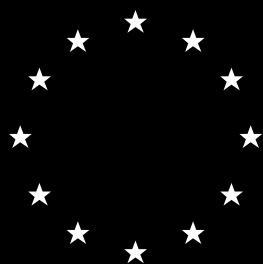

**User Source Term Interface
STERM/USER(NPP) in RODOS PV6final**

RODOS migration
contract

D R A F T
V e r s i o n 2 . 0



RODOS
REPORT

DECISION SUPPORT FOR NUCLEAR EMERGENCIES

User Source Term Interface STERM/USER(NPP) in RODOS PV6final
RODOS(RA1)-TN(04)-03

Claudia Landman

Forschungszentrum Karlsruhe GmbH
Institut für Kern- und Energietechnik
Postfach 3640
D-76021 Karlsruhe
Email: claudia.landman@iket.fzk.de

Draft, 23 August 2007

Management Summary¹

The program package described in this report, STERM/USER, is part of the RODOS PV6 interface STERM between the input of source terms for RODOS and the source term input for the Atmospheric Dispersion and Deposition Models (Near_ADMs) of RODOS. It controls and checks the source term input by a user, and calculates the complete Near_ADM source term from that input in both the automatic and the interactive operation modes of RODOS.

Note: Since RODOS Version PV6.0 Patch 07 it is not only possible to consider accident scenarios for nuclear power plants (nuclear accidents), but also radiological emergencies, for example the explosion of a radiological dispersion device. **The source term package STERM/USER as addressed in the report at hand applies exclusively to nuclear accidents. The source term package for STERM/UNONPP for radiological emergencies is described elsewhere.**

Report RODOS(RA1)-TN(04)-03 is part of the documentation of the source term treatment in RODOS PV6final which is summarised in report RODOS(RA2)-TN(04)-04.

¹ The work described in this report has been performed with support of the European Commission under the contract “Migration of RODOS to practical applicability for supporting decisions in operational emergency response to nuclear accidents” (RODOS migration), contract no. FIKR-CT-2000-00077

Table of Contents

Management Summary	4
1 Introduction	9
2 General Information and Overview	10
2.1 General Information	10
2.1.1 Summary	10
2.1.2 Developer	10
2.1.3 References	10
2.2 Overview	10
2.2.1 Background	10
2.2.2 Objectives	11
2.2.3 Proposed Methods and Procedures	11
2.3 Additional Information	11
2.3.1 Source Term Input Modes for nuclear accidents	11
2.3.2 Nuclides in STERM/USER	11
2.3.3 STERM Support Tasks	11
2.3.4 Denotations	13
2.4 Abbreviations and Acronyms	14
3 Operation Manual for STERM/UINPUT	15
3.1 Program Description	15
3.2 Description of Operation Modes and RODOS User Interfaces	15
3.2.1 Diagnosis	15
3.2.2 Interactive Prognosis	16
3.3 Connections to other Software	16
3.4 Help Functions	16
3.5 Description of Runs	16
4 Functional Specification and Data Specification Documents for STERM/UINPUT	17
4.1 Software Organisation	17
4.1.1 Overview	17
4.1.2 Program and Module Inventory	18
4.1.3 File Inventory	18
4.2 Data and Interface Description	19
4.2.1 Input Data	19
4.2.2 Output Data	19
4.2.3 Data Files	19
4.2.3.1 User Source Term Files	19
4.2.3.1.1 File Name and File Location	20
4.2.3.1.2 Mode-independent File Header Section	20
4.2.3.1.3 Mode-dependent File Section - Modes F1, F5	24
4.2.3.1.4 Mode-dependent File Section - Modes F3, F4	25
4.2.3.1.5 Mode dependent File Section - Modes F2, F6, F7	28
4.2.3.1.6 How to Upgrade Version PV5 User Source Term Files for RODOS PV6	29
4.2.3.1.7 How to Upgrade Version PV4 User Source Term Files for RODOS PV5	30
4.2.3.2 Aerosol Release Fraction File <in.aerosolfractions>	31
4.2.3.2.1 File Name and File Location	31
4.2.3.2.2 File Description	31
4.2.3.2.3 File Errors and File Maintenance	32
4.3 Module description	32
4.4 Description of Shared Memory	32

5	Program Maintenance and Test Plan for STERM/UINPUT	37
5.1	Operating Environment	37
5.1.1	Program Language	37
5.1.2	Programming Conventions	37
5.1.3	Support Software	37
5.2	Error Correction Procedures	37
5.3	Verification Procedures	38
5.4	Test Plan and Test Analysis Report	38
5.4.1	Tests with a Stand-alone Version of the Software	38
5.4.2	Tests with the Integrated Software within the RODOS system	38
5.5	Error report	38
6	Operation Manual for STERM/UMAKEQ	39
6.1	Program Description	39
6.2	Description of Operation Modes and RODOS User Interfaces	39
6.3	Connections to other Software	39
6.4	Help Functions	39
6.5	Description of Runs	39
7	Functional Specification and Data Specification Documents for STERM/UMAKEQ	40
7.1	Software Organisation	40
7.1.1	Overview	40
7.1.2	Program and Module Inventory	41
7.1.3	File Inventory	41
7.2	Data and Interface Description	41
7.2.1	Input Data	41
7.2.2	Output Data	42
7.2.3	Data Files	42
7.3	Module description	42
7.4	Description of Shared Memory	42
8	Program Maintenance and Test Plan for STERM/UMAKEQ	45
8.1	Operating Environment	45
8.1.1	Program Language	45
8.1.2	Programming Conventions	45
8.1.3	Support Software	45
8.2	Error Correction Procedures	46
8.3	Verification Procedures	46
8.4	Test Plan and Test Analysis Report	46
8.4.1	Tests with a Stand-alone Version of the Software	46
8.4.2	Tests with the Integrated Software within the RODOS System	46
8.5	Error report	46
9	APPENDIX 1: STERM Support Task SETINV for Setting Up a Radionuclide Inventory	47
9.1	Task	47
9.2	Design	47
9.3	Software	47
9.4	Input Data	48
9.5	Output Data	48
9.6	Verification and Tests	49
9.7	Error Report	49
9.8	Radionuclide Inventory Data Files	49
9.8.1	File Name, Location, Size	49
9.8.2	File Description	49
9.8.3	File Errors	51
9.8.4	File Maintenance, Implementation of New Files	51

10	APPENDIX 2: STERM Support Task SETCHN for Setting Up Decay Chain Information	52
10.1	Task	52
10.2	Design	52
10.3	Software	52
10.4	Input Data	53
10.5	Output Data	53
10.6	Verification and Tests	54
10.7	Error Report	54
10.8	Decay chain file <in.decaychains>	54
10.8.1	File Name, Location, Size	54
10.8.2	File Description	54
10.8.3	File Errors	56
10.8.4	File Maintenance - Whose Task, When Necessary, What To Do	56
11	APPENDIX 3: STERM Support Task SETRLG for Setting Up Release Group Information	57
11.1	Task	57
11.2	Design	57
11.3	Software	59
11.4	Input Data	59
11.5	Output Data	60
11.6	Verification and Tests	60
11.7	Error Report	60
11.8	Release Group File <in.releasegroups.structure> → since RODOS PV6final	61
11.8.1	File Name, Location, Size	61
11.8.2	File Description	61
11.8.3	File Errors	63
11.8.4	File Maintenance, Implementation of New Entries	64
11.9	Release Group NLS Files /in.releasegroups.keywords> → since RODOS PV6final	64
11.9.1	File Name, Location, Size	64
11.9.2	File Description	64
11.9.3	File Errors	64
11.9.4	File Maintenance, Implementation of New Entries	64
11.10	Release group information for RODOS-Lite → since RODOS PV6final	66
11.10.1	File Name, Location, Size	66
11.10.2	File Description	66
11.10.3	File Maintenance	66
12	Figures	68
13	References	70
	Document History	72

List of Figures

Figure 1: Input for and output from program package STERM/UINPUT 68
 Figure 2: Input for and output from program package STERM/UMAKEQ 69

List of Tables

Table 1: Source term input modes for nuclear accidents in RODOS PV6final 11
 Table 2: Single-input-mode user source term file: Header section 20
 Table 3: User source term file: Mode dependent file section - modes F1 and F5 24
 Table 4: User source term file: Mode dependent file section - modes F3 and F4 25
 Table 5: User Source term file: Mode dependent file section - modes F2, F6, and F7 28
 Table 6: Aerosol fraction file <in.aerosolfractions> 31
 Table 7: Subroutine inventory of STERM/UINPUT 32
 Table 8: Shared memory of STERM/UINPUT with support software 32
 Table 9: Support software for STERM/UINPUT 37
 Table 10: Subroutine inventory of STERM/UMAKEQ 42
 Table 11: Shared memory of STERM/UMAKEQ with support software 42
 Table 12: Support software for STERM/UMAKEQ 45
 Table 13: Subroutine inventory of SETINV 48
 Table 14: Support software for SETINV 48
 Table 15: Structure of a RODOS radionuclide inventory file 49
 Table 16: Subroutine inventory of SETCHN 53
 Table 17: Support software for SETCHN 53
 Table 18: Fraction of the decay chain file <in.decaychains> 54
 Table 19: Current nuclide release group schemes 57
 Table 20: Subroutine inventory of SETRLG 59
 Table 21: Support software for SETRLG 59
 Table 22: Release group file <in.releasegroups.structure> 61
 Table 23: Release group file </english/in.releasegroups.keywords> 64
 Table 24: Example for an NLS-release group file for RODOS-Lite 66

1 Introduction

In the real-time on-line decision support system RODOS, the interface STERM/NPP between the input of nuclear accident source terms for RODOS in general on the one hand and the source term input for the near range Atmospheric Dispersion and Deposition Models (Near_ADMs) on the other hand is realised with several functionally separate program packages:

- STERM/DIAG steers the construction of Near_ADM source terms in the automatic operation mode of RODOS.
- STERM/USER controls and checks the input of source terms by a user, and calculates the complete Near_ADM source term from the user input in both the automatic and the interactive operation modes.
- STERM/UTOADM delivers for each Near_ADM time step a respective section of the entire Near_ADM source term to the Near_ADM, also in both operation modes.

In addition, all program packages provide source term printout for the user.

Subject of the report is program package STERM/USER. The report applies to RODOS version PV6.0 onwards. Changes with respect to the predecessor versions PV5 and PV4 are pointed out in the text.

Note: Since RODOS Version PV6.0 Patch 07 it is not only possible to consider accident scenarios for nuclear power plants (nuclear accidents), but also radiological emergencies, for example the explosion of a radiological dispersion device. **The source term package STERM/NPP as addressed in the report at hand applies exclusively to nuclear accidents.** The source term package STERM/UNONPP for radiological emergencies is described elsewhere.

2 General Information and Overview

2.1 General Information

2.1.1 Summary

Program package STERM/USER is part of the source term interface STERM for nuclear accidents of the real-time on-line decision support system RODOS [1]. It controls and checks the source term input by a user, and calculates the complete Near_ADM source term from that input in the automatic as well as in the interactive operation modes of RODOS.

2.1.2 Developer

STERM/USER was developed by the RODOS Team at Forschungszentrum Karlsruhe (FZK), Institut für Kern- und Energietechnik (IKET), in Karlsruhe, Germany.

2.1.3 References

An overview of the source term treatment for releases from Nuclear Power Plants ("nuclear accidents") in RODOS PV6final is given in [2]. That report also contains a description of the physical formalism and a list of all STERM and related documents.

2.2 Overview

2.2.1 Background

The operational and comprehensive real-time on-line decision support system RODOS is designed for use in a real emergency as well as for training and exercises. With respect to the input of source terms, this means that RODOS must be able to deal with frequently used data formats and parameterisations in order to make adaptations to real or fictive scenarios as easy as possible.

On the other hand, the source term input for the near range Atmospheric Dispersion and Deposition Models (Near_ADMs) of RODOS, has a fixed format prescribed by the Near_ADMS, so that an interface is required between the RODOS source term input and the Near_ADM source term input.

Until RODOS PV4, this interface was managed by code package SourceInfo/I [3]. Due to the transition to long source terms and long prognoses in RODOS PV5, SourceInfo/I had been split into the three parts STERM/DIAG, STERM/USER, and STERM/UTOADM. In RODOS PV6, the source term input via intialisation windows is made with dynamic instead of static ones, which required a further splitting of STERM/USER

into two functionally totally disjunct parts, STERM/UINPUT and STERM/UMAKEQ².

In RODOS PV6final, the Native Language Support (NLS) via external files was introduced also for the source term code, and the former ".ff"-files became normal "FORTRAN .f"-files.

2.2.2 Objectives

Within the RODOS system, STERM/USER fulfils three tasks:

- (1) The control of the flow of the source term input with the dynamic intialisation windows, for the diagnosis as well as for the prognosis Near_ADM.
- (2) The check of the source term input data from all RODOS user interfaces.
- (3) The calculation of the complete Near_ADM source term from the user input , fo all user interfaces, for the diagnosis as well as for the prognosis Near_ADM.

2.2.3 Proposed Methods and Procedures

The programming language is FORTRAN 77. The STERM interface will run under the operating systems HP-UX and Linux.

2.3 Additional Information

2.3.1 Source Term Input Modes for nuclear accidents

For the input of the released activity seven different parameterisations are possible. Each parameterisation defines one "source term input mode". Table 1 lists the source term input modes currently implemented in RODOS and their identification key (the numbering of the keys is historical).

Table 1: Source term input modes for nuclear accidents in RODOS PV6final

key	specification of the activity release ¹⁾
F1	Fractions [%] of the initial activity inventory released for some set of nuclide release groups known to RODOS ²⁾ .
F5	Released activity [Bq] for the sum of near range nuclides ³⁾ in some set of nuclide release groups known to RODOS ²⁾ .
F3	Released activity [Bq] for I-131 and the sums of near range nuclides ³⁾ in the noble gas and total aerosol ⁵⁾ group, together with release fractions for a fixed set of aerosol release groups ⁴⁾ .
F4	Released activity [Bq] for the sums of near range nuclides ³⁾ in the noble gas, iodine

² The letter "Q" in UMAKEQ stands for the German word "Quellterm", which means source term.

	and total aerosol ⁵⁾ group, together with release fractions for a fixed set of aerosol release groups ⁴⁾ .
F2	Activity release [Bq] for individual nuclides - with inventory reference.
F6	Activity release [Bq] for individual nuclides - without inventory reference.
F7	Activity release rate [Bq/s] for individual nuclides - without inventory reference.
Notes	
1) All input modes also need the specification of the start of initial release, the release height, the released thermal power, the vertically released volume flux, the vent area of the release, and the iodine fractions.	
2) The sets of nuclide release groups (also called "nuclide release blocks") known to RODOS are shown in Table 19.	
3) Near range nuclides are a subset of 1 to 25 nuclides from all possible nuclides selected for a run. If, for instance, a release of 10E18 Bq is specified for the noble gas group, the results would be different if Kr- 88 only or Kr- 88, Xe-133, Xe-135 were selected as near range nuclides.	
4) These are the aerosol groups in nuclide release block "DRS-A_7_GROUPS" of Table 19.	
5) "Aerosols" means here fission products released to the atmosphere as solid or liquid suspensions in air via a carrier (e.g. CsJ, NaOH etc.).	

2.3.2 Nuclides in STERM/USER

RODOS contain data for a large number of free or chemically bound radionuclides, which are called "RODOS nuclides", or, by historical reason, also "FixData nuclides"³⁾. However, all application-oriented near-range calculations with RODOS are carried out with a subset currently consisting of between 1 and 25 user selectable nuclides, which are termed "near range nuclides". The Near_ADM source term produced by the STERM source term interface contains the activity release rates for the near range nuclides only.

2.3.3 STERM Support Tasks

The near range Near_ADMs of RODOS demand the activity release information in the terms of release rates for individual radionuclides, which are a user-defined subset of 1 to 25 so-called "near range nuclides" of all possible "RODOS nuclides" or "FixData nuclides". Information about the RODOS nuclides and the near range nuclides is set up by program package "SETNUK" (SETup of NUKlides), that is described in report [4].

Source term input modes F1 to F5 of Table 1 require a radionuclide inventory of the reactor considered for generating the nuclide-specific information for the Near_ADMs. Radionuclide inventory information is set up by task SETINV ("SETup of INVentry", see Chapter 9).

In all time spans between the reactor shutdown and the releases of radionuclides, STERM takes account of the in-growth of nuclides from radioactive decay chains. Decay chains information is set up by task SETCHN ("SETup of decay CHaiNs", see Chapter 10).

³⁾ Since RODOS PV5, the RODOS nuclide definition description is no longer part of the FixData base of RODOS, but is kept as ASCII files on disk.

Several source term input modes of Table 1 refer to "nuclide release groups". Release group information is set up by task SETRLG ("SETUp of ReLease Groups", see Chapter 11).

For source term input modes F1 to F5 of Table 1, STERM uses an enhanced nuclide list called "STERM nuclides" for the calculation of the Near_ADM source term from the input source term, that consists of the following nuclides:

- **Primary nuclides:** (+) all *near range nuclides*, but (-) excluding those marked for exclusion by program package SETRLG (i.e. all nuclides assigned to the so-called "waste bin release group", see below).
- **Auxiliary nuclides:** (+) all RODOS nuclides which were found by program package SETCHN as feeders for above primary nuclides via some decay chain, and all which again feed the feeders, and so on, upwards the whole chain, but (-) excluding again those marked for exclusion by SETRLG.

The possibility to exclude radionuclides from the calculations by assigning them to the waste bin release group is technically necessary, because for SETRLG all FixData base nuclides must be assigned to one and only one nuclide release group in order to formally control the completeness of the specifications. All radionuclides not fitting into the release group scheme of a given release block can be put into the waste bin⁴. Then, the activity release rates for near range nuclides in the waste bin release group will always be zero, and a warning message is issued to remind the user of this special setting.

2.3.4 Denotations

All software packages mentioned in this report are collections of FORTRAN subroutines. It is distinguished between "program packages", that can (but need not) be operated as "program modules" in the RODOS external program environment, and "code packages", that are simply libraries with auxiliary FORTRAN subroutines.

The RODOS source term interface STERM consists of the three packages STERM/USER, STERM/UTOADM, and STERM/DIAG. If in a given context it is only important to know that the "RODOS source term interface" is involved, it is summarily referred to as "STERM"; otherwise, the package name is mentioned explicitly.

STERM/USER consists of the two packages, STERM/UINPUT and STERM/UMAKEQ. If in a given context it is only relevant to know that the "user source term package" is involved, it is summarily referred to as "STERM/USER"; otherwise, the package name is mentioned explicitly.

File names and directory paths are given in italics between the two brackets <>. The brackets themselves are not part of the name.

⁴ This is for instance the case for the FixData items tritiated hydrogen (HT) and tritiated water (HTO) in the release group scheme "DRS-A_7_Groups" of Table 19 on page 57.

2.4 Abbreviations and Acronyms

Near_ADM	Atmospheric Dispersion and Deposition Model for the near range in RODOS
ASY	Analysing subSYstem of RODOS
NLS	National Language Support
SETNUK	program package " <u>Nuclide Information</u> "
RODOS	Real-time On-line DecisiOn Support system
run_ID	RODOS signature of run
SETCHN	code package "SETup of decay CHaiNs"
SETINV	code package "SETup of INVentory"
SETNUK	code package "SETup of NUKlides"
SETRLG	code package "SETup of ReLease Groups"
STERM	RODOS Source TERM Interface
STerm	Source Term
SHM	Shared Memory
user_ID	RODOS signature of user

3 Operation Manual for STERM/UINPUT

3.1 Program Description

From RODOS version 6.0 onwards, all source term input by a RODOS A-user is made interactively with dynamic initialisation windows that open automatically at run time [5]. The flow of the user input with the dynamic initialisation windows is controlled by program package STERM/UINPUT that also performs checks on the input.

To start with, the user must put in the time span between the end of the chain reaction and the start of the initial release⁵.

Then, the user must define between 1 and 24 time intervals covering the time span between the onset and the termination of all significant releases from the reactor.

Finally, for each time interval values must be given for a range of source term items listed below for. In comparison with RODOS PV5.0, there are two additional items.

- The release height.
- The released thermal power.
- The vertically released volume flux (SINCE RODOS PV6.0).
- The vent area of the release to the atmosphere (SINCE RODOS PV6.0).
- The fractions of elementary, organically bound and aerosol iodine of the total iodine released ("relative iodine fractions").
- The activity released from the reactor, in one of the seven input modes shown in Table 1.

3.2 Description of Operation Modes and RODOS User Interfaces

Program package STERM/UINPUT is part of the Analysing Subsystem (ASY) of RODOS. In the diagnosis mode, it belongs to program RMC3DIAG. In the automatic prognosis mode, it is not used. In the interactive prognosis mode it belongs to each Near_ADM.

3.2.1 Diagnosis

In the Automatic Mode, the user can apply only the dynamic initialisation windows for putting in source terms. All source term items can be accessed

⁵ Only in the interactive operation mode of RODOS. In the automatic mode, this information is set via the RODOS system interface at a higher level in the input hierarchy; see also Chapter 4.1).

except the "time span between the end of the chain reaction and the start of the initial release"⁶, which is set elsewhere in RODOS [6].

3.2.2 Interactive Prognosis

When the user applies the dynamic initialisation windows, all source term items can be accessed.

When the user applies the web-based user interface [7], the source term comes via shared memory to STERM/UINPUT, and all interactions with the initialisation windows must be suppressed. Nevertheless, the user source term must be checked, and the "approved user source term" must be generated. As a consequence, STERM/UINPUT contains code that distinguishes between the two source term user interfaces of RODOS.

3.3 Connections to other Software

STERM/UINPUT gets input from

- The RODOS shared memory (default settings).
- The embedding external program of ASY (steering parameters, site information etc.).
- The dynamic initialisation windows of RODOS (user source term).
- The web-based user interface (user source term).
- Several STERM support tasks.

STERM/UINPUT produces output for

- The RODOS graphics (tables with source term and inventory information).
- Program package STERM/UMAKEQ.

3.4 Help Functions

Help buttons are contained in the dynamic initialisation windows for the source term input.

3.5 Description of Runs

How to start an ASY run with source term input with the dynamic initialisation windows is described in [5].

The source term input with the web-based user interface is described in [7].

⁶ Shared Memory variable BEGFRE of data class DSTRMX. The steering subroutine of STERM/UINPUT contains a flag in the argument list that can be used by the caller to suppress user input of BEGFRE.

4 Functional Specification and Data Specification Documents for STERM/UINPUT

4.1 Software Organisation

4.1.1 Overview

Figure 1 illustrates the input from STERM support tasks and from file for STERM/UINPUT, and the output from STERM/UINPUT.

As can be seen from the figure, STERM/UINPUT gets information from SETNUK (RODOS nuclides and near range nuclides), from SETCHN (STERM nuclides), and from SETRLG (nuclide release groups).

From the web-based user interface, STERM/UINPUT receives the user-specified source term from the RODOS shared memory (not shown in Figure 1).

With the dynamic initialisation windows, users interact directly with STERM/UINPUT. They can build up a source term from scratch by hand. Or they can begin the source term input with a "start-up source term" (see Figure 1) taken from one of the following sources (<FILENAME>⁷ denotes a source term file with a name according to the rules in Chapter 4.2.3.1.1; and *user_ID* the RODOS user identification):

- From the RODOS library: These source terms are stored as files with names <FILENAME> in directory <~rodos/roextern/data/sourceterm/FixData/FILENAME>.
- From the public source term library: These source terms are stored as files with names <FILENAME> in directory <~rodos/roextern/data/sourceterm/public/>.
- From the private source term library of the user: These source terms are stored as files with names <user_ID.FILENAME> in directory <~rodos/roextern/data/sourceterm/input/>.
- The last source term the user has applied before (the file name is explained further below).

In the interactive input phase, STERM/UINPUT communicates with the user by calling a procedure⁸ of RODOS that opens an Initialisation Window specified by STERM/UINPUT for the user. In such a window, the user has either to decide between different options offered, for example, between the above possibilities to take a start-up source from. Or the user has to type in data according to input masks presented in the window. When the user closes the window, control returns to STERM/UINPUT, that, first of all, performs checks on the input: If an error is detected, the user is forced to correct it interactively. After that, STERM/UINPUT

⁷ The name <FILENAME> of the start-up file is kept in Shared Memory variable CQTFIN.

⁸ C-procedure win_request in Table 9.

proceeds to the next appropriate window, or to the termination of the interactive phase.

For source terms that were specified with the web-based user interface, the same checks are performed as for source terms from the dynamic initialisation windows. However, if an error is detected, STERM/UINPUT only flashes a message indicating the error condition on the RODOS screen, writes a text explaining the problem(s) to the RODOS run protocol, and then terminates.

On termination of the interactive phase, or the pure checking phase, STERM/UINPUT stores the "approved user source term":

- Always in a file with a pre-defined name in RODOS directory `<~rodos/roextern/data/sourceterm/input/>`. In the international version of the executable, the file name is `<user_ID.F0.MyLastSTerm>`, but in the German version it is `<user_ID.F0.MeinLetzterQTerm>`⁹. Therefore, the file name is contained in the RODOS shared memory¹⁰ for information of the follow-up package STERM/UMAKEQ.
- On request of the user, addionally under a different name in the private or public source term library (but not in the RODOS library: This, a RODOS system administrator can do, but only manually).

The file with the "approved user source term" has a format corresponding to one of the source term input modes from Table 1.

STERM/UINPUT also prints tables with user source term information to file `<Prognosis|Diagnosis.STermFromUser>` in directory `<~rodos/roextern/outall/user_Id/run_Id/>`. This file is accessible with the Graphics Manager of the RODOS system, see [5], Chapter on "Prognosis Results Accessible with the RODOS Graphics".

4.1.2 Program and Module Inventory

STERM/UINPUT is imbedded in a module of an external program of ASY, and not operated as a program or module on its own in RODOS.

4.1.3 File Inventory

Input: File with the "start-up source term". For the file name and directory see Chapter 4.1.1. The file size depends on the source term input mode; it is typically about 0.5 kilobytes¹¹ to some 10 kilobytes.

Output: File with the "last applied user source term". For the file name and directory see Chapter 4.1.1. The file size depends on the source term input mode; it is typically about 0.5 kilobytes to some 10 kilobytes.

⁹ The key F0 simply characterises the "last source term applied". The respective input mode key is anyway contained in the file (variable IRLTYP, see Chapter 4.2.3.1.2).

¹⁰ Shared Memory variable CQTFIL of data class CQTRMX.

¹¹ 1 kilobyte = 1024 bytes.

4.2 Data and Interface Description

4.2.1 Input Data

- The SHM section with the "Nuclide Information" (see Table 8).
- The COMMON blocks filled by the support tasks SETCHN (see Chapter 10.5) and SETRLG (see Chapter 11.5).
- ASY internal variable RUSER of a COMMON block contained in ASY-include member 'xruser.inc' for the information whether the current run requires interaction with the initialisation windows, or not. From RUSER STERM/UINPUT sets a SHM variable, IQTDYN of data class DSTRM0, to activate or suppress the interaction with the windows.
- The File with start-up source term (if addressed).
- Steering parameters in the argument list of the steering routine.

The SHM variables accessible with the RODOS user interfaces are described in the layout-files of the embedding ASY program. The variables lie in data classes CQTRMX, CSTRMX, DSTRM0, and DSTRMX (see Table 8); they are the same for all Near_ADMs, and also for all RODOS user interfaces. The layout files can be found in RODOS directory `<~rodos/rogui>`.

4.2.2 Output Data

- The name of the current internal file with the "last applied user source term" in SHM variable CQTFIL of data class CQTRMX.
- The file containing the "last applied user source term".

4.2.3 Data Files

4.2.3.1 User Source Term Files

A source term file created by the user contains all source term data for only one of the seven input modes of Table 1.

Chapter 4.2.3.1.1 explains the file name syntax and the location of user source term files in the RODOS system.

Each source term file starts with a section called "file header". The format of the file header is the same for all input modes. Chapter 4.2.3.1.2 describes the file header section. The physical data in this section are only described briefly with respect to their functional specification; for further details about meaning, application and value range see [5].

A data section for all data depending on the input mode follows the file header section. Chapters 4.2.3.1.3 to 4.2.3.1.5 describe this section of the user source term files for the different input modes. The physical data in this section are only described briefly with respect to their functional specification; for further details about meaning, application and value range see [5].

How to upgrade version PV5 source term files for use in RODOS PV6.0 is explained in Chapter 4.2.3.1.6.

As in RODOS PV6.0 there is no longer a need for the user to create user source term files manually, the Chapter on detection and handling of file errors, file maintenance, and help for file implementation, that was contained in predecessor report [3], are now obsolete und were removed.

4.2.3.1.1 File Name and File Location

File name and location are under full control of STERM/UINPUT.

The file name starts with one of the source term input mode codes shown in column 1 of Table 1, followed by a dot, followed by up to 29 arbitrary alphanumerical characters. The allowed characters are 0 - 9, a - z (upper and lower case), dot, underscore.

In RODOS version PV6.0, the location of the file depends on its purpose and accessibility, see Chapter 3.1.

4.2.3.1.2 Mode-independent File Header Section

Table 2 contains the header section of some source term file and the respective formats. Explanations follow after the table!

The example given in the Table shows a file header for input mode F1.

Table 2: Single-input-mode user source term file: Header section

format	file content (example)
F-01	# =====
F-02	NCOMM= 4 ***** Number of comment lines to follow this line *****
F-01	# -----
F-01	# File <F6.STERMout> created by {STerm/User}
F-01	# --> RODOS MODEM source term file (ASCII) in F6_Extended format <---
F-01	# =====
F-01+	CQTORI=SouTerEx
F-01+	IRLTYP= 0 0 0 0 0 1 0 0 0 0 0 0
F-05\$	CRLGID=DRS-A_7_GROUPS
F-06\$	IRLGRP= 7
F-07\$	COMFR1=Demo source term. Sum =52.5% noble gases; 0.75% iodine, aerosols
F-07\$	COMFR2=After EOC+3.75hrs. filt.vent. 01hr.; no rel. 03.5hrs.;
F-01	# BEGIN OF RELEASE ("FREISETZUNG") AFTER EOC [h]
F-07b\$	BEGFRE= 3.75000E+00
F-01	# NUMBER OF SOURCE TERM ("QT") USER INPUT TIME INTERVALS
F-06\$	NQTUIT= 8
F-01	#LOWER EDGES OF SOURCE TERM ("QT") USER INPUT TIME INTERVALS [h]
F-01+	#QTUIT1=
F-10	0.00 0.50 4.50 5.00 5.50
F-10	6.00 6.50 7.00
F-01	#UPPER EDGES OF SOURCE TERM ("QT") USER INPUT TIME INTERVALS [h]
F-01+	#QTUIT2=
F-10	0.50 1.00 5.00 5.50 6.00
F-10	6.50 7.00 7.50
F-01	# RELEASE HEIGHT [m]
F-01+	#HFSIN =
F-08	1.50000E+02 1.00000E+01 1.00000E+01 1.00000E+01 1.00000E+01
F-08	1.00000E+01 1.00000E+01 1.00000E+01
F-01	# RELEASED THERMAL ENERGY [MW]
F-01+	#QHIN =
F-08	0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
F-08	0.00000E+00 0.00000E+00 0.00000E+00
F-01	# VERTICALLY RELEASED VOLUME FLUX [m**3/s]
F-01+	#VOLFIN=
F-08	0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
F-08	0.00000E+00 0.00000E+00 0.00000E+00
F-01	# VENT AREA OF RELEASE TO THE ATMOSPHERE [m**2]
F-01	#VENTIN=

F-01+	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
F-08	0.00000E+00	0.00000E+00	0.00000E+00		
F-08	#				IODINE FRACTIONS [%]
F-01	#ANTJIN=				
F-01+	#				ELEMENTARY
F-01	1.00000E+02	1.00000E+02	1.00000E+02	1.00000E+02	1.00000E+02
F-08	1.00000E+02	1.00000E+02	1.00000E+02		
F-08	#				ORGANICALLY BOUND
F-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
F-08	0.00000E+00	0.00000E+00	0.00000E+00		
F-08	#				AEROSOLS
F-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
F-08	0.00000E+00	0.00000E+00	0.00000E+00		
F-08	#				-----
F-01	#				NOW FOLLOWS THE SECTION WITH DATA DEPENDING ON THE INPUT MODE
F-01	#				-----

Formats (Character "^" represents the blank character, "d" some digit 0...9)

F-01 : Comment line with up to 64 arbitrary characters.
F-01+: Line with fixed content - type exactly as shown! (New in RODOS PV6FINAL)
F-02 : 9 arbitrary characters, 1 right-adjusted integer with 4 digits.
F-03 : 8 arbitrary characters, 8 characters for file type flag.
F-04 : 8 arbitrary characters, 12 right-adjusted integers with 2 digits each.
F-05 : 8 arbitrary characters, 16 characters for release block name, left adjusted.
F-06 : 8 arbitrary characters, 1 right-adjusted integer with 2 digits each.
F-07 : 8 arbitrary characters, 80 characters for source term comment.
F-07b: 8 arbitrary characters, 1 floating point number with 12 digits
"{sign}d.dddddE+dd".
F-08 : 5*(floating point number with 12 digits and trailing blank "{sign}d.dddddE+dd^").

(SINCE RODOS PV6FINAL) Postfix "\$" means "type the keyword part exactly as shown". The keyword part comprises all alphanumerical characters (including the Character '#' and the blank character until inclusively the character '=').

The file syntax is described below. The character "^" represents the blank character. **Please note that** - due to requirements of the new RODOS-Lite user interface - **there may be changes between RODOS versions PV6.0 and PV6final in the degree of freedom in the character (non-numerical) part of line(s).** See the format description part of the Table!

Leading lines

- **Line 1:** Obligatory comment line ("---...---" in the Table).
- **Line 2:** Number of comment lines to follow (variable NCOMM).
- **Lines 3 to 2+NCOMM:** Arbitrary comment lines.

Data lines (D0=2+NCOMM)

- **Line D0+1:** Always exactly as shown in the example.
- **Line D0+2:** Always exactly as shown in the example.
- **Line D0+3:** Flag to signal the source term input mode (variable IRLTYP; values are shown below).

F1: 1 0 0 0 0 0 0 0 0 0 0 0 / F2: 0 1 0 0 0 0 0 0 0 0 0 0
F3: 0 0 1 0 0 0 0 0 0 0 0 0 / F4: 0 0 0 1 0 0 0 0 0 0 0 0
F5: 0 0 0 0 1 0 0 0 0 0 0 0 / F6: 0 0 0 0 0 1 0 0 0 0 0 0
F7: 0 0 0 0 0 0 1 0 0 0 0 0

- **Line D0+4:** Release block name (16 characters).
The list of allowed release block names is shown in column 1 of Table 19). A name from Table 19 is specified even for input modes F6 and F7 (SINCE RODOS PV5.0).
- **Line D0+5:** Number of release groups in the block (variable IRLGRP).
The second column in Table 19 shows the number of release groups in each release block. IRLGRP contains the number shown in the table for the selected release block. The correct value for IRLGRP is specified even for input modes F6 and F7 (SINCE RODOS PV5.0).
- **Lines D0+6,D0+7:** Comments to describe the source term (80 characters per line); appears in text output of a run.
- **Line D0+8:** Obligatory comment line.
- **Line D0+9:** Time span between the end of the chain reaction and the start of the release.
- **Line D0+10:** Obligatory comment line.
- **Line D0+11:** Number of release time input intervals to follow (variable NQTUIT). More information at end of current Chapter.
- **Next 2 lines:** Obligatory comment lines.
- **Next lines:** Lower edges of NQTUIT release time input intervals. Five data entries per line. More information at end of current Chapter.
- **Next 2 lines:** Obligatory comment lines.
- **Next lines:** Upper edges of NQTUIT release time input intervals. Five data entries per line. More information at end of current Chapter.
- **Next 2 lines:** Obligatory comment lines.
- **Next lines:** Release height for NQTUIT time intervals, five data entries per line. *Comment:* A value "-1" signals that the stack height of the nuclear power plant under consideration shall be substituted for that time interval at runtime of RODOS.
- **Next 2 lines:** Obligatory comment lines.
- **Next lines:** Released thermal energy for NQTUIT time intervals, five data entries per line.
- **Next 2 lines (NEW LINES IN RODOS PV6.0):** Obligatory comment lines.
- **Next lines (NEW ITEM IN RODOS PV6.0):** Vertically released volume flux for NQTUIT time intervals, five data entries per line. *Comment:* A value "-1" signals that the average volume flux through the stack vent opening for the site under consideration shall be substituted for that time interval at runtime of RODOS.
- **Next 2 lines (NEW LINES IN RODOS PV6.0):** Obligatory comment lines.
- **Next lines (NEW ITEM IN RODOS PV6.0):** Vent area of the release to the atmosphere for NQTUIT time intervals, five data entries per line. *Comment:* A value "-1" signals that the area of the stack vent opening

for the site under consideration shall be substituted for that time interval at runtime of RODOS.

- **Next 2 lines:** Obligatory comment lines.
- **Next lines:** Relative iodine fractions for elementary iodine, organically bound iodine, iodine in aerosol form, for NQTUIT time intervals, five data entries per line. The blocks of data for the three iodine forms are preceded by one obligatory comment line.
- **Last 3 lines:** Obligatory comment lines.

How to specify NQTUIT, QTUIT1, QTUIT2

In RODOS, the release height, the released thermal power, the iodine fractions, and the released activity are given for time intervals with variable duration. For all time periods without data specification, it is assumed that there is no release.

Between 1 and 24 time intervals can be used for the specification of a source term. If a release would be constant over time, one time interval suffices. If a release would strongly vary over time, the provider must condense the information to fit into the 24 time boxes maximally available. There is no upper limit for the duration of a source term, provided that the specification fits into 24 intervals.

The lower and upper boundaries of the time intervals are given in terms of hours after the start of the release. The decimal precision is two trailing digits, which means that 0.01 is the smallest increment possible. The overlap of interval boundaries is not allowed and will be detected and rejected at runtime of RODOS.

Intervals j with $QTUIT2(j) < QTUIT1(j)$ are not allowed. Boundary specifications of this type will be detected and rejected at runtime of RODOS.

For intervals j with $QTUIT2(j) = QTUIT1(j)$, all data in the corresponding input position are skipped at runtime of RODOS.

Intervals j with $QTUIT2(j) > QTUIT1(j)$ are valid for source term input. At runtime of RODOS, all data in the corresponding input position are interpreted.

The intervals need not to be ordered with respect to time. *Example:* $NQTUIT = 2$, $QTUIT1(1,2) = (7.0, 0.0)$, $QTUIT2(1,2) = (10.0, 5.0)$ would be a valid specification for the two time intervals [0.0 to 5.0] hours and [7.0 to 10.0] hours.

The release specification must start with the begin of the release!

In valid intervals j , there must be a non-zero input for the activity release, which means at least one entry for the activity release must be greater than zero!

NOTE: A source term file imported from an external source may contain no-release periods filled with zero release values, or some time periods for which one would like to specify other values. Making use of the

possibilities to mark intervals for skipping or adding intervals at the end with boundaries lower in time that the previous ones can facilitate the adaptation of such files for RODOS.

4.2.3.1.3 Mode-dependent File Section - Modes F1, F5

Table 3 shows the input mode dependent source term file section for input modes F1/F5, and the respective formats. The file structure is identical for modes F1 and F5 (the first line in the Table "PROZIN [%] or ABQGIN [Bq] =" is a comment line without functional relevance¹²). The data follow immediately after the file header section described in Chapter 4.2.3.1.2. Explanations follow after the table!

The table below contains for the eight time intervals and the release groups of release block "DRS-A_7_GROUPS" from the example file header of Table 2 the mode-dependent data input for source term input mode F1 ↔ Fractions [%] released of the initial core inventory.

Table 3: User source term file: Mode dependent file section - modes F1 and F5

format	file content (example: input mode F1 with "PROZIN")
F-01+	#PROZIN= (F5:"#ABQGIN=")
F-01	# Noble gases
F-08	1.25000E+01 1.25000E+01 7.20000E+00 6.40000E+00 4.80000E+00
F-08	3.60000E+00 3.60000E+00 1.90000E+00
F-01	# Iodine
F-08	5.00000E-04 5.00000E-04 1.95000E-01 1.65000E-01 1.35000E-01
F-08	1.05000E-01 1.05000E-01 4.50000E-02
F-01	# Alkaline Metals
F-08	5.00000E-04 5.00000E-04 1.95000E-01 1.65000E-01 1.35000E-01
F-08	1.05000E-01 1.05000E-01 4.50000E-02
F-01	# Tellurium and Antimony
F-08	3.27500E-04 3.27500E-04 1.27725E-01 1.08075E-01 8.84250E-02
F-08	6.87750E-02 6.87750E-02 2.94750E-02
F-01	# Alkaline Earths
F-08	5.50000E-05 5.50000E-05 2.14500E-02 1.81500E-02 1.48500E-02
F-08	1.15500E-02 1.15500E-02 4.95000E-03
F-01	# Ruthenium Group
F-08	2.25000E-04 2.25000E-04 1.13100E-02 9.57000E-03 7.83000E-03
F-08	6.09000E-03 6.09000E-03 2.61000E-03
F-01	# Lanthanides
F-08	4.50000E-06 4.50000E-06 1.75500E-03 1.48500E-03 1.20600E-03
F-08	9.45000E-04 9.45000E-04 4.05000E-04
Formats (Character "^" represents the blank character, "d" some digit 0...9)	
F-01	: 81 arbitrary characters
F-01+	: Line with fixed content - type exactly as shown! (NEW IN RODOS PV6FINAL)
F-08	: 5*(floating point number with 12 digits and trailing blank "{sign}d.dddddE+dd^")

¹² It is variable IRLTYP in the file header section that tells STERM/USER how to interpret the data!

The file syntax is described below. The character "^" represents the blank character. **Please note that** - due to requirements of the new RODOS-Lite user interface - **there may be changes between RODOS versions PV6.0 and PV6final in the degree of freedom in the character (non-numerical) part of line(s).** See the format description part of the Table!

Leading line

- **Line 1:** Obligatory comment line.

Then, for each release group in the same order as shown in Table 19 for the release block selected in the file header section (!)

- **One** obligatory comment line.
- Either
 - (F1) **Fractions released of the initial core inventory for the corresponding release group, [%],**
 - or
 - (F5) **Released activity for the sum of near range nuclides in the corresponding release group, [Bq],**
 for the NQTUIT time intervals defined in the file header section, five data entries per line.

4.2.3.1.4 *Mode-dependent File Section - Modes F3, F4*

NEW CONSTRAINT SINCE RODOS PV6.0: Input modes F3 and F4 can be used only in connection with the nuclide release block "DRS-A_7_GROUPS" from Table 19.

Table 4 shows the input mode dependent source term file section for input modes F3/F4, and the respective formats. The file structure is identical for modes F3 and F4 (the line in the Table with the text "I-131 or Iodine" is a comment line without functional relevance¹³). The data follow immediately after the file header section described in Chapter 4.2.3.1.2. Explanations follow after the table!

The table below contains for the eight time intervals and the release groups of release block "DRS-A_7_GROUPS" from the example file header of Table 2 the mode-dependent data input for source term input mode F4 ↔ Released activity [Bq] for the sum of near range nuclides in the nuclide groups noble gases, iodine, and total aerosol, together with release fractions for the various aerosol groups in nuclide release block "DRS-A_7_GROUPS".

Table 4: User source term file: Mode dependent file section - modes F3 and F4

¹³ It is variable IRLTYP in the file header section that tells STERM/USER how to interpret the data!

format	file content (example: input mode F4 with "Iodine")
F-01	# -----
F-01	# AEROSOL FRACTIONS
F-01	# -----
F-04\$	IAEMOD= 0 1
F-05\$	CAENAM=DRS-A_FK2
F-01	# -----
F-01	# ACTIVITY RELEASE
F-01	# -----
F-01+	#ABQ3IN AI131IN=
F-01	# Noble_gases
F-08	1.41130E+18 1.40783E+18 8.06517E+17 7.16388E+17 5.37487E+17
F-08	4.03572E+17 4.04219E+17 2.13791E+17
F-01	# I-131 or Iodine
F-08	9.92862E+13 9.74583E+13 3.31472E+16 2.76111E+16 2.22462E+16
F-08	1.70436E+16 1.67933E+16 7.09341E+15
F-01	# Aerosol
F-08	3.14029E+13 3.03007E+13 1.03696E+16 8.67252E+15 7.01714E+15
F-08	5.40640E+15 5.36264E+15 2.28170E+15
<p>Formats (Character "^" represents the blank character, "d" some digit 0...9)</p> <p>F-01 : 81 arbitrary characters</p> <p>F-01+: Line with fixed content - type exactly as shown! (NEW IN RODOS PV6FINAL)</p> <p>F-04 : 8 arbitrary characters, 2 right-adjusted integers with 2 digits each</p> <p>F-05 : 8 arbitrary characters, 16 characters for aerosol entry name (see text)</p> <p>F-08 : 5*(floating point number with 12 digits and trailing blank "{sign}d.dddddE+dd^")</p> <p>(SINCE RODOS PV6FINAL) Postfix "\$" means "type the keyword part exactly as shown". The keyword part comprises all alphanumerical characters (including the Character '#' and the blank character until inclusively the character '=').</p>	

The file syntax is described below. The character "^" represents the blank character. **Please note that** - due to requirements of the new RODOS-Lite user interface - **there may be changes between RODOS versions PV6.0 and PV6final in the degree of freedom in the character (non-numerical) part of line(s)**. See the format description part of the Table!

Leading line

- **Lines 1, 2, 3:** Obligatory comment lines ("---...---" / "AEROSOL FRACTIONS" / "---...---" in the Table).

Then, for the input of aerosol release fractions

- **Line 4:** Input mode flag for aerosol release fractions (variable IAEMOD)

IAEMOD = 1 0	Input of fractions follows below.
IAEMOD = 0 1	Input of fractions to be taken from RODOS aerosol release fraction file described in Chapter 4.2.3.2.

- IF IAEMOD = 0 1 THEN
Line 5: Entry name in the RODOS aerosol release fraction library file (variable CAENAM).

- IF IAEMOD = 1 0 THEN

Lines 5+: Release fractions of the initial core inventory in per cent for each (!) release group belonging to the release block specified in the source term file header with variable CRLGID. For each of the release groups two lines: One comment line with arbitrary text, and one line with the release fraction.

The release fractions ANTAER are given EXACTLY for the number, sequence, and type of release groups belonging to the release block specified in the source term file header, INCLUDING noble gases, iodine, and eventual other non-aerosol groups. The input for the non - aerosol-groups is needed for synchronisation of the aerosol fractions within the release block, but the data are ignored.

Example for the two input modes of aerosol release fractions

The input example with IAEMOD = 0 1 in Table 4 reads:

"Take release fractions from RODOS aerosol release fraction input file, entry name CAENAM =DRS-A_FK2."

In the aerosol fraction file shown in Table 19, you can find the data associated with entry DRS-A_FK2. Using mode IAEMOD = 1 0 for direct input of the aerosol fractions, the same specification as above can be realised directly in the user source term file:

```
IAEMOD= 1 0
#      Noble_gases
1.00000E+00
#      Iodine
4.07000E-01
#      Alkaline_metal
2.90000E-01
# Tellurium/Antim.
1.90000E-01
#      Alkaline_earth
3.20000E-02
#      RutheniumGrp.
1.70000E-02
#      Lanthanides
2.60000E-03
```

- After the aerosol release fraction section, there follow **three** obligatory comment lines ("----...----" / "ACTIVITY RELEASE" / "----...----" in the Table).
- Then follows **one** obligatory comment line ("# ABQ3IN [Bq] =" in the Table).

After that, for the input of the released activity, with each separate item preceded by one obligatory comment line

- Either

(F3) Released activity [Bq] for the sum of near range nuclides in the group "noble gases", for I-131, and for the sum of near range nuclides in the group "total aerosol",

or

(F4) Released activity [Bq] for the sum of near range nuclides in the release groups "noble gases", "iodine", and "total aerosol",

for the NQTUIT release time intervals defined in the file header section, five data entries per line.

4.2.3.1.5 Mode dependent File Section - Modes F2, F6, F7

Table 5 shows the input mode dependent source term data for input modes F6/F7, and the respective formats. The file structure is identical for input modes F2, F6 and F7 (the first line in the Table "ARISIN [Bq] or ARATIN [Bq/s] =" is a comment line without functional relevance¹⁴). The data follow immediately after the file header section described in Chapter 4.2.3.1.2. Explanations follow after the table!

The table below contains for the eight time intervals and the release groups of release block "DRS-A_7_GROUPS" from the example file header of Table 2 the mode-dependent data input for source term input mode F6 ↔ Activity [Bq] released for individual nuclides.

Table 5: User Source term file: Mode dependent file section - modes F2, F6, and F7

format	file contents (example: input mode F2 or F6 with "ARISIN")
F-01+	#ARISIN= (F7: "#ARATIN=")
F-09	# Kr- 88
F-08	1.22717E+17 1.08619E+17 2.35691E+16 1.85435E+16 1.23099E+16
F-08	8.17182E+15 7.23303E+15 3.37888E+15
F-09	# Rb- 88
F-08	5.48050E+12 4.45793E+12 5.34624E+14 3.75915E+14 2.50177E+14
F-08	1.58710E+14 1.30831E+14 4.62745E+13
F-09	# Sr- 89
F-08	2.11785E+12 2.11724E+12 8.23837E+14 6.96893E+14 5.70022E+14
F-08	4.43224E+14 4.43097E+14 1.89844E+14
F-09	# Sr- 90
F-08	1.22099E+11 1.22098E+11 4.76178E+13 4.02919E+13 3.29661E+13
F-08	2.56402E+13 2.56402E+13 1.09886E+13
F-09	# Zr- 95
F-08	2.88347E+11 2.88281E+11 1.12227E+14 9.49398E+13 7.70852E+13
F-08	6.03889E+13 6.03753E+13 2.58693E+13
F-09	# Te-132
F-08	1.65111E+13 1.64362E+13 6.18138E+15 5.20670E+15 4.24072E+15
F-08	3.28339E+15 3.26851E+15 1.39444E+15
F-09	# I -131
F-08	1.79091E+13 1.78811E+13 6.88584E+15 5.81727E+15 4.75205E+15
F-08	3.69018E+15 3.68432E+15 1.57649E+15
F-09	# I -132
F-08	2.58761E+13 2.57744E+13 9.71900E+15 8.18857E+15 6.67136E+15
F-08	5.16698E+15 5.14524E+15 2.19584E+15
F-09	# I -133
F-08	3.28951E+13 3.23515E+13 1.10425E+16 9.18928E+15 7.39427E+15
F-08	5.65607E+15 5.56261E+15 2.34458E+15
F-09	# I -135
F-08	2.26060E+13 2.14512E+13 5.49984E+15 4.41599E+15 3.42853E+15
F-08	2.53042E+15 2.40116E+15 9.76503E+14
F-09	# Xe-133

¹⁴ It is variable IRLTYP in the file header section that tells STERM/USER how to interpret the data!

F-08	9.44794E+17	9.44759E+17	5.44991E+17	4.84559E+17	3.63603E+17
F-08	2.72897E+17	2.73138E+17	1.44311E+17		
F-09	# Xe-135				
F-08	3.43787E+17	3.54450E+17	2.37957E+17	2.13285E+17	1.61574E+17
F-08	1.22503E+17	1.23848E+17	6.61004E+16		
F-09	# Cs-134				
F-08	1.75471E+12	1.75468E+12	6.84220E+14	5.78945E+14	4.73673E+14
F-08	3.68406E+14	3.68399E+14	1.57882E+14		
F-09	# Cs-137				
F-08	1.49498E+12	1.49498E+12	5.83037E+14	4.93338E+14	4.03640E+14
F-08	3.13942E+14	3.13941E+14	1.34546E+14		
F-09	# Ba-140				
F-08	3.63335E+12	3.62924E+12	1.40264E+15	1.18550E+15	9.68859E+14
F-08	7.52704E+14	7.51852E+14	3.21857E+14		

Formats (Character "^" represents the blank character, "d" some digit 0...9)
 F-01 : 81 arbitrary characters
F-01+: Line with fixed content - type exactly as shown! (NEW IN RODOS PV6FINAL)
 F-08 : 5*(floating point number with 12 digits and trailing blank "{sign}d.dddddE+dd^")
 F-09 : 2 arbitrary characters, 7 characters containing the nuclide name; names must comply with the naming conventions described in [4]).

The file syntax is described below. The character "^" represents the blank character. **Please note that** - due to requirements of the new RODOS-Lite user interface - **there may be changes between RODOS versions PV6.0 and PV6final in the degree of freedom in the character (non-numerical) part of line(s).** See the format description part of the Table!

Leading line

- **Line 1:** Obligatory comment line.

Then, for an arbitrary number of nuclides in any order:

- **Nuclide name:** Names should comply with the naming conventions described in [4]¹⁵.
- Either
F2/F6: Activity released for the corresponding nuclide, [Bq],
 or
FT7: Activity release rate for the corresponding nuclide, [Bq/s],
 for the NQTUIT release time intervals defined in the file header section, five data entries per line.

4.2.3.1.6 How to Upgrade Version PV5 User Source Term Files for RODOS PV6

In RODOS PV6.0 source input term files there are only three changes in comparison to RODOS PV5, and all of them concern the file header section.

(Change 1) The line "#ISTFIL= 0 1" that appeared in PV5 files before the line "#CQTORI=ORIGINAL" in Table 2 does no longer exist.

(Changes 2 and 3) There are two additional variables and corresponding values VOLFIN and VENTIN (after the values for QHIN, see Table 2).

¹⁵ STERM/USER identifies nuclides by their names and ignores any entry not of interest or not recognisable. If a nuclide name does not comply with the naming conventions of [4], it cannot be recognised and will always be skipped when the file is read.

4.2.3.1.7 How to Upgrade Version PV4 User Source Term Files for RODOS PV5

Follow the recipe below! The description is oriented on the old PV4 file structure.

1. Rename the file (if necessary) according to the new naming rules explained in Chapter 4.2.3.1.1.
2. Replace the line containing old variable CSTFIL with the line "CQTORI=ORIGINAL".
3. Swap the old lines with "IRLGRP" and "CRLGID".
4. ONLY FOR INPUT MODES F6 AND F7: Even for input modes F6 and F7, you must specify a valid release block name with variable CRLGID, and also the correct value for IRLGRP for the respective CRLGID (**since version PV5.0**).
5. After the line "BEGFRE =" insert the following lines:

#	NUMBER OF INPUT RELEASE TIME INTERVALS				
NQTUIT=24					
#	LOWER EDGES OF INPUT RELEASE TIME INTERVALS [h]				
#QTUIT1=					
	0.00	0.50	1.00	1.50	2.00
	2.50	3.00	3.50	4.00	4.50
	5.00	5.50	6.00	6.50	7.00
	7.50	8.00	8.50	9.00	9.50
	10.00	10.50	11.00	11.50	
#	UPPER EDGES OF INPUT RELEASE TIME INTERVALS [h]				
#QTUIT2=					
	0.50	1.00	1.50	2.00	2.50
	3.00	3.50	4.00	4.50	5.00
	5.50	6.00	6.50	7.00	7.50
	8.00	8.50	9.00	9.50	10.00
	10.50	11.00	11.50	12.00	

This means that your file contains data for "24 input release time intervals for a release duration of 12 hours in consecutive half hour time steps", which is exactly the timing in all PV4 source term files! **ATTENTION (since version PV5.0)**: If the file contains time intervals describing a "zero release", you must *either* mark such intervals for skipping by setting QTUIT1=QTUIT2, *or* you must completely reformat the file according to the new rules (see also Chapter 4.2.3.1.2).

6. Insert the three new comment lines between the three blocks of iodine fractions (the result must look like in Table 2).

THIS WAS ALL! The upgraded file header is now in accordance with the PV5.0 rules, and the activity release data section needs no modification.

If you compare a file upgraded by carrying out the above steps with the respective tables from this chapter, you may notice that the names of almost all variables have changed (for instance, former variable ANTJOD is now called ANTJIN, former HFS is now HFSIN, and so on). However, the lines containing the names in the files are only comment lines to make the file more structured for humans. STERM/UINPUT reads these lines without interpreting the text. That means (for your convenience) that you do not need to change the variable names in the already existing PV4 files. However, for the sake of clarity, you should use the new PV5.0 - names when creating new files from scratch for version PV5.0.

4.2.3.2 Aerosol Release Fraction File <in.aerosolfractions>

For source term input modes F3 and F4 of Table 1 STERM uses release fractions for aerosols to account for the contributions of different aerosol nuclide groups to the total aerosol release. Optionally, the aerosol release fractions can be read from file <in.aerosolfractions> that is provided with the RODOS system.

4.2.3.2.1 File Name and File Location

The aerosol release fraction input has the fixed name *in.aerosolfractions* and resides in directory <~rodos/roextern/data/sourceterm/setup/>.

4.2.3.2.2 File Description

Table 6 shows a part of file <in.aerosolfractions> together with the respective formats. The file structure and content is SINCE RODOS PV6.0. Explanations can be found in the comment section of the file.

Table 6: Aerosol fraction file <in.aerosolfractions>

format	file contents (example)
F-01	# =====
F-02	# NCOMM = 19 ***** number of comment lines to follow *****
F-01	#*****
F-01	#
F-01	VERSION 08.05.2004
F-01	#*****
F-01	#
F-01	# The fractions <ANTAER> must be given EXACTLY for the number, sequence,
F-01	# and type of release groups belonging to the respective release block,
F-01	# INCLUDING Noble_gases, iodine, and eventual other non-aerosol groups.
F-01	# The input for the non - aerosol groups is needed for synchronisation
F-01	# of the aerosol fractions within the release block, but the data are
F-01	# ignored.
F-01	#
F-01	# Names for the groups may be written to the right of the fractions,
F-01	# but they are only for making the file better readable.
F-01	#
F-01	# CAENAM : Name of aerosol-group entry to follow (max. 16 characters).
F-01	# NSKIP : Number of lines to follow in the aerosol-group entry
F-01	# ANTAER : Release fractions of the initial inventory in per cent.
F-01	#
F-01	# =====
F-01	#-----1234567890123456<-----
F-03	CAENAM =DRS-A_FK1
F-04	NSKIP = 8
F-01	ANTAER =
F-05	1.00000E+02 Noble_gases
F-05	7.97000E+01 Iodine
F-05	5.00000E+01 Alkaline_metal
F-05	3.50000E+01 Tellurium/Antim.
F-05	6.70000E+00 Alkaline_earth
F-05	3.80000E+01 RutheniumGrp.
F-05	2.60000E-01 Lanthanides
F-01	#-----1234567890123456<-----
F-03	CAENAM =DRS-A_FK2
F-04	NSKIP = 8
F-01	ANTAER =
F-05	1.00000E+02 Noble_gases
F-05	4.07000E+01 Iodine
F-01	2.90000E+01 Alkaline_metal
F-05	1.90000E+01 Tellurium/Antim.
F-05	3.20000E+00 Alkaline_earth
F-05	1.70000E+00 RutheniumGrp.
F-05	2.60000E-01 Lanthanides
.....and so on.....

Formats (Character "^" represents the blank character, "d" some digit 0...9)

F-01 : 72 arbitrary characters.
 F-02 : 9 arbitrary characters, right-adjusted integer with 4 digits.
 F-03 : 8 arbitrary characters, 16 characters (aerosol group entry name).
 F-04 : 8 arbitrary characters, right-adjusted integer with 3 digits.
 F-08 : 1 floating point number with 8 digits d.ddE+dd), followed by 1 blank character, followed by voluntary comment.

4.2.3.2.3 File Errors and File Maintenance

File <in.aerosolfractions> is used by STerm at runtime of RODOS and is checked when it is read in. Format errors, structure inconsistencies, values out of range, or requested but not found aerosol fraction entries are reported, and program execution stops.

Maintenance of file <in.aerosolfractions> is matter of the RODOS team.

4.3 Module description

STERM/UINPUT consists of the FORTRAN 77 subroutines listed in Table 7. by means of the corresponding statements in the STERM comment library, member "Programs". The steering routine, UINPUT_STEER, is called from the embedding ASY program.

Note: In RODOS PV6final, the Native Language Support (NLS) via external files was introduced for the source term code, and former ".ff"-files became normal "FORTRAN .f"-files.

Table 7: Subroutine inventory of STERM/UINPUT

C	=====	C
C	PACKAGE -> UINPUT of STERM/USER	C
C	FILE -> [setqtr.uinput.f]	C
C	TASKS -> GET - PRINT - CHECK OF USER SOURCE TERM INPUT	C
C	=====	C
C	UINPUT_STEER -> STEERING ROUTINE	C
C	-----	C
C	UINPUT_FILEIN -> READING OF START-UP STERM FROM FILE	C
C	-----	C
C	UINPUT_STAINP -> STATIC SOURCE TERM INPUT	C
C	(WITHOUT CALL TO DYNAMIC WINDOWS)	C
C	UINPUT_DYNINP -> INTERACTIVE SOURCE TERM INPUT	C
C	(WITH CALL TO DYNAMIC WINDOWS)	C
C	-----	C
C	UINPUT_BEGWIN -> INIT-WINDOW(S) FOR START-UP OF STERM INPUT	C
C	UINPUT_COMWIN -> INIT-WINDOW(S) FOR STERM COMMENT	C
C	UINPUT_FINLAS -> FILE CHECK FOR WINDOW-BRANCH "Last applied ..."	C
C	UINPUT_FINWIN -> INIT-WINDOW(S) FOR STERM INPUT FILE NAME	C
C	UINPUT_FOUWIN -> INIT-WINDOW(S) FOR STERM OUTPUT FILE NAME	C
C	UINPUT_LOOWIN -> INIT-WINDOW(S) FOR STERM INPUT LOOP	C
C	-----	C
C	UINPUT_REDWIN -> INIT-WINDOW(S) FOR STERM INPUT WITH REDUCED	C
C	INPUT MASK (LMOD08)	C
C	-----	C
C	UINPUT_HEAWIN -> INIT-WINDOW(S) FOR HEADER SECTION OF STERM	C
C	UINPUT_MODWIN -> INIT-WINDOW(S) FOR USER INPUT MODE	C
C	UINPUT_GRPWIN -> INIT-WINDOW(S) FOR STERM RELEASE GROUPS	C
C	UINPUT_DATWIN -> INIT-WINDOW(S) FOR MODE DEPENDENT STERM DATA	C
C	UINPUT_DATW26 -> INIT-WINDOW(S) FOR MODE 2 AND 6 DATA	C
C	UINPUT_DATW07 -> INIT-WINDOW(S) FOR MODE 7 DATA	C


```

C      UINPUT_AEFWIN -> INIT-WINDOW(S) FOR AEROSOL FRACTIONS          C
C      ----- C
C      UINPUT_HEADCH -> CHECK HEADER SECTION OF USER SOURCE TERM      C
C      UINPUT_HEADPR -> PRINT HEADER SECTION OF USER SOURCE TERM      C
C      UINPUT_LMOD## -> GET-PRINT-CHECK MODE DEPENDENT USER SOURCE TERM C
C      (## FOR DIFFERENT QTMODES)                                     C
C      ----- C
C      ICHECK_FXINPT -> CHECK IF FX FILE INPUT IS COMPATIBLE WITH      C
C      INVENTORY AND NEAR RANGE NUCLIDES IN THIS RUN                  C
C      ----- C
C      IMESSG_NOREL  -> ERROR MESSAGE FOR ZERO STERM IN GOOD INTERVAL  C
C      ----- C
C      ===== C

```

For source term input via the dynamic initialisation windows, program flow is complicated because the source term input has many branching possibilities that determine different courses of action. As can be seen from Table 7, most of the subroutines are indeed devoted to the control of and interaction with the input windows.

UINPUT_STATIN manages the proceeding in case of source term input without interaction with the user, that is, for the web-based user interface, for debugging, and in the stand-alone operation of STERM/UINPUT.

UINPUT_FILEIN manages the different steps in reading the file with the start-up source term, if there is any.

The remaining two subroutines have auxiliary character.

4.4 Description of Shared Memory

The SHM data classes and parameters required for proper operation of STERM/UINPUT are displayed in Table 8 by means of the coding in the RODOS define file of a RODOS program containing STERM/UINPUT. The list constitutes only a fraction of the entire program shared memory. Data classes not relevant for STERM/UINPUT are not shown at all. Lines with dots indicate that the data class contains more variables not relevant for STERM/UINPUT.

Corresponding assign, load, and archive instructions for the variables shown in Table 8 can be located in the assign, load, and archive files in RODOS directory <~rodos/rodecfiles/include> of the RODOS program STERM/UINPUT is contained in.

Table 8: Shared memory of STERM/UINPUT with support software

```

#####
# Site Information (for printout and support task SETINV) #
#####

CANLAG:
  BlockInve [1:32] Char (0:0) ---
  'NPP-Block Inventar';
  BlockType [1:32] Char (0:0) ---
  'NPP-Block type';
END CLASS

```

```

CSITES:
  BlockName [1:32] Char (0:0) ---
    'NPP-block name';
  SiteName [1:32] Char (0:0) ---
    'NPP name';
  .....
  .....
END CLASS

DANFIX:
  ANLBHE [] Real (0:5000) m
    'Building height';
  ANLBWI [] Real (0:5000) m
    'Building width';
  ANLKAM [] Real (0:5000) m
    'Stack height of NPP-block';
  ANLSVO [] Real (-2:100) m**2
    'Area of stack vent opening [m**2]
    (-1 <=> Take default value)';
  ANLVOL [] Real (-2:10000) m**3/s
    'Average volume flux through stack vent opening [m**3/s]
    (-1 <=> Take default value)';
  BlockLat [] Real (-90.0:90.0) Deg(decimal)
    'geogr. Latitude of NPP-block';
  BlockLong [] Real (-180.0:180.0) Deg(decimal)
    'geogr. Longitude of NPP-block';
  IANLEI [] Integer (0:0) MW
    'Thermal power of NPP-block';
  IANTGE [] Integer (0:0) Days
    'Burn-up of NPP-block';
  .....
  .....
END CLASS

```

```

#####
# Nuclide Information (from support task SETNUK) #
#####

```

```

CNUKPD:
  CNUDFL [1:7][1:25] Char (0:0) ---
    'Default near range nuclides';
  CNUKLN [1:7][1:25] Char (0:0) ---
    'Names of near range nuclides';
  CRHALFX [1:1][1:25] Char (0:0) ---
    'Phys. unit for RHALFX in DNUKPD';
  ZzCNUKPD [1:7] Char (0:0) ---
    'Alignment-Correction';
END CLASS

DNUKPD:
  INUDFL [1:2] Integer (0:0) ---
    'Run with default (1 0) or user (0 1) near range nuclides';
  NAERO [] Integer (0:0) ---
    'Number of aerosol nuclides in NNUKLN';
  NIODS [] Integer (0:0) ---
    'Number of iodine nuclides in NNUKLN';
  NNOBL [] Integer (0:0) ---
    'Number of noble gas nuclides in NNUKLN';
  NNUKLN [] Integer (0:0) ---
    'Total number of near range nuclides';
  NSPCL [] Integer (0:0) ---
    'Number of special nuclides in NNUKLN';
  RADEC [1:25][1:5] Real (0:0) 1/(s,m,h,d,a)
    'Radioactive decay constants (ln2/rad.half-life)';
  RHALF [1:25][1:5] Real (0:0) (s,m,h,d,a)
    'Radioactive half-lives';
  RHALFX [1:25] Real (0:0) CRHALFX

```

```

        'Radioactive half-lives (nice format for printout)';
END CLASS

CNUKGO:
    CNUGRF [1:8][1:25] Char (0:0) ---
    'Near range nuclides names for graphics';
END CLASS

CNUKSD:
    CISOTOP [1:7][1:100] Char (0:0) ---
    'Nuclide names (FixData/StammDaten nuclides)';
    CISOTYP [1:4][1:100] Char (0:0) ---
    'Deposition type (noble gas/iodine/aerosol/special)';
    CRHLISO [1:1][1:100] Char (0:0) ---
    'Phys. unit for RHLISO in DNUKSD';
END CLASS

DNUKSD:
    ISONUK [1:100] Integer (0:0) ---
    'Cross-reference table between FixData and calculation
    nuclides';
    ISOYES [1:100] Integer (0:0) ---
    'Flag to select near range nuclides from FixData nuclides';
    NISOT [] Integer (0:0) ---
    'Total number of nuclides (FixData/StammDaten)';
    RHLISO [1:100] Real (0:0) CRHLISO
    'Radioactive half-lives';
    ZDNUKS [] Real (0:0) ---
    'Alignment-Correction';
END CLASS

#####
# User Source Term Input #
#####

CQTRMX:
    COMFR1 [1:80] Char (0:0) ---
    'Comment on the source term data (line 1)';
    COMFR2 [1:80] Char (0:0) ---
    'Comment on the source term data (line 2)';
    CQTFIL [1:32] Char (0:0) ---
    'Last applied user source term file name';
END CLASS

CSTRMX:
    CAENAM [1:16] Char (0:0) ---
    'Release group specifier for aerosol fractions';
    CRLGID [1:16] Char (0:0) ---
    'Release group specifier';
    CQTFIN [1:32] Char (0:0) ---
    'Source term input file name';
    CQTORI [1:8] Char (0:0) ---
    'Comment on origin type of source term input';
END CLASS

DSTRM0:
    A131IN [1:24] Real (0:0) ---
    'Activity released for I-131 for 24 time slots';
    ANTJIN [1:24][1:3] Real (0:0) ---
    'Percent of elem/org.aerosol iodine (sum=100%) for 24 time
    slots';
    ANTAER [1:16] Real (0:0) ---
    'Relative aerosol fractions (sum=1)';
    ABQ3IN [1:24][1:3] Real (0:0) ---
    'Activity released for NobleG-Iodine-Aerosol for 24 time
    slots';
    ABQGIN [1:24][1:16] Real (0:0) ---

```

```

    'Activity released for 16 release groups for 24 time slots';
ARATIN [1:24][1:100] Real (0:0) ---
    'Activity release rate for 100 nuclides for 24 time slots';
ARISIN [1:24][1:100] Real (0:0) ---
    'Activity released for 100 nuclides for 24 time slots';
BEGFRE [] Real (0:0) h
    'Begin of initial release relative to end of chain reaction';
HFSIN [1:24] Real (0:0) ---
    'Release height for 24 time slots';
IAEMOD [1:2] Integer (0:0) ---
    'Aerosol fraction input via shared memory (1,0) or from file
(0,1)';
IQTDYN [] Integer (0:0) ---
    'CALL TO DYN. STERM WINDOWS (0 NO, ELSE YES)';
IQTSRC [1:5] Integer (0:0) ---
    'FLAG FOR INITIAL SOURCE TERM INPUT';
IRLGRP [] Integer (0:0) ---
    'Number of release groups';
IRLTYP [1:12] Integer (0:0) ---
    'Indicator flag for source term input type';
NQTUIT [] Integer (0:0) ---
    'Number of time slots used in user input of source term';
PROZIN [1:24][1:16] Real (0:0) ---
    'Activity release fractions for 16 release groups for 24 time
slots';
QHIN [1:24] Real (0:0) ---
    'Thermal power released for 24 time slots';
QTUIT1 [1:24] Real (0:0) ---
    'Lower time interval boundaries of time slots [hours after
BEGFRE]';
QTUIT2 [1:24] Real (0:0) ---
    'Upper time interval boundaries of time slots [hours after
BEGFRE]';
VENTIN [1:24] Real (0:10000) m**2
    'User input: Vent area of release to the atmosphere [m**2]';
VOLFIN [1:24] Real (0:10000) m**3/s
    'User input: Vertically released volume flux [m**3/s]';
ZDSTRM0 [] Real (0:0) ---
    'Alignment-Correction';
END CLASS

CTOGGLS:
    CCCTOG [1:32][1:256] Char (0:0) ---
    'Buffer array for interaction with dyn. STerm windows';
END CLASS

DTOGGLS:
    IIITOG [1:256] Integer (0:0) ---
    'Buffer array for interaction with dyn. STerm windows';
    RRRTOG [1:256] Real (0:0) ---
    'Buffer array for interaction with dyn. STerm windows';
END CLASS

```

5 Program Maintenance and Test Plan for STERM/UINPUT

5.1 Operating Environment

5.1.1 Program Language

STERM/UINPUT is written in FORTRAN 77 with the following extension to the Language Standard:

- FORTRAN INCLUDE statements.
- Statement IMPLICIT NONE.
- Names longer than 6 characters.
- Lower case characters.

5.1.2 Programming Conventions

- In all subroutines, use IMPLICIT NONE to force declaration of variables.
- Do not use lower case characters unless unavoidable due to references to external pieces of code.

5.1.3 Support Software

Table 9: Support software for STERM/UINPUT

- Code package SETCHN for decay chain information.
- Code package SETRLG for release group information.
- Program package SETNUK [4] for FixData and near range nuclide information.
- Code package SETQTR_FILEIO¹⁶ for subroutines for read/write of STERM files.
- Code package SETQTR_QPRINT¹⁶ for STERM printout subroutines.
- Code package UTILTY¹⁶ for general auxiliary subroutines.
- C-procedure win_request of the RODOS system.

5.2 Error Correction Procedures

On input errors detected during the interactive phase, STERM/UINPUT flashes up a window explaining the problem(s), and then goes back to the window that caused the problem(s). In that way, the user is forced to correct all detectable errors on-line.

¹⁶ For the subroutine inventory of this package see the STERM comment library, member "Programs".

On input errors detected during the passive checking phase for all source term input without the dynamic initialisation windows, STERM/UINPUT flashes a message indicating the error condition on the RODOS screen, writes a text explaining the problem(s) to the RODOS run protocol, and terminates.

In rare cases, a fatal error is caused by some mistake made when RODOS was installed¹⁷. Such errors are termed installation errors. When STERM/UINFO encounters such an error, it flashes a message on the RODOS screen to tell the operator about the error exit, and where to get more information about what is going wrong.

5.3 Verification Procedures

- none -

5.4 Test Plan and Test Analysis Report

5.4.1 Tests with a Stand-alone Version of the Software

The stand-alone version of STERM/UINPUT operates by bypassing the interaction with the dynamic initialisation windows, in the same way as for RODOS user input with the web-based user interface, which means that only the general and auxiliary subroutines can be checked this way.

After major program changes, the general performance of STERM/UINPUT - excluding its window interaction part - is checked with the test source terms <F1.Repro_FZK-Sep00> and <FX.Repro_FZK-Sep00> for all source term input modes, and inventory <INVE.PWR_3733MWth_Leitfaden95>.

5.4.2 Tests with the Integrated Software within the RODOS system

STERM/UINPUT is part of the standard test procedures of the RODOS system.

5.5 Error report

Since the first issue with RODOS PV6.0 in 2004, no errors were detected at FZK or reported by external users.

¹⁷ Missing entries in the FixData base, or wrong entries in STERM auxiliary files provided with the RODOS installation can cause errors of this type for example.

6 Operation Manual for STERM/UMAKEQ

6.1 Program Description

The main task of STERM/UMAKEQ is to derive the Near_ADM source term from the user source term.

Within RODOS, STERM/UMAKEQ is "invisible" to the operator of the system (at least, as long as it is performing correctly).

6.2 Description of Operation Modes and RODOS User Interfaces

Program package STERM/UMAKEQ is part of the Analysing Subsystem of RODOS. In the diagnosis mode, it belongs to program RMC3DIAG. In the automatic prognosis mode, it is not used. In the interactive prognosis mode it belongs to each Near_ADM.

STERM/UMAKEQ does not distinguish between operation modes, nor between RODOS A or B Users.

6.3 Connections to other Software

STERM/UMAKEQ gets input from

- The RODOS shared memory (default settings).
- The embedding external program of ASY (steering parameters, site information etc.).
- Several STERM support tasks.
- Program package STERM/UINPUT (File with last applied user source term and name of that file).

STERM/UMAKEQ produces output for

- The RODOS graphics (tables with source term and inventory information).
- Program package STERM/UTOADM (File with entire Near_ADM source term and name of that file).

6.4 Help Functions

- none -

6.5 Description of Runs

- none (as STERM/UMAKEQ is invisible for the operator) -

7 Functional Specification and Data Specification Documents for STERM/UMAKEQ

7.1 Software Organisation

7.1.1 Overview

Figure 2 illustrates the input from STERM support tasks and from file for STERM/UMAKEQ, and the output from STERM/UMAKEQ.

As can be seen from the figure, STERM/UMAKEQ needs support from SETNUK about the RODOS nuclides and the near range nuclides, from SETCHN about the STERM nuclides, from SETRLG about the nuclides release groups to be considered for the run, and from SETINV about the radionuclide inventory.

To start with, STERM/UMAKEQ reads the file containing the "approved user source term"¹⁸ from the user source term library. After that, it calculates the complete Near_ADM source term from the beginning of the release to its end in the time steps of fixed duration determined by the target Near_ADM (for a description of the physical model behind the calculation see [2]).

The resulting Near_ADM-source term is kept a file with name `<FU.STERMout>` in directory `<~rodos/roextern/outall/user_ID/run_ID/>`. This is the file program package STERM/UTOADM will use later; the file format is described in [9]. Because STERM/UTOADM works with files from different sources with different names, the file name is contained in the RODOS shared memory¹⁹ (see Figure 2).

STERM/UMAKEQ also generates several other files:

- A file with name `<user_ID.FX.MyLastSTerm>` or `<user_ID.FX.MeinLetzterQTerm>` in the user source term library, dependent on the language version of the STERM/UINPUT RODOS executable. This file contains the user source term in the original input mode, plus data for all modes from Table 1 that are derivable from the original input mode. It can be re-used as a start-up source term in another RODOS run, thereby employing the possibilities to switch between input modes, see [5], Chapter on "Source Term Initialisation Windows". The file format is different to all other STERM files and not described in the report.
- A "Source Term Exchange File" with the fixed name `<F6.STERMout>` for the export RODOS of source terms to non-RODOS programs. Such a file has an "extended F6 format", consisting of a standard F6 file header and a standard F6 data part, and of a section with additional

¹⁸ The file name will be `<user_ID.FX.MyLastSTerm>` or `<user_ID.FX.MeinLetzterQTerm>`, see also Chapter 4.1).

¹⁹ Shared Memory variable CQTNear_ADM.

information for RODOS-external programs in between the two standard parts; for a file description see [10].

- An work file with the name `<FY.STERMout>` in directory `<~rodos/roextern/outall/user_ID/run_ID/>` (not shown in Figure 2). The file format is different to all other STERM files and not described in this report.

Apart from that, STERM/UMAKEQ prints tables with derived source term information to file `<Prognosis|Diagnosis.STermMoreInfo>` in directory `<~rodos/roextern/outall/user_ID/run_ID/>`. This file is accessible with the Graphics Manager of the RODOS system, see [5], Chapter on "Prognosis Results Accessible with the RODOS Graphics".

7.1.2 Program and Module Inventory

STERM/UMAKEQ is imbedded in a module of an external program of ASY, and not operated as a program or module on its own in RODOS.

7.1.3 File Inventory

Input: File with the "last applied user source term". For the file name and directory see Chapter 7.1.1. The file size depends on the source term input mode; it is typically about 0.5 kilobytes²⁰ to some 10 kilobytes.

Output: File `<FU.STERMout>` with the complete Near_ADM source term. For the file directory see Chapter 7.1.1. The file size depends on the source term input mode; it is typically around 10 kbytes to several 10 kbytes. For the other output files see Chapter 7.1.1; their sizes are in the same order of magnitude as for file `<FU.STERMout>`.

7.2 Data and Interface Description

7.2.1 Input Data

- The SHM section with the "Nuclide Information" (see Table 8).
- The COMMON blocks filled by the support tasks SETINV (see Chapter 9.5), SETCHN (see Chapter 10.5) and SETRLG (see Chapter 11.5).
- SHM variable CQTFIL of data class CQTRMX telling the name of the "last applied user source term".
- The file with the "last applied user source term".
- Steering parameters in the argument list of the steering routine.

STERM/UMAKEQ reads the file with the "last applied user source term", thereby filling with values the SHM variables of data classes CQTRMX, CSTRMX, DSTRM0, and DSTRMX (see Chapter 7.4) that refer to the user input source.

²⁰ 1 kilobyte = 1024 bytes.

7.2.2 Output Data

- SHM variable CQTNear_ADM of data class CNear_ADMQT telling name of the current internal file with the "complete Near_ADM source term".
- The file containing the "complete Near_ADM source term".²¹

7.2.3 Data Files

File <FU.STERMout> is described in [9].

7.3 Module description

STERM/UMAKEQ consists of the FORTRAN 77 subroutines listed in Table 10 by means of the corresponding statements in the STERM comment library, member "Programs". The steering routine, UMAKEQ_STEER, is called from the embedding ASY program.

Note: In RODOS PV6final, the Native Language Support (NLS) via external files was introduced for the source term code, and former ".ff"-files became normal "FORTRAN .f"-files.

Table 10: Subroutine inventory of STERM/UMAKEQ

C	=====	C
C	PACKAGE -> UMAKEQ of STERM/USER	C
C	FILE -> [setqtr.umakeq.f]	C
C	TASKS -> GET - PRINT - CHECK OF USER SOURCE TERM INPUT	C
C	=>	C
C	1. MAKE ADM SOURCE TERM FROM USER S.TERM INPUT AND WRITE THAT	C
C	TO THE FU - FILE.	C
C	2. MAKE USER S.TERM INPUT FOR ALL SOURCE TERM MODES POSSIBLE	C
C	WITH ORIGINAL MODE AND WRITE THAT TO THE FX-FORMAT-FILE.	C
C	CARRY OUT BOTH TASKS FOR THE TOTAL TIME SPAN COVERED BY	C
C	THE USER SOURCE TERM INPUT TIMES.	C
C	~~~~~	C
C	UMAKEQ_STEER -> STEERING SUBROUTINE	C
C	-----	C
C	UMAKEQ_UINPUT -> SETUP OF USER SOURCE TERM INPUT,	C
C	RELEASE GROUPS AND DECAY CHAINS,	C
C	INVENTORY (IF USED IN RUN)	C
C	NO CHECKS, NO PRINTOUT OF USER S.TERM INPUT.	C
C	ALL THIS WAS DONE IN [SETQTR_UINPUT]!	C
C	UMAKEQ_QTUINF -> SET USER_SOURCE_TERM-YES-NO-IN-TIME-STEP FLAG	C
C	FOR SUBROUTINE UMAPXR_1D_NONULL	C
C	-----	C
C	UMAKEQ_HEAMAP -> FOR HEADER VARIABLES HFS-QH-VOFL-VOAR-ANTJOD:	C
C	MAPPING OF USER SOURCE TERM INPUT ON	C
C	ADM SOURCE TERM SIDE.	C
C	UMAKEQ_HEAPRN -> ADJUST MIN. VALUES AND PRINT "DAILY" RELEASE	C
C	-HEIGHT, -TH.POWER, -VOL.FLUX, -VOL.FLUX AREA,	C
C	-IODINE FRACTIONS	C
C	UMAKEQ_QINI## -> INITIALISE ADM SOURCE TERM CALCULATIONS	C
C	(## FOR DIFFERENT QTMODES)	C

²¹ The variables for the data in the Near_ADM source term are kept in internal STERM/UMAKEQ COMMON blocks. After termination of STERM/UMAKEQ they cease to exist, and are therefore not described here.

```

C   UMAKEQ_QCAL## -> CALCULATE ADM SOURCE TERM                               C
C                                     (## FOR DIFFERENT QTMODES)                C
C   UMAKEQ_CALBEG -> BEGIN OF ADM SOURCE TERM CALCULATION:                  C
C   GENERAL STARTUP PROCEDURES FOR MOST OF THE                             C
C   QINI##- ROUTINES                                                         C
C   UMAKEQ_CALEND -> END   OF ADM SOURCE TERM CALCULATION:                  C
C   SETUP OF INFORMATION WHICH IS STILL MISSING.                            C
C   -----                                                                    C
C   UMAKEQ_FXFILE -> MAKE TIME-MAPPED+CONDENSED MULTPL-LMODE FX FILE        C
C                                                                                                                       C
C =====                                                                    C

```

The recursive formalism for the Near_ADM source term generation is described in [2]. STERM/UMAKEQ can process input source terms without principle limitation of the release duration, because it constructs the Near_ADM source term in cyclic episodes called "calculation days". Each such "day" consists of 24 time steps of Near_ADM time interval duration. That means, that for Near_ADM time intervals with 10 minutes duration one "day" lasts 8 hours, with 30 minutes 12 hours, with 60 minutes 24 hours, and so on; the duration of the Near_ADM time steps is passed via the argument list of the steering routine. At the end of each "calculation day", the generated Near_ADM source term is appended to file <FU.STERMout>, all computation buffers and arrays are cleared, and all items relevant for the recursive algorithm described in [2] from the last time step of the current day are used as start-up values for the following day.

7.4 Description of Shared Memory

The SHM data classes and parameters required for proper operation of STERM/UMAKEQ are displayed in Table 11 by means of the coding in the RODOS define file of a RODOS program containing STERM/UMAKEQ. The list constitutes only a fraction of the entire program shared memory. Data classes not relevant for STERM/UMAKEQ are not shown at all. Lines with dots indicate that the data class contains more variables not relevant for STERM/UMAKEQ.

Corresponding assign, load, and archive instructions for the variables shown in Table 11 can be located in the assign, load, and archive files in RODOS directory <~rodos/rodecfiles/include> of the RODOS program STERM/UMAKEQ is contained in.

Table 11: Shared memory of STERM/UMAKEQ with support software

```

The Shared Memory from STERM/UINPUT in Table 8,
EXCEPT the data classes CTOGGLS and DTOGGLS,
and ADDITIONALLY

#####
# Near_ADM Source Term Input File Name
#
#####

CNear_ADMQT:
  CQTNear_ADM [1:32] Char (0:0) ---
  'Name of Source Term to Near_ADM file';

```

END CLASS

8 Program Maintenance and Test Plan for STERM/UMAKEQ

8.1 Operating Environment

8.1.1 Program Language

STERM/UMAKEQ is written in FORTRAN 77 with the following extension to the Language Standard:

- FORTRAN INCLUDE statements.
- Statement IMPLICIT NONE.
- Names longer than 6 characters.
- Lower case characters.

8.1.2 Programming Conventions

- In all subroutines, use IMPLICIT NONE to force declaration of variables.
- Do not use lower case characters unless unavoidable due to references to external pieces of code.

8.1.3 Support Software

Table 12: Support software for STERM/UMAKEQ

- Program package STERM/UINPUT for the STERM internal file with the approved user source term input, and the current name of that file.
- Code package SETINV for radionuclide inventory information.
- Code package SETCHN for decay chain information.
- Code package SETRLG for release group information.
- Program package SETNUK [4] for FixData and near range nuclide information.
- Code package SETQTR_FILEIO²² for subroutines for read/write of STERM files.
- Code package SETQTR_QPRINT²² for STERM printout subroutines.
- Code package SETQTR_SUBLIB²² for STERM internal auxiliary subroutines.
- Code package UMAPIT for mapping of source term items in different time representations on the user input side and the Near_ADM side [8].
- Code package UTILTY²² for general auxiliary subroutines.

²² For the subroutine inventory of this package see the STERM comment library, member "Programs".

8.2 Error Correction Procedures

In rare cases, a fatal error is caused by some mistake made when RODOS was installed²³. Such errors are termed installation errors. When STERM/UMAKEQ encounters such an error, it flashes a message on the RODOS screen to tell the operator about the error exit, and where to get more information about what is going wrong.

8.3 Verification Procedures

After major program changes, the radioactive decay formalism in STERM/UMAKEQ is checked routinely for selected nuclides and decay chains of the type $m \rightarrow n$ and $p \rightarrow m \rightarrow n$:

- By comparison of STERM/UMAKEQ results with corresponding results from earlier RODOS versions, with test source terms `<F1.Repro_FZK-Sep00>` and inventory `<INVE.PWR_3733MWth_Leitfaden95>`.
- By comparison of STERM/UMAKEQ results obtained with test source term `<F1.TestLeitfaden120h>` and RODOS radionuclide inventory file `<INVE.PWR_3733MWth_Leitfaden95>` with the inventory tables from [11], pp. 8-9.
- By hand, using the full analytic formulae (e.g. [12], Chapter 15) on selected STERM/UMAKEQ results.

8.4 Test Plan and Test Analysis Report

8.4.1 Tests with a Stand-alone Version of the Software

After major program changes, the performance of STERM/UMAKEQ is checked with test the source terms `<F1.Repro_FZK-Sep00>` and `<FX.Repro_FZK-Sep00>` for all source term input modes, and radionuclide inventory file `<INVE.PWR_3733MWth_Leitfaden95>`.

8.4.2 Tests with the Integrated Software within the RODOS System

STERM/UMAKEQ is part of the standard test procedures of the RODOS system.

8.5 Error report

Since the first issue with RODOS PV5.0 in 2003, one error was detected in March 2004 (in the calculation part of source term input modes F3/F4) that was corrected in RODOS PV6.0.

²³ Missing entries in the FixData base or wrong entries in STERM auxiliary files provided with the RODOS installation can cause errors of this type for example.

9 APPENDIX 1: STERM Support Task SETINV for Setting Up a Radionuclide Inventory

9.1 Task

Source term input modes F1 to F5 of Table 1 require a radionuclide inventory of the reactor considered for generating the nuclide-specific information for the Near_ADMS. RODOS contains for each nuclear power plant installed in its FixData base a reference to a radionuclide inventory file that will become attached at runtime for the site under consideration. Code package SETINV ("SETUp of INVENTORY") reads such a file, analyses the data, and edits them for disposal in STERM.

Structure, format and content of radionuclide inventory files are explained in Chapter 9.8. Reference [13] lists the cross-references between sites and files in the RODOS FixData base, and reference [14] lists all radionuclide inventory files provided with the RODOS system.

9.2 Design

A radionuclide inventory file contains the nuclide-specific activity inventory in a reactor at the time "end of chain reaction" for one nominal thermal power and between one and four times of operation at the given nominal thermal power. The physical model for editing these data for STERM is described in [2], Chapter on "Start-up radionuclide inventory for source term calculations".

9.3 Software

Code package SETINV is part of ASY and resides in the same directory as STERM/USER. It does not distinguish between operation modes, nor between RODOS A or B Users.

SETINV consists of a group of FORTRAN 77 subroutines with the following extension to the Language Standard:

- FORTRAN INCLUDE statements.
- Statement IMPLICIT NONE.
- Names longer than 6 characters.

Table 13 lists the subroutines of SETCHN as specified in the STERM comment library, member "Programs". Table 14 summarises the support software.

Note: In RODOS PV6final, the Native Language Support (NLS) via external files was introduced for the source term code, and former ".ff"-files became normal "FORTRAN .f"-files.

Table 13: Subroutine inventory of SETINV

C	=====	C
C	PACKAGE -> SETINV of STERM	C
C	TASKS -> SETUP OF INITIAL INVENTORY	C
C	=====	C
C	SETINV -> MAKE OUTPUT INVENTORY <ATINV>	C
C	SETINV_PRINT -> PRINT INVENTORY INFORMATION	C
C	SETINV_READ -> READ INPUT INVENTORY FROM FILE	C
C	=====	C

Table 14: Support software for SETINV

- Code package SETCHN for STERM internal nuclide list.
- Program package SETNUK [4] for FixData and near range nuclide information.
- Code package UTILITY²⁴ for general auxiliary subroutines.

9.4 Input Data

- SHM section on "Site Information" (see Table 8).
- SHM section on "Nuclide Information" filled by SETNUK (see Table 8).
- COMMON/XNUSEL/ with the source term nuclide cross-reference list ISOQTR provided by SETCHN (see Chapter).
- File with the radionuclide inventory with name given by SHM variable BlockInve of data class CANLAG (see Table 8).

9.5 Output Data

The information SETINV generates for SETINV-external use is contained in the following STERM internal common block:

- COMMON/XINVNT/ with ATINV telling for each ISOQTR source term nuclide its radionuclide inventory.

SETINV prints tables with inventory information in file <Prognosis|Diagnosis.InventoryData> in directory <~rodos/roextern/outall/user_ID/run_ID/>. This file is accessible with the Graphics Manager of the RODOS system, see [5], Chapter on "Prognosis Results Accessible with the RODOS Graphics".

In addition, SETINV can also provide debug printout in the RODOS log-file for investigation of its performance steered by flag IPRINT in the argument list (see comment section in header of subroutine SETINV).

²⁴ For the subroutine inventory of this package see the STERM comment library, member "Programs".

9.6 Verification and Tests

Most of the SETINV functionality's is checked when performing the tests described in Chapter 8.3. The formulae for interpolating the thermal power and operation days were tested separately by putting in some new values and comparing the results with hand-calculations.

9.7 Error Report

Since the first issue of SETINV in 1999 (as successor of former SETIN1/2), there was one reported error in March 2002 that was removed with the next issued patch of RODOS.

9.8 Radionuclide Inventory Data Files

9.8.1 File Name, Location, Size

The name of a RODOS radionuclide inventory file must start with the 5 characters "INVE." followed by up to 27 arbitrary alphanumerical characters for the inventory name.

Examples for good and bad file names: "INVE.MyInventory" is an allowed file name. "MyInventory" is not an allowed file name, because the prefix part is missing.

Radionuclide inventory files reside in RODOS directory `<~rodos/roextern/data/sourceterm/inventories/>`.

Typically, the file sizes are between about 2000 and 8000 bytes.

9.8.2 File Description

Table 15 shows a part of a radionuclide inventory file named `<INVE.PWR_3733Mwth_Leitfaden95>` and the respective format statements. Explanations follow after the table!

Table 15: Structure of a RODOS radionuclide inventory file

format	file contents (example)
F-01	# =====
F-02	# NCOMM = 14 *****Number of comment lines to follow this line*****
F-01	# =====
F-01	# INVNME maximally 32 characters (Prefix "INVE." is cogent)
F-01	# INVPOW in [MW therm.]
F-01	# INVNUM number of inventories with INVDAY in ascending order (max. 4).
F-01	# INVDAY in [days]; 999 for equilibrium core.
F-01	# RINVIN in [Bq] for time point <end of chain reaction>.
F-01	# =====
F-01	# Inventare aus Leitfaden fuer den Fachberater ... , Veroeffentlichungen
F-01	# der SSK, Band13, 2.Aufl.(1995), ISBN 3-437-11639-8, Seiten 2-9
F-01	# =====
F-01	# Claudia Landman Version:11.10.2001
F-01	# =====
F-01	# =12345678901234567890123456789012
F-03	# =====
F-04	INVNME =INVE.PWR_3733Mwth_Leitfaden95
F-04	INVPOW = 3733
F-05	INVNUM = 4
F-01	INVDAY = 1 33 333 999
....	# ----- RINVIN for INVPOW MW therm.; burn-up time 1 days
....

F-06
F-07	# Kr- 85m Kr- 85 Kr- 87 Kr- 88
....	1.43E+18 4.08E+13 2.86E+18 4.03E+18
.....
F-06
F-07	# I -129 I -131 I -132 I -133 I -134 I -135
....	0.00E+00 2.36E+17 9.66E+17 4.16E+18 8.76E+18 6.76E+18
.....
F-06
F-01	# BLCKEND -----
....	# ----- RINVIN for INVPOW MW therm.; burn-up time 33 days
.....
..... and so on
F-06
F-01	# BLCKEND -----
....	# ----- RINVIN for INVPOW MW therm.; burn-up time 333 days
.....
..... and so on
F-06
F-01	# BLCKEND -----
....	# ----- RINVIN for INVPOW MW therm.; burn-up time 999 days
.....
F-06
F-07	# Kr- 85m Kr- 85 Kr- 87 Kr- 88
....	1.04E+18 2.81E+16 1.95E+18 2.77E+18
.....
F-06
F-07	# I -129 I -131 I -132 I -133 I -134 I -135
....	0.00E+00 3.63E+18 5.32E+18 7.58E+18 8.21E+18 7.06E+18
.....
F-06
F-01	# FILEEND -----

Formats (Character "^" represents the blank character, "d" some digit 0...9)	
F-01	: 81 arbitrary characters
F-02	: 9 arbitrary characters, right-adjusted integer with 4 digits
F-03	: 8 characters, 32 characters with inventory name
F-04	: 8 characters, right-adjusted integer with 5 digits
F-05	: 8 characters, 4 * (right-adjusted integer with 5 digits)
F-06	: 7 * (3 blank characters, 7 characters with nuclide names)
F-07	: 7 * (1 blank character, floating point number with 12 digits {sign}d.ddE+dd)

The file syntax is described below. The character "^" represents the blank character.

Leading lines

- **Line 1:** Obligatory comment line ("-----" in the Table).
- **Line 2:** Number of comment lines to follow (variable NCOMM).
- **Lines 3 to 2+NCOMM:** Arbitrary comment lines.

Data lines (D0=2+NCOMM) - general information about the inventory

- **Line D0+1:** Name of the inventory (up to 32 characters).
- **Line D0+2:** Nominal power [MW] for which the inventory applies. **Must be greater than zero!**
- **Line D0+3:** Number of inventory data subsets to follow (between 1 and 4). Each subset applies to the thermal power specified in the previous line, and to a specific burn-up time in the unit [day] that will be specified in the following line.
- **Line D0+4:** Specification of the INVNUM burn-up times in the unit [day]. **1st time must be greater than zero; times must increase from left to right!**

Then for each burn-up time subset

- One obligatory comment line.

followed in turn by

- One line with nuclide names (between 1 and seven per line).
- One line with inventory data for above nuclides.

Data may be given for any number of nuclides in any order. Each data line must not contain less than one or more than seven entries.

STERM/USER recognises the nuclides by their names and skips any entry which is not of interest. If a nuclide name does not comply with the naming conventions described in [4], it is not recognised as a "good nuclide" and will always be ignored.

ended by

- BLCKEND (another subset follows), or FILEEND (nothing more to follow).

9.8.3 File Errors

A radionuclide inventory file is used by SETINV at runtime of RODOS and is checked when it is read in. Format errors, structure inconsistencies, values out of range, or requested but not found inventory names are reported, and program execution stops.

9.8.4 File Maintenance, Implementation of New Files

Radionuclide inventory files are provided by plant or source term experts. File maintenance is duty of the provider.

A RODOS system administrator can add inventory files to the RODOS inventory data library by. Report [14] gives hints for successful implementation of such Sinceventory files.

10 APPENDIX 2: STERM Support Task SETCHN for Setting Up Decay Chain Information

10.1 Task

In all time spans between the reactor shutdown and the releases of radionuclides, STERM takes account of the in-growth of nuclides from radioactive decay chains of any depth. The definition of the decay chains (nuclide members and decay fractions) is contained in file *<in.decaychains>* that is provided with the RODOS system and described in Chapter 10.8. Code package SETCHN ("SETUp of decay CHaiNs") reads that file, analyses the data, and edits them for disposal in STERM.

10.2 Design

SETCHN starts processing by activating all RODOS near range nuclides also as "STERM internal nuclides", except the ones marked for exclusion by program package SETRLG. Then it begins to read file *<in.decaychains>*. For each decay chain it first builds a temporary array which maps all daughter and mother nuclides belonging to the decay chain with the available fix data nuclides, again ignoring all those marked for exclusion by SETRLG.

When the information for the decay chain is complete, SETCHN inspects the temporary array for the presence of at least one active near range nuclide with a mother in the chain. If none was found, the chain is ignored. If one was found, all chain members upwards of the active near range nuclide are added to the STERM internal nuclides, and the number of mothers and lists with the mother addresses and decay fractions are updated. After that, SETCHN turns to the next decay chain.

10.3 Software

Code package SETCHN is part of ASY and resides in the same directory as STERM/USER. It does not distinguish between operation modes, nor between RODOS A or B Users.

SETCHN consists of a group of FORTRAN 77 subroutines with the following extension to the Language Standard:

- FORTRAN INCLUDE statements.
- Statement IMPLICIT NONE.
- Names longer than 6 characters.

Table 18 lists the subroutines of SETCHN as specified in the STERM comment library, member "Programs". Table 17 summarises the support software.

Note: In RODOS PV6final, the Native Language Support (NLS) via external files was introduced for the source term code, and former ".ff"-files became normal "FORTRAN .f"-files.

Table 16: Subroutine inventory of SETCHN

C	=====	C
C	PACKAGE -> SETCHN of STERM	C
C	FILE -> [setchn.f]	C
C	TASKS -> SETUP OF DECAY CHAIN INFORMATION	C
C	=====	C
C	SETCHN -> READ AND DECODE DECAY CHAIN DATA	C
C	SETCH1 -> CHECK CONTINUATION CARD CODE	C
C	SETCH2 -> MARK NUCLIDE AS SELECTED AND BUILD UP TEMPORARY	C
C	MOTHER LIST	C
C	SETCH3 -> BUILD UP FINAL MOTHER LIST	C
C	=====	C

Table 17: Support software for SETCHN

- Code package SETRLG for FixData nuclide ↔ release group cross-reference table.
- Program package SETNUK [4] for FixData and near range nuclide information.
- Code package UTILITY²⁵ for general auxiliary subroutines.

10.4 Input Data

- SHM section on "Nuclide Information" filled by SETNUK (see Table 8).
- COMMON/XDRLGR/ filled by SETQTR (see Chapter 11.4).
- File <in.decaychains>.

10.5 Output Data

The information SETCHN generates for SETCHN-external use is contained in the following STERM internal common blocks:

- COMMON/XNUSEL/ with ISOQTR telling of each FixData nuclide ISO if it is STERM internal nuclide (ISOQTR(ISO)=1) or not (ISOQTR(ISO)=0).
- COMMON/XDECHN/ with NMOTHR (number of mothers for each STERM internal nuclide), IMOTHR (FixData nuclide addresses ISO of the mothers for each STERM internal nuclide), and FMN (fractions of the mother decays leading to the daughter, with the fractions taken from [15], Appendix 4).

SETCHN does not generate any printout unless flag IPRINT in the argument list tells it to provide debug printout in the RODOS log-file for investigation of its performance (see comment section in header of subroutine SETCHN).

²⁵ For the subroutine inventory of this package see the STERM comment library, member "Programs".

10.6 Verification and Tests

In the initial test phase in 1996, the debug output was inspected and it was ascertained that the task performs correctly.

10.7 Error Report

Since the first issue in 1996, the task is in use and no errors were detected at FZK or reported by external users.

10.8 Decay chain file <in.decaychains>

10.8.1 File Name, Location, Size

The decay chain file has the fixed name <in.decaychains> and resides in RODOS directory <~rodos/roextern/data/sourceterm/setup/>.

The file size is 11395 bytes (file version 6.2.2005).

10.8.2 File Description

Table 18 shows a part of file <in.decaychains> together with the respective formats. Explanations follow after the table!

Table 18: Fraction of the decay chain file <in.decaychains>

format	file contents (example)
F-01	-----
F-02	NCOMM = 13 ***** number of comment lines to follow *****
F-01	-----
F-01	C. Landman/FZK ***** file in.decaychains ***** Version : 06.02.2005
F-01	-----
F-01	Nuclide names must comply with the conventions described in
F-01	[RODOS(RA1)-TN(02)-04].
F-01	-----
F-01	Decay data from D.C.KOCHER: Radioactive Decay Data Tables, Appendix 4,
F-01	DOE/TIC-11026, U.S. Department of Energy (1981).
F-01	-----
F-01	For inventory calculations (decay branches to stable nuclides omitted).
F-01	Special setup for isotopes in RODOS-PV5.0 (A few decay branches were
F-01	combined to one channel so nothing gets lost).
F-01	-----
F-03	& chain-idepth=1-----times : nstep= -1 dstep=0.000E+00h---crd=01
F-04	& I -132 from Te-132 fmd=100.00%
F-03	&skchain-idepth=4-----times : nstep= -1 dstep=0.000E+00h---crd=06
F-04	d Te-133 from Te-133m fmd= 13.00%
F-04	d I -133 from Te-133m fmd= 87.00%
F-04	d I -133 from Te-133 fmd=100.00%
F-04	d Xe-133m from I -133 fmd= 2.88%
F-04	d Xe-133 from I -133 fmd= 97.12%
F-04	& Xe-133 from Xe-133m fmd=100.00%
F-03	& chain-idepth=3-----times : nstep= -1 dstep=0.000E+00h---crd=05
F-04	d Te-133 from Te-133m fmd= 13.00%

F-04	d	I -133	from	Te-133m	fmd= 87.00%
F-04	d	I -133	from	Te-133	fmd=100.00%
F-04	c	Xe-133	from	I -133	fmd=100.00%
F-05	& Simplified chain (Xe-133m state left out).				
....	& chain.....				
Formats (Character "^" represents the blank character, "d" some digit 0..9)					
F-01	: 72 arbitrary characters				
F-02	: 9 arbitrary characters, right-adjusted integer with 4 digits				
F-03	: A16,I1,26X,I3,7X,1PE9.3,A1,7X,I2				
F-04	: A3,7X,A7,3X,A4,3X,A7,7X,0PF6.2				
	(Apart from the key-word "from", the key word "void" is possible; however, cards with this key-word are ignored in RODOS).				
F-05	: A3, A69				

The sequence of decay chains in the file has no functional relevance at all. Rearrangements can freely be made at any time.

The file syntax is as described below. The character "^" represents the blank character.

Leading lines

- **Line 1:** Obligatory comment line ("-----" in the Table).
- **Line 2:** Number of Comment lines to follow (variable NCOMM).
- **Lines 3 to 2+NCOMM:** Comment lines.

Then for each decay chain

- **The chain definition card ("&-cards")**

The first character must be "&" and characters 4-16 must be "chain-idepth=". The value of idepth is currently not used in RODOS. NSTEP and DSTEP must have the values as shown in the example. The value following "crd=" is the number of lines to follow for the decay chain and must be greater than zero.

If the first three characters in the line are "&^^", then the decay chain represented by the chain definition card is evaluated; if they are "&sk", then the chain will be ignored (i.e. the number of cards given by "crd=" is skipped).

- **Data cards ("d-cards")**

The first three characters tell the type of the next card and can have the values "^&^", "^d^" or "^c^" Then follow the name of the daughter, the keyword "from", the name of the mother and the fraction of the mother decay leading to the daughter (in percent). The nuclide names must comply with the conventions described in [4].

- **Comment cards ("c-cards")**

The first three characters tell the type of the next card and can have the values "^&^", "^d^" or "^c^" The rest of the line can contain any arbitrary comment.

10.8.3 File Errors

File *<in.decaychains>* is used by STERM at runtime of RODOS and is checked when it is read in. Format errors, structure inconsistencies, or values out of range are reported, and program execution stops.

If after modification of the file *<in.decaychains>* there are error stops and the problem cannot be detected by eye inspection of the file, calling subroutine SETCHN with print flag IPRINT=2 helps: SETCHN then writes to the RODOS log-file every card it reads, and any problem card can be located.

10.8.4 File Maintenance - Whose Task, When Necessary, What To Do

Maintenance of file *<in decaychains>* is matter of the RODOS team.

After the introduction of new nuclides into the RODOS FixData base, file *<in.decaychains>* must be checked for missing or partial decay chains:

- If a new nuclide is member of a decay chain not yet implemented in the file => The new decay chain must be implemented.
- If a new nuclide complements a decay chain which was already implemented in a simplified form to include the contribution from the nuclide formerly not present => The decay branch for this nuclide must be implemented.

An example is shown in Table 18. Nuclide Xe-133m is currently not in the FixData base of RODOS, and the simplified chain part I-133 -> Xe-133 is implemented. The full chain is also included in the file but marked for skipping. If Xe-133m would be included among the RODOS nuclides, the full chain should be activated and the reduced chain eliminated or marked for skipping.

When new data are added, the decay fractions from [15], Appendix 4, should be used to preserve consistency in the file.

11 APPENDIX 3: STERM Support Task SETRLG for Setting Up Release Group Information

Note: In RODOS PV6final, the Native Language Support (NLS) via external files was introduced for the source term code, with the following consequences for STERM support task SETRLG:

- Former ".ff"-files became normal "FORTRAN .f"-files.
- Former file *<in.releasegroups>* became *<in.releasegroups.structure>* with some changes in the file format (see Chapter 11.8).
- Additional input files *<in.releasegroups.keywords>* became necessary (see Chapter 11.9).
- Additional output files had to be generated (see Chapter 11.10).

11.1 Task

The source term release group schemes are contained in a file named *<in.releasegroups.structure>* that is provided with the RODOS system and described in Chapter 11.8. The file declares which sets of nuclide release groups, or "nuclide release blocks" are known to RODOS, which release groups are contained in each block, and which chemical elements and individual molecules belong to each group. Code package SETRLG ("SETup of ReLease Groups") reads the file, analyses the data, and edits them for disposal in STERM.

Table 19 shows the release blocks currently implemented in RODOS, by means of the block key and the release groups making up the blocks.

Table 19: Current nuclide release group schemes

release block key	release groups contained in nuclide release block
NOBLE-JOD-AEROSO	01. noble gases 02. iodine 03. aerosols (all aerosol nuclides are treated equally!)
NOB-JOD-AER-T-TO	<i>As NOBLE-JOD-AEROSO plus HT and HTO:</i> Groups N°1 to 3 as for NOBLE-JOD-AEROSO 04. tritium (HT) 05. tritiated water (HTO)
DRS-A_7_GROUPS	<i>Groups from the German Risk Study, Phase A [16]:</i> 01. noble gases 02. iodine 03. alkaline metal 04. tellurium/antimony 05. alkaline earth 06. ruthenium group 07. lanthanides
DRS-A_7_GROUPS09	<i>As DRS-A_7_GROUPS plus HT and HTO:</i> Groups N°1 to 7 as for DRS-A_7_GROUPS

	08. tritium (HT) 09. tritiated water (HTO)
DRS-B_9_GROUPS	<i>Groups from the German Risk Study, Phase B</i> [[17]: 01 noble gases 02 iodine 03 alkaline metal 04 tellurium/antimony 05 strontium 06 barium 07 ruthenium group 08 lanthanides 09 cerium group
MAAP4_10_GROUPS	01 noble gases 02 iodine 03 rubidium 04 strontium 05 ruthenium group 06 alkaline earth 07 barium 08 lanthanides 09 actinides 10 tellurium

11.2 Design

Each user source term file contains the name key²⁶ of the release block it refers to. SETRLG begins processing by searching file *<in.releasegroups.structure>* for the corresponding block name. If it was not found, an error message is issued and execution stops. If the block was found, the number of release groups belonging to the release block is determined, and two arrays are set up, one for the names of the release groups belonging to the block, and one to indicate for all RODOS nuclides to which release group they belong to. The latter is achieved by comparing from file *<in.releasegroups.structure>* the two character abbreviations for the chemical elements or molecules belonging to the group, with the first two characters of the RODOS nuclides names from the nuclide section of the ASY shared memory of RODOS. On match, the following release group membership indicator is given to the FixData nuclide:

- If not assigned to the "waste bin", the number of the release group it belongs to (numbering is 1...NRLGRP-1 in the order as the groups stand in the file).
- If assigned to the "waste bin", the value zero.

²⁶ STERM shared memory variable CRLGID of data class CSTRMX.

11.3 Software

Code package SETRLG is part of ASY and resides in the same directory as STERM/USER. It does not distinguish between operation modes, nor between RODOS A or B Users, nor between the RODOS User Interfaces.

SETRLG consists of one FORTRAN 77 subroutine with the following extension to the Language Standard:

- FORTRAN INCLUDE statements.
- Statement IMPLICIT NONE.
- Names longer than 6 characters.

Table 20 lists the subroutines of SETRLG as specified in the STERM comment library, member "Programs". Table 21 summarises the support software.

Table 20: Subroutine inventory of SETRLG

C	=====	C
C	PACKAGE -> SETRLG of STERM	C
C	FILE -> [setrlg.f]	C
C	TASK -> SETUP OF RELEASE GROUPS	C
C	=====	C
C	SETRLG -> SETUP OF RELEASE GROUPS	C
C	GET_GRPNAM -> GET RELEASE GROUP NAME IN NATIVE LANGUAGE	C
C	(TAKE SYMBOLIC ENGLISH NAME IF NOT FOUND)	C
C	=====	C

Table 21: Support software for SETRLG

- | |
|---|
| <ul style="list-style-type: none"> • Program package SETNUK [4] for FixData and near range nuclide information. • Code package UTILTY²⁷ for general auxiliary subroutines. |
|---|

11.4 Input Data

- SHM section on "Nuclide Information" filled by SETNUK (see Table 8).
- Variable CRLGID of data class CSTRMX of the SHM section on "User Source Term Input" (see Table 8), telling the release block key.
- File <in.releasegroups.structure>
This file in RODOS PV6final corresponds to the former file <in.releasegroups> (but with slightly changed file format).
- File <in.releasegroups.keywords> in English
This file translates the symbolic names of the release groups to English for Native Language Support. This file must exist!

²⁷ For the subroutine inventory of this package see the STERM comment library, member "Programs".

- Files *<in.releasegroups.keywords>* in other languages than English
 These files serve for the translation of the symbolic names of the release groups to other languages than English for Native Language Support. They should be generated when adapting the RODOS system to national conditions (see [18]), but - contrary to the English file - *<in.releasegroups.keywords>* files in other languages need not necessarily exist²⁸.

11.5 Output Data

The information SETRLG generates for SETRLG-external use is contained in STERM internal common blocks.

- COMMON/XCRLGR/ with CRLGRP telling the release group names and CRLTYP their types for the release block with key CRLGID.
- COMMON/XDRLGR/ with the release groups definitions for the release block with key CRLGID: NRLGRP²⁹ is the number of release groups in the release block. NINRLG is the number of near range nuclides in each release group. JRLGRP tells for each FixData nuclide that is a near range nuclide to which release group it belongs.

SETRLG prints the current nuclide composition of the release groups during a RODOS run to file *<Prognosis|Diagnosis.STermFromUser>* in directory *<~rodos/roextern/outall/user_ID/run_ID/>*. This file is accessible with the Graphics Manager of the RODOS system, see [5], Chapter on "Prognosis Results Accessible with the RODOS Graphics".

In addition, SETRLG can also provide debug printout in the RODOS log-file for investigation of its performance steered by flag IPRINT in the argument list (see comment section in header of subroutine SETRLG).

11.6 Verification and Tests

In the initial test phase in 1996 and after the inclusion of the new "waste bin group" to accommodate the tritium compounds in 1998, the extended output was inspected and it was ascertained that the task performs correctly.

11.7 Error Report

Until RODOS PV6.0_Patch 7

Since the first issue in 1996, there was one reported error in May 2004. The error was caused by the transition to the dynamic initialisation windows, in which the user may now choose between different release blocks. It was detected during the installation phase and removed before the first issue of RODOS PV6.0.

²⁸ If a national file does not exist, the English one will be taken.

²⁹ NRLGRP will be the number of release groups read from file *<in.releasegroups>* minus one for getting rid of the "waste bin group", which is always the last group in each release block in the data file.

No errors were reported so far by external users.

Since RODOS PV6final (with new NLS features)

No errors were reported so far.

11.8 Release Group File <in.releasegroups.structure> → since RODOS PV6final

11.8.1 File Name, Location, Size

The file has the fixed name <in.releasegroups.structure> and resides in RODOS directory <~rodos/roextern/data/sourceterm/setup/>.

The file size is about 5000 bytes.

11.8.2 File Description

Table 22 shows a part of file <in.releasegroups.structure> together with the respective formats. Explanations follow after the table!

Note: File <in.releasegroups.structure> in RODOS PV6final corresponds to the former file <in.releasegroups> but shows a slightly changed file format and some conceptual changes concerning the names of the release groups - see the explanations further below.

Table 22: Release group file <in.releasegroups.structure>

format	file contents (example)
F-01	-----
F-02	NCOMM = 28 ***** number of comment lines to follow *****
F-01	-----
F-01	C. Landman/FZK *** 1st version *** 15.05.2007
F-01	-----
F-01	Formerly file <in.releasegroups> (up to RODOS PV6Patch7).
F-01	RODOS Language support => New file structure.
F-01	Documentation: [User Source Term Interface STERM/USER],
F-01	Chapter on "STERM Support Task SETRLG ...".
F-01	-----
F-01	Specifications for EACH RELEASE BLOCK
F-01	-----
F-01	NRLGRP : Number of release groups to follow.
F-01	CRLGRP : Name of release group IN ENGLISH !!!
F-01	* Names left-adjusted, 1-16 characters
F-01	* There MUST be a "waste bin group" (even if NMEMBR=0)
F-01	* The "waste bin group" MUST come last.
F-01	* The "waste bin group" name MUST start with character "+".
F-01	NMEMBR : Number of members in release group.
F-01	(NEW IN JULY 2000) FOLLOWED BY (1 BLANK, CHARACTER*1)
F-01	n to characterise noble gases,
F-01	i to characterise iodine,
F-01	a to characterise aerosols,
F-01	x to characterise all other types.
F-01	Names of release groups members (CHARACTER*2):
F-01	* All entries must be separated by at least one blank.
F-01	* Last character in each line must be the blank character.
F-01	* Each release group must contain ALL CHEMICAL ELEMENTS in the
F-01	* file ~rodos/roextern/data/sourceterm/setup/in.FixDataNuclides.
F-01	-----
F-03	NOBLE-JOD-AEROSO*****NRLGRP = 4*****SKIP =15
F-01	-----
F-04	CRLGRP =Noble_gases
F-05	NMEMBR = 2 n
see text	Kr Xe
F-04	CRLGRP =Iodine
F-05	NMEMBR = 1 i
see text	I
F-04	CRLGRP =Aerosols
F-05	NMEMBR = 33 a

see text	Ag Am Ba Ce Cf Cm Co Cs Ir La Mn Mo Na Nb Nd Np Po Pr Pu Ra Rb Rh Ru
see text	Se Sb Sr Tc Te U W Y Yb Zr
F-04	CRLGRP =+To_be_ignored
F-05	NMEMBR = 2 x
see text	T TO
F-01	-----
F-03	NOB-JOD-AER-T-TO*****NRLGRP = 6*****SKIP =20
F-01	-----
F-04	CRLGRP =Noble_gases
F-05	NMEMBR = 2 n
see text	Kr Xe
F-04	CRLGRP =Iodine
F-05	NMEMBR = 1 i
see text	I
F-04	CRLGRP =Aerosols
F-05	NMEMBR = 33 a
see text	Ag Am Ba Ce Cf Cm Co Cs Ir La Mn Mo Na Nb Nd Np Po Pr Pu Ra Rb Rh Ru
see text	Se Sb Sr Tc Te U W Y Yb Zr
F-04	CRLGRP =Tritium(HT)
F-05	NMEMBR = 1 x
see text	T
F-04	CRLGRP =Trit.Water(HTO)
F-05	NMEMBR = 1 x
see text	TO
F-04	CRLGRP =+To_be_ignored
F-05	NMEMBR = 0 x
see text	-----
....	DRS-A_7_GROUPS *****NRLGRP = 8*****SKIP =26
....	-----
....26 lines for release block DRS-A 7 GROUPS

Formats (Character "^" represents the blank character, "d" some digit 0..9)

F-01 : 72 arbitrary characters

F-02 : 9 arbitrary characters, right-adjusted integer with 4 digits

F-03 : 16 characters (block name), 13 arbitrary characters, right-adjusted integer with 2 digits, 34 arbitrary characters, right-adjusted integer with 2 digits.

F-04 : **The keyword indicator "CRLGRP =", followed by 16 characters giving the symbolic group name in English language - once defined it must not be changed!!!**

F-05 : 8 arbitrary characters, right-adjusted integer with 3 digits, 1 arbitrary character, 1 characters giving the atmospheric dispersion classification of the group (a,i,n,x, see explanations).

The sequence of release blocks in the file has no functional relevance. Rearrangements can freely be made at any time.

Inside each release block a "waste bin group" must be present, even if it is empty (NMEMBR=0), and it must be the last group in the block. Apart from that, the sequence of release groups in a release block is arbitrary, but only when the release block is freshly introduced into the file. **The sequence of release groups belonging to one particular release block must never be changed later!** This would corrupt all already existing source terms files that rely on the fixed sequence of the nuclide release groups.

The file syntax is as described below. The character "^" represents the blank character.

Leading lines

- **Line 1 in file:** Obligatory comment line ("-----" in the Table).

- **Line 2 in file:** Number of comment lines to follow (variable NCOMM).
- **Lines 3 to 2+NCOMM in file:** Comment lines.

Then for each release block

- **Line 1 in each block:** (a) Name of release block (b) Number of release groups in current release block (variable NRLGRP) (c) Number of lines to follow for this block (**including** the last obligatory comment line which closes the release block, see further below).
- **Line 2 in each block:** Obligatory comment line ("-----" in the Table).

Then for each release group

- **Line 1 in each group:** The keyword indicator "CRLGRP =", followed by 16 characters giving a SYMBOLIC group name (variable CRLGRP).

(SINCE RODOS PV6FINAL) SUCH LINES DEFINE THE RELEASE GROUP NAME KEYWORDS FOR THE NATIVE LANGUAGE SUPPORT FILES AS DESCRIBED IN CHAPTER 11.9. ONCE DEFINED, A SYMBOLIC RELEASE GROUP NAME MUST BE USED ANYWHERE IN FILE <IN.RELEASEGROUPS.STRUCTURE> FOR IDENTICAL RELEASE GROUP DEFINITIONS.

- **Line 2 in each group:** (a) Number of group members (variable NMEMBR; can be zero). (b) Characterisation of release group with respect to atmospheric dispersion("n" for noble gas, "i" for iodine, "a" for aerosol, "x" for unspecified).
- **If the number of members is greater zero, the next line(s) contain the release group members!** Group members must be specified by *the first 2 characters including the blank character for names with only one character, e.g. "I^"* of the SETNUK names described in [4]. Separators between entries are one or more blanks. Members can be entered on one or more lines; each line can contain any number of entries in any order. However, character 72 in each line must be the blank character (if it is not, SETRLG will issues a corresponding message and stops).
- **Last Line in each release block:** Obligatory comment line ("-----" in the Table).

11.8.3 File Errors

File <in.releasegroups.structure> must exist. The existence is checked by SETRLG whenever SETRLG is called. Non-existence of the file is reported, and SETRLG stops executing.

File <in.releasegroups.structure> is checked by SETRLG whenever SETRLG is called. Format errors, structure inconsistencies, values out of

range, or requested but not found release blocks are reported, and SETRLG stops executing.

11.8.4 File Maintenance, Implementation of New Entries

Maintenance of file `<in.releasegroups.structure>` is matter of the RODOS team.

If you are interested in adding a new release block to RODOS, contact the RODOS team (cc. Claudia Landman).

11.9 Release Group NLS Files `/in.releasegroups.keywords>` → since RODOS PV6final

In RODOS PV6final, the Native Language Support (NLS) via external files was introduced. Because of that, new files `<in.releasegroups.keywords>` became necessary for the NLS adaptation of release group names for the source term input for RODOS.

In the following, the term `{national_language_identifier}` stands for the RODOS-internal name identifier representing some national language, e.g. "german" for the German language.

11.9.1 File Name, Location, Size

All NLS release group name files have the fixed name `<in.releasegroups.keywords>`.

For a given `{national_language_identifier}`, the corresponding `<in.releasegroups.keywords>` file resides in RODOS directory `<~rodos/rolang/nls/{national_language_identifier}>`.

A file `<english/in.releasegroups.keywords>` is delivered with the standard installation of the RODOS system.

The file sizes are typically around 500 bytes.

11.9.2 File Description

File `<english/in.releasegroups.keywords>`, that is delivered with the standard installation of the RODOS system, serves for the translation of the symbolic names of the release groups in file `<in.releasegroups.structure>` described in Chapter 11.8 into English. This file will be used as the default NLS file if a corresponding NLS-file does not exist for some language - therefore it must always exist.

Table 23 shows the current file `<english/in.releasegroups.keywords>` corresponding to file `<in.releasegroups.structure>` from Table 22. Explanations follow after the Table!

Table 23: Release group file `</english/in.releasegroups.keywords>`

file contents (example)


```

CRLGRP =Actinides/Ce
Actinides/Ce
CRLGRP =Aerosols
Aerosols
CRLGRP =Ba/Ra/Sr
AlkaliEarth_Grp.
CRLGRP =Cs/Na/Rb
AlkaliMetal_Grp.
CRLGRP =Ba/Ra
Barium/Radium
CRLGRP =Cs/Na
Cs/Na
CRLGRP =Iodine
Iodine
CRLGRP =Lanthanides
Lanthanides
CRLGRP =Noble_gases
Noble_gases
CRLGRP =Rb
Rubidium
CRLGRP =RutheniumGrp.
RutheniumGrp.
CRLGRP =Sb/Se/Te/Po
Sb/Se/Te/Po
CRLGRP =Sr
Strontium
CRLGRP =Tritium(HT)
Tritium(HT)
CRLGRP =Trit.Water(HTO)
Trit.Water(HTO)
CRLGRP =+To_be_ignored
+To_be_ignored

```

For each release group scheme, there are two lines:

- The first line contains the keyword indicator "CRLGRP =", followed by the unique symbolic group name from file *<in.releasegroups.structure>*.
- The second line contains a NLS (here: English) name equivalent of the symbolic group name.

The sequence of pairs (keyword line, translation) in the file has no functional relevance. Rearrangements can freely be made at any time.

11.9.3 File Errors

File *<english/in.releasegroups.keywords>* must exist. The existence is checked by SETRLG whenever SETRLG is called. Non-existence of the file is reported, and SETRLG stops executing.

If there is (erraneously) more than one line referring to the same symbolic group name, the first pair (keyword line, translation) would be taken by SETRLG and all others would get ignored.

Checks on the file format or contents are not made in RODOS.

11.9.4 File Maintenance, Implementation of New Entries

Maintenance of the file *<english/in.releasegroups.keywords>* is matter of the RODOS team.

Files NLS-file /in.releasegroups.keywords> for other language than English are not essential for the functioning of RODOS - if they does not exist, the English file will be taken.

For providing a new NLS-file /in.releasegroups.keywords> for some language {national_language_identifier}, see Reference [18].

11.10 Release group information for RODOS-Lite → since RODOS PV6final

In the following, the term {national_language_identifier} stands for the RODOS-internal name identifier representing some national language, e.g. "german" for the German language.

11.10.1 File Name, Location, Size

The RODOS-Lite user interface requires for each nuclide release block from Table 19 ASCII-files with names

<~rodos/rolite/in.releasegroups.{release_block_name}.{national_language_identifier}>.

The sizes of such files are typically about 900 bytes.

11.10.2 File Description

Each file in.releasegroups.{release_block_name}.{national_language_identifier} contains the release group scheme of the corresponding release block in the respective national language, together with the release group composition for the current FixData nuclides of RODOS. An example is shown in Table 24 below.

Table 24: Example for an NLS-release group file for RODOS-Lite

Example file <in.releasegroups.DRS-A_7_GROUPS.german>							
14DRS-A_7_GROUPS							
NRLGRP = 8							
1Edelgase 7							
Kr- 85m	Kr- 85	Kr- 87	Kr- 88	Xe-133	Xe-135	Xe-138	
2Jod 6							
I -129	I -131	I -132	I -133	I -134	I -135		
3Alkalimetalle 7							
Cs-134	Cs-136	Cs-137	Cs-138	Na- 24	Rb- 86	Rb- 88	
4Sb/Se/Te/Po 14							
Sb-127	Sb-129	Se- 75	Te-127m	Te-127	Te-129m	Te-129	Te-
131	Te-131m	Te-132					
Te-133m	Te-133	Te-134	Po-210				
5Erdalkalimetalle 6							
Ba-140	Ra-226	Sr- 89	Sr- 90	Sr- 91	Sr- 92		
6Ruthen-Gruppe 10							
Ag-110m	Co- 58	Co- 60	Mn- 54	Mo- 99	Ru-103	Ru-105	Ru-
106	Rh-105	Tc- 99m					
7Lanthaniden 27							
Am-241	Ce-141	Ce-143	Ce-144	Cf-252	Cm-242	Cm-244	Ir-
192	La-140	Nb- 95					
Nd-147	Np-239	Pr-143	Pu-238	Pu-239	Pu-240	Pu-241	U -
234	U -235	U -238					
W -185	W -187	Y - 90	Y - 91	Yb-169	Zr- 95	Zr- 97	

8+NichtBetrachtet 2
T - 3x TO- 3x

11.10.3 File Maintenance

The maintenance of the release group information for RODOS-Lite is up to the FZK RODOS team. For that purpose, there exists a STERM-stand-alone FORTRAN program package [RelGrps].

Package [RelGrps] creates for all available RODOS language identifiers *{national_language_identifier}* all files

<~rodos/rolite/in.releasegroups.{release_block_name}.{national_language_identifier}>.

for all nuclide release blocks contained in file *<~rodos/roextern/data/sourceterm/setup/in.releasegroups.structure>*, with all radionuclides contained in file *<~rodos/roextern/data/sourceterm/setup/in.FixDataNuclides>*.

Program package [RelGrps] must be executed:

- On each fresh installation of the RODOS system.
- If a new RODOS language identifier was introduced for which a file *in.releasegroups.keywords* exists.
- If modifications were made to any existing file *in.releasegroups.keywords*.
- If a new nuclide release block was added to file *<in.releasegroups.structure>*.
- If changes were made in file *<in.FixDataNuclides>*.

12 Figures

Figure 1: Input for and output from program package STERM/UIINPUT

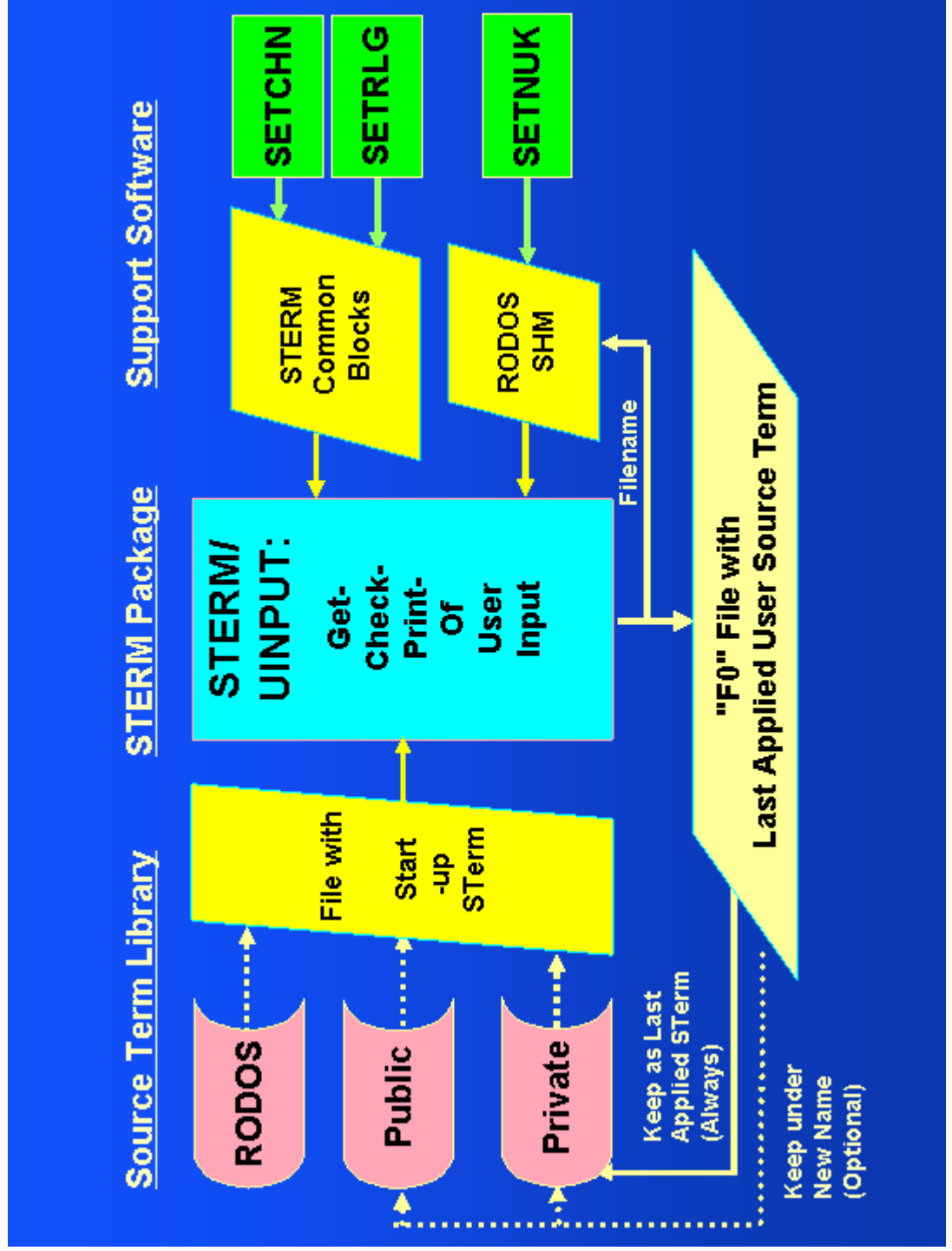
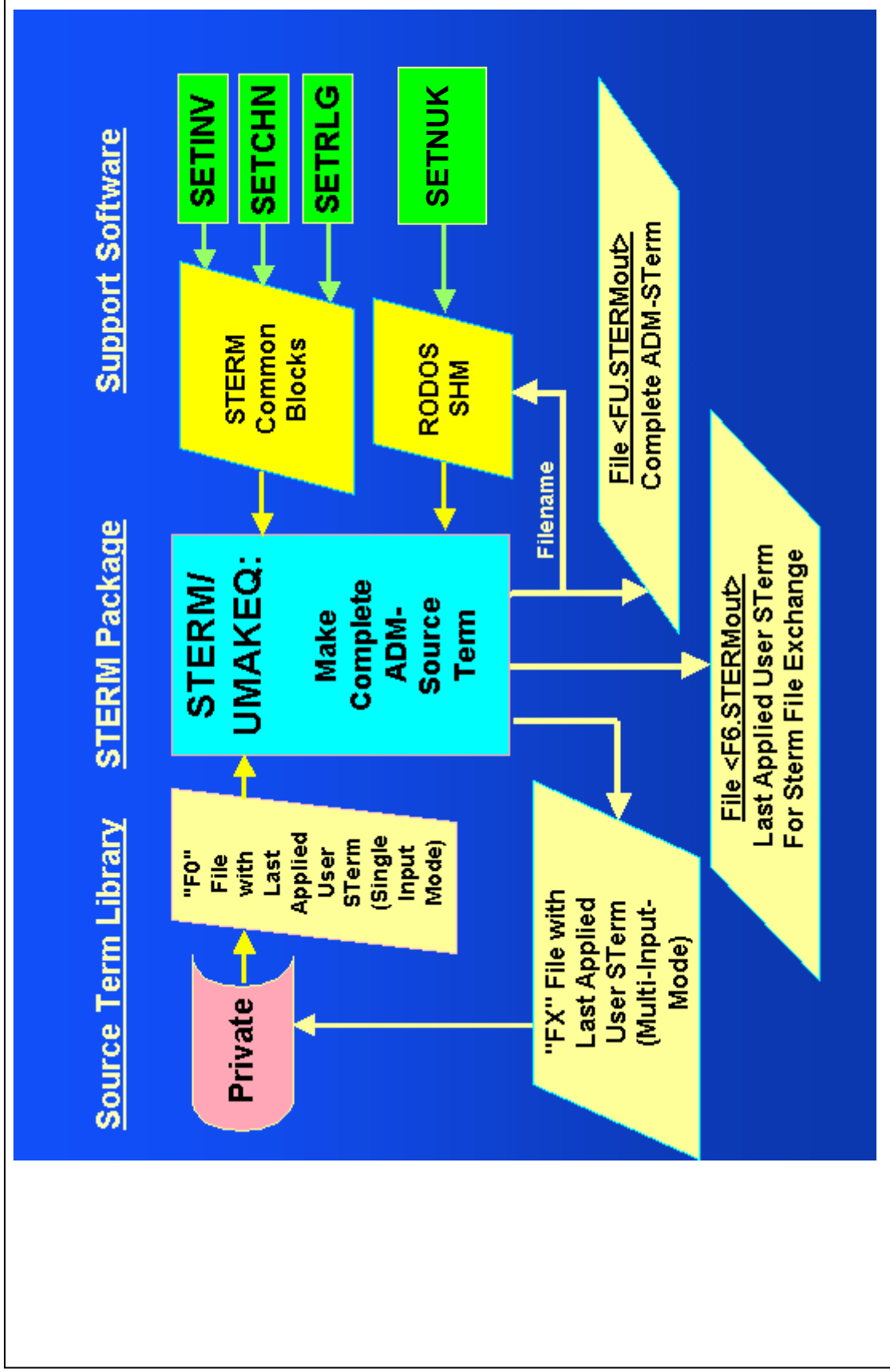


Figure 2: Input for and output from program package STERM/UMAKEQ



13 References

- [1] European Commission, J. Ehrhardt and A. Weis (ed.), RODOS: Decision support system for off-site nuclear emergency management in Europe, final project report. Report EUR 19144 EN, ISBN 92-828-9773-7, 2000.
- [2] C. Landman, Source Term Treatment for Nuclear Accidents in RODOS PV6final. Internal report RODOS(RA2)-TN(04)-04, Draft Version 2, 2007.
- [3] C. Landman, Documentation of the SourceInfo/I software package in RODOS-PV4.0F2. Internal report [RODOS(RA1)-TN(01)-01], 2001.
- [4] C. Landman, FixData nuclides, near range nuclides, and far range nuclides, in RODOS PV6final. Internal report RODOS(RA1)-TN(02)-04, Draft Version 2, 2007.
- [5] Irmgard Hasemann, Claudia Landman, Jürgen Päsler-Sauer, User Guide for the Prognosis Models of RODOS PV6final. RODOS internal report (no number), 2007.
- [6] Gerhard Benz, Irmgard Hasemann, RODOS User Guide: System Interface. RODOS internal report without number, 2004.
- [7] Antje Müller, Elvira Munz, Irmgard Hasemann, User Guide for the RODOS B-User Interface. RODOS internal report without number, 2004.
- [8] C. Landman, Mapping of data between user-defined and Near_ADM time intervals. Internal report RODOS(RA1)-TN(03)-01, 2003.
- [9] C. Landman, Near_ADM Source Term Interface STERM/UTOADM in RODOS PV6.0. Internal report RODOS(RA2)-TN(04)-04, 2004.
- [10] C. Landman, RODOS MODEM source term files (ASCII) for nuclear accidents in RODOS PV6final. Internal report RODOS(RA1)-TN(04)-01, Draft Version 2, 2007.
- [11] Leitfaden für den Fachberater Strahlenschutz der Katastrophenschutzleitung bei kerntechnischen Notfällen. Veröffentlichungen der SSK, Band 13, 2. Auflage, ISBN 3-437-11639-8, 1995.
- [12] R. D. Evans, The Atomic Nucleus. Mc.Graw-Hill Book Co., New York (1955), Tenth Printing September 30, 1965.
- [13] J. Ehrhardt, I. Hasemann, C. Landman: Summary of site specific data of European nuclear power plants as input to RODOS. Internal report DSSNET(WP03)-TN(01)-02, 2001.

- [14] C. Landman, Scenario data sets and scenarios for RODOS Version PV6. Internal report RODOS(RA7)-TN(04)-02, 2004.
- [15] D.C.Kocher: Radioactive Decay Data Tables. DOE/TIC-11026, U.S. Department of Energy, 1981.
- [16] Gesellschaft für Reaktorsicherheit: Deutsche Risikostudie Kernkraftwerke. Fachband 8, Unfallfolgerechnungen und Risikoergebnisse. Verlag TÜV Rheinland, ISBN 3-88585-019-2, 1981.
- [17] Gesellschaft für Reaktorsicherheit (GRS) mbH: Deutsche Risikostudie Kernkraftwerke, Phase B. Eine zusammenfassende Darstellung. GRS-72 (Juni 1989), ISBN 3-923875-22-3, 1989.
- [18] W. Raskob (editor), F. Fischer, I. Hasemann, C. Landman, Nicole Verhoef: Guidance on adaptation of RODOS to national conditions. Internal report RODOS(RA1)-TN(01)-03, 2007.

Document History

Document Title: User Source Term Interface STERM/USER(NPP) in RODOS
PV6final

RODOS number: RODOS(RA1)-TN(04)-03

Version and status: Version 1.0 (Draft)

Authors/Editors: C. Landman, Forschungszentrum Karlsruhe GmbH, Institut für
Kern- und Energietechnik, Postfach 3640, D-76021 Karlsruhe,
e-mail: claudia.landman@iket.fzk.de

Issued by: C. Landman

History: Follow-up of RODOS(RA1)-TN(04)-03 Draft Version 1

Date of Issue: May 2007

Circulation: all

File Name: QT_User_v200.doc

Date of print: August 23, 2007