■ Change of access code (Robert Baca)

06. Persons Authorized Access

Name	RO/RW	A/C/D

07. Element Manager Approval: Date:

08. Remarks:

A copy of the software package C14H, CNWRA Ver. 1.1 was retained by the Principle Investigator for use in the CNWRA work center; therefore, a new release may not be necessary.

CNWRA Form TOP-6 (08/93)

SOFTWARE SUMMARY FORM

01.Summary Date: 03/09/94	02. Summary prepared by(Name and Phone) T.J. Ratchford 522-3083	03. Summary Action: New
04. Software Date: 8/15/93	05. Short Title: C14H	
06. Software Title: C14H - Gaseous Transport of Radionuclides.		07. Internal Software ID: NONE

08. Software Type:

- ☐ Automated Data System
- ☐ Computer Program
- ■ Subroutine/Module

09.Processing Mode:

- ☐ Interactive
- ☐ Batch
- ■ Combination

10. APPLICATION AREA

A. General:

- ☐ Scientific/Engineering
- ☐ Auxiliary Analyses
- ☐ Total System PA
- ■ Subsystem PA
- ☐ Other

b. Specific:

11. Submitting Organization and Address:

CNWRA, SwRI, San Antonio, Texas

12. Technical Contact(s) and Phone:

R. Janetzke, (210) 522-3318

13. Narrative:

The C14H code determines Gaseous Transport of Radionuclides at the Yucca Mountain site.

14. Computer Platform CRAY/XMP	15. Computer Operating System: UNIX	16. Programming Language(s): FORTRAN	17. Number of Source Program Statements: 3,446 lines of code
18. Computer Memory Requirements: UNKNOWN	19. Tape Drives: NONE	20. Disk/Drum Units: N/A	21. Graphics: UNKNOWN

22. Other Operational Requirements

NONE

23. Software Availability:
■ Available ☐ Limited ☐ In-House ONLY

24. Documentation Availability:
■ Available ☐ Inadequate ☐ In-House ONLY

25. Submission Package Status:

Acceptance Criteria: Met ■ Not Met ☐ Software QA Assessment: Successful ■ Unsuccessful ☐

Code Custodian: T.J. Ratchford

Date: 3/9/94

CNWRA Form TOP-4-1 (08/93)

C14H Cray Listing

```
gemstone.7 ~/tpa/C14H/VCS => ls
C14HTAR.TJR*      p.Makefile*       p.TPA_C14.CGD*    p.blkdat.F*       p.c14ds.F*
p.c14h.pre*       p.c14ha.F*        p.c14map.dat*     p.c14time.F*      p.c14trds.F*
p.datarep.F*      p.gasdev.F*       p.gwt.F*          p.itemp.F*        p.iter.F*
p.layer.F*        p.lhsoooo.out*    p.opnfil.F*       p.presid.F*       p.ran1.F*
p.rdgas.F*        p.read1.F*        p.readgl.F*       p.readlhs.F*      p.relout.F*
p.relrat.F*       p.seth.F*         p.sotc14.dat*     p.source.F*       p.ufun.F*
p.vapor.F*        p.vfun.F*         s.Makefile*       s.TPA_C14.CGD*    s.blkdat.F*
s.c14ds.F*        s.c14h.pre*       s.c14ha.F*        s.c14map.dat*     s.c14time.F*
s.c14trds.F*      s.datarep.F*      s.gasdev.F*       s.gwt.F*          s.itemp.F*
s.iter.F*         s.layer.F*        s.lhsoooo.out*    s.opnfil.F*       s.presid.F*
s.ran1.F*         s.rdgas.F*        s.read1.F*        s.readgl.F*       s.readlhs.F*
s.relout.F*       s.relrat.F*       s.seth.F*         s.sotc14.dat*     s.source.F*
s.ufun.F*         s.vapor.F*        s.vfun.F*
gemstone.8 ~/tpa/C14H/VCS =>
```

TJR 3/9/94

C14H Fortran Program
Static and Dynamic Analysis

June 28, 1993

Earl S. Marwil
John E. Tolli
Scientific Computing Unit
Idaho National Engineering Laboratory

1. Introduction

This analysis was performed on the Cray version of the software as provided by Southwest Research Institute (SwRI).

The C14H program contains 25 Fortran routines. Access to the source code was provided by SwRI on the INEL Cray. The code is normally passed through a preprocessor named prefor to select a version. In the c14h.pre file, there are apparently versions for VAX and Cray.

One sample problem was supplied along with the source code. The program was analyzed using the Craft (Cross Reference Analysis of Fortran) tool, FORWARN, the Fortran 77 analyzer, and PC-Metric. These tools provide static analysis, coverage analysis, and complexity analysis.

2. References

[1] N.H. Marshall and E.S. Marwil, Cross Reference Analysis of Fortran (CRAFT), EG&G-CATT-9198, EG&G Idaho, Inc., July 1991.
[2] Fortran 77 Analyzer User's Manual, National Bureau of Standards, NBS GCR 81-359, 1981
[3] FORWARN User's Guide, Quibus Enterprises, Inc., July 1991.
[4] PC-Metric User's Guide, SET Laboratories, Inc., 1987.

3. Functions

There were no alternate entry points in C14H. There was one unreferenced statement function in *c14ds*. This could be eliminated without affecting the functionality.

4. Common Block Irregularities

The common block declarations are consistent throughout the code.

There are a number of variables in common block *com1* which are not used. Some of these are defined but unused while other are neither defined nor used. These variables are *perm, gamma, c5, dhdt0, hvzero, amcon,* and *hzero.* No elements of *com2* are used in *c14time*; this common block declaration may be safely removed from that routine.

5. Interface Irregularities

There is a size mismatch of dummy arguments *nyear* and *gasin* in *rdgas* compared to those in the calling routine, *c14ha*. In *rdgas*, the dimensions are 10000, while in *c14ha* the dimensions are 300. The subroutine could simply use assumed-size array dimensions, such as *nyear*(*). This would eliminate the need for hard coded dimensions in the subroutine. In addition, the array size in the parent program could be declared using a parameter statement and the input array dimension tested against this limit.

There is a similar mismatch of arguments *nyear* and *gasin* in *source* compared to *c14htrds*.

6. Local Variable Irregularities

The parameter *ileft* is declared in routine *c14ds* but is unused. Local variables *c7*, *rhop4*, and *t4* are unused in *c14ds*.

In *c14ha*, variables *ivect, lhs, map, numdat, place*, and *sub* are unused. In *c14time*, previous usage of variables *sum* and *tavg* was apparently deactivated by changing statements to comments.

In *c14trds*, the character variable *title* is used but not defined. Since this occurs in a write statement, there is no effect on the computations. Some compilers will initialize character variables to blanks. This error may not be noticed on printed reports and output.

In *opnfil*, the variables *attrib* and *iostat* are not used and could be eliminated.

Several dummy arguments (*nkjend, nkjst, nklay, nskip, ntime*, and *times*) which are not used in *readgl* could be eliminated. This routine is called from *c14ha* where corresponding changes in the call list could be made as well.

7. Fortran Extensions

Fortran 77 requires that entity names be no longer than 6 characters. There are 39 instances of entity names which are 7 characters or longer. Fortran 77 requires use of only the uppercase alphabetic characters. There are 19 instances of entity names which use lowercase characters. These are extensions to the language which are recognized by most compilers. No changes need be made to these names.

A count is required preceding the x format specifier. This is omitted in the read statement:

```
read(map,'(a6,x,i5,x,i5)',end=199) varnam,place,sub
```

in *readlhs* at approximately line 33. This should be changed to standard FORTRAN as follows:

```
read(map,'(a6,1x,i5,1x,i5)',end=199) varnam,place,sub
```

8. Optimization

The following table summarizes the performance data gathered from execution of the sample problem. Only those routines exercised by the sample problem are shown (see "Coverage Analysis" for a list of routines not exercised by the sample problem, i.e., coverage = 0%). The table lists all program modules in descending order according to CPU time. To optimize code execution time, emphasis should be placed on those modules which appear highest in the listing.

In order to obtain meaningful statistics for performance evaluation, the program should execute for least 10 CPU seconds. The performance data show that the sample problem executed for a total of 35.849 CPU seconds.

The performance data show that a high percentage of the overall execution time (85.929%) is spent in the first 4 routines listed (ITER, GWT, RAN1, GASDEV). This is due primarily to the following (applies to some or all of the 4 routines):

1) a low percentage of floating point operations which are performed in vector mode (%Vflops is small)

2) a high overhead factor for calls to the routines (IFact > 1)

3) a high level of memory conflicts (MC/MR > 1)

4) a high rate of instruction buffer fetches (IBFR > 1).

A detailed optimization analysis effort should focus on these 4 areas.

PERFORMANCE DATA FOR C14H

ROUTINE NAME	Time	%ExTime	%AccumT	%Vflops	IFact	MC/MR	IBFR
ITER	18.609	51.910	51.910	67.93081	0.00	0.462	0.002
GWT	6.517	18.180	70.090	7.52978	0.01	0.396	1.191
RAN1	3.457	9.642	79.732	0.00000	1224.75	1.008	0.817
GASDEV	2.221	6.197	85.929	0.00000	368.43	0.510	1.132
ITEMP	1.581	4.410	90.338	57.15903	0.00	0.158	0.911
VFUN	1.184	3.302	93.640	0.00000	691.43	0.691	0.787
UFUN	1.101	3.071	96.711	0.00000	743.45	0.683	0.847
PRESID	0.987	2.754	99.465	65.65439	0.00	0.581	0.023
VAPOR	0.052	0.146	99.611	98.94281	0.00	0.119	0.154
C14TIME	0.035	0.098	99.710	0.07572	0.01	0.211	0.361
RELOUT	0.025	0.071	99.781	16.37933	0.00	0.303	1.033
C14DS	0.025	0.069	99.849	98.93375	0.00	0.470	0.197
READLHS	0.022	0.060	99.910	68.13826	0.00	0.134	0.758
RDGAS	0.012	0.033	99.943	87.74908	0.00	0.078	0.919
DATAREP	0.008	0.023	99.966	55.46730	0.00	0.174	0.958
C14TRDS	0.004	0.012	99.978	25.60517	0.00	0.339	0.704
READ1	0.002	0.005	99.982	85.88373	0.00	0.088	1.153
SETH	0.002	0.004	99.987	99.92641	0.00	0.191	0.195
C14HA	0.001	0.004	99.991	91.92902	0.00	0.395	0.683
SOURCE	0.001	0.003	99.994	94.92870	0.00	0.006	0.003
READGL	0.001	0.003	99.997	96.78000	0.00	0.261	0.780
OPNFIL	0.001	0.002	99.999	60.64642	0.00	0.367	0.817
LAYER	0.000	0.001	100.000	0.00000	0.00	0.007	0.011
Totals (All Traced Routines)	35.849	100.000	100.000	61.98849	637.13	0.470	0.462

Key:

- %AccumT = accumulated percentage of total CPU time
- %ExTime = percentage of total CPU time
- %Vflops = percentage of floating point operations due to vector floating point operations
- IBFR = Instruction Buffer Fetch Rate (megafetches/sec)
- IFact = Inline Factor (total calls to routine / average time spent in routine for each call)
- MC = number of memory conflicts
- MR = number of memory references
- Time = total CPU time (sec)

9. Coverage Analysis

One sample problem was supplied. A coverage analysis shows that this problem yielded an 87% segment coverage of C14H. Sample problems provided with typical simulation programs achieve only 35% to 50% coverage. A statement of software quality cannot be made for subroutines that have low coverage, i.e. large portions of the code are untested.

Note that subroutine *relrat* has 0% coverage. This routine is not tested with the supplied sample problem. Routine *opnfil* has under 40% coverage. All other routines exceeded 75% coverage. The following table shows the percent coverage for each routine.

Module Name	Number of Segments in module	Number of Segments Executed	Percent Segment Coverage
C14HA	9	7	77.8
C14DS	42	38	90.5
C14TIM	14	13	92.9
C14TRD	9	9	100.0
DATARE	5	5	100.0
GASDEV	6	6	100.0
GWT	14	12	85.7
ITEMP	27	27	100.0
ITER	15	15	100.0
LAYER	11	11	100.0
OPNFIL	28	11	39.3
PRESID	16	16	100.0
RAN1	7	6	85.7
RDGAS	3	3	100.0
READ1	3	3	100.0
READGL	1	1	100.0
READLH	25	24	96.0
RELOUT	3	3	100.0
RELRAT	6	0	0.0
SETH	5	5	100.0
SOURCE	10	10	100.0
UFUN	1	1	100.0
VAPOR	5	5	100.0
VFUN	1	1	100.0
Totals	266	232	87.2

```
                   0.20      0.40      0.60      0.80      1.00
               ----+----|----+----|----+----|----+----|----+----|
C14HA          |***************************************|         |
C14DS          |*********************************************    |
C14TIM         |**********************************************   |
C14TRD         |**************************************************
DATARE         |**************************************************
GASDEV         |**************************************************
GWT            |*******************************************      |
ITEMP          |**************************************************
ITER           |**************************************************
LAYER          |**************************************************
OPNFIL         |********************          |         |         |
PRESID         |**************************************************
RAN1           |*******************************************      |
RDGAS          |**************************************************
READ1          |**************************************************
READGL         |**************************************************
READLH         |************************************************ |
RELOUT         |**************************************************
RELRAT         |         |         |         |         |         |
SETH           |**************************************************
SOURCE         |**************************************************
UFUN           |**************************************************
VAPOR          |**************************************************
VFUN           |**************************************************
               ----+----|----+----|----+----|----+----|----+----|
```

```
     coverage = 0.          RELRAT

0.20 <= coverage < 0.40     OPNFIL

0.60 <= coverage < 0.80     C14HA

0.85 <= coverage < 0.90     GWT       RAN1

0.90 <= coverage < 0.95     C14DS     C14TIM

0.95 <= coverage < 1.00     READLH

     coverage = 1.00        C14TRD    DATARE    GASDEV    ITEMP     ITER
                            LAYER     PRESID    RDGAS     READ1     READGL
                            RELOUT    SETH      SOURCE    UFUN      VAPOR
                            VFUN
```

Program coverage for this run =0.87

10. Complexity Analysis

Some key metrics are the number of executable statements (sloc), the number of non-blank comments (ncomt), McCabe's extended cyclomatic complexity (vg2), the number of branching statements (cgoto, ugoto, bIF, and lIF), and Halstead's predicted number of errors in (re)writing the code (bhat). Measures are normalized per 100 executable statements for ease of comparison and are listed in the table below.

The branching measures for this code indicate very few unconditional GO TO statements and logical IFs. This code appears to be well structured.

Seventeen routines have a good ratio of non-blank comments to source code. Seven routines (*datarep, gasdev ,layer, ran1, ufun, vapor, vfun*) have less than 10 non-blank comments per 100 lines of source code.

McCabe's extended cyclomatic complexity (vg2), normalized per 100 lines of source code indicates relatively high values. Generally, the routines with the highest complexity are those most likely to have defects. As a guideline, normalized measures of 15 or greater should be considered complex. A software maintenance program should focus on those routines with the highest measures.

Complexity Report by Subprogram C14H

Name	loc	sloc	cmnt	ncomt	ncomt /sloc	vg2 /sloc	cgoto	cgoto /sloc	ugoto	ugoto /sloc	bIF	bif /sloc	lIF	lif /sloc	Bhat
C14Ha	126	29	61	57	196.6	10.3	0	0.0	1	3.4	2	6.9	0	0.0	1
C14DS	227	121	88	86	71.1	17.4	0	0.0	1	0.8	2	1.7	2	1.7	3
C14TIME	66	22	34	32	145.5	31.8	0	0.0	0	0.0	1	4.5	0	0.0	0
C14TRDS	79	21	44	44	209.5	23.8	0	0.0	0	0.0	0	0.0	0	0.0	0
DATAREP	40	30	0	0	0.0	10.0	0	0.0	0	0.0	0	0.0	0	0.0	1
GASDEV	17	16	0	0	0.0	18.8	0	0.0	1	6.3	1	6.3	1	6.3	0
GWT	47	35	6	5	14.3	17.1	0	0.0	1	2.9	3	8.6	0	0.0	0
ITEMP	62	35	13	13	37.1	40.0	0	0.0	0	0.0	0	0.0	1	2.9	1
ITER	64	35	11	11	31.4	22.9	0	0.0	0	0.0	0	0.0	0	0.0	1
LAYER	21	15	0	0	0.0	33.3	0	0.0	0	0.0	1	6.7	0	0.0	0
opnfil	208	59	131	118	200.0	27.1	0	0.0	0	0.0	8	13.6	0	0.0	0
PRESID	72	42	11	11	26.2	19.0	0	0.0	0	0.0	1	2.4	0	0.0	2
RAN1	29	24	1	1	4.2	25.0	0	0.0	0	0.0	1	4.2	1	4.2	0
rdgas	27	8	16	14	175.0	25.0	0	0.0	0	0.0	0	0.0	0	0.0	0
READ1	40	22	9	9	40.9	9.1	0	0.0	0	0.0	0	0.0	0	0.0	0
READGL	34	14	13	13	92.9	7.1	0	0.0	0	0.0	0	0.0	0	0.0	0
READLHS	69	40	26	23	57.5	25.0	0	0.0	3	7.5	4	10.0	2	5.0	0
relout	33	24	8	8	33.3	8.3	0	0.0	0	0.0	0	0.0	0	0.0	0
relrat	20	9	7	7	77.8	33.3	0	0.0	0	0.0	1	11.1	0	0.0	0
SETH	24	9	5	5	55.6	33.3	0	0.0	0	0.0	0	0.0	0	0.0	0
SOURCE	36	21	11	10	47.6	23.8	0	0.0	0	0.0	0	0.0	0	0.0	0
UFUN	19	13	1	1	7.7	7.7	0	0.0	0	0.0	0	0.0	0	0.0	0
VAPOR	47	28	4	2	7.1	10.7	0	0.0	0	0.0	0	0.0	0	0.0	0
VFUN	27	22	0	0	0.0	4.5	0	0.0	0	0.0	0	0.0	0	0.0	0

Legend of Metrics in Report

loc -- lines of code
sloc -- number of executable statements
cmnt -- total number of commnts
ncomt -- number of non-blank COMMENT statements
100*ncomt/sloc -- percent, nonblank comments to number of executable statements
100*vg2/sloc -- percent, extended complexity of number of executable statements
cgoto -- number of COMPUTED GO TO statements
100*cgoto/sloc -- percent, computed GOTO's to number of executable statements
ugoto -- number of UNCONDITIONAL GO TO statements
100*ugoto/sloc -- percent, unconditional GOTO's to number of executable statements
bIF -- number of BLOCK IF statements
100*bif/sloc -- percent, Block IF statements to number of executable statements
lIF -- number of LOGICAL IF statements
100*lif/sloc -- percent, logical IF statements to number of executable statements
Bhat -- Halstead's predicted number of errors in writing code