

---

# Mapping of data between user-defined and Near\_ADM time intervals

---

**RODOS migration**  
contract

**F i n a l D R A F T**



**RODOS**  
REPORT

DECISION SUPPORT FOR NUCLEAR EMERGENCIES



**Mapping of data between user-defined and Near\_ADM time intervals**  
RODOS(RA1)-TN(03)-01

---

**Claudia Landman and Jürgen Päsler-Sauer**

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

Final Draft, 23 August 2007

## Management Summary<sup>1</sup>

From version PV5.0 onwards, RODOS allows for user input of time-dependent data describing release and meteorological conditions for a number of time intervals of variable duration. The RODOS atmospheric dispersion and deposition models for the near range (near\_ADM), however, need all that information in time intervals of fixed width.

Chapter 2 (J. Päsler-Sauer) describes the calculation of an average wind vector for each near-range ADM time interval from the wind speeds and directions given by the RODOS user in time intervals defined by the user.

Chapter 3 (C. Landman) describes the mapping with respect to the time variable between the user input side and the near-range ADM side for all source term input items and the remaining meteorology items.

---

<sup>1</sup> 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*

<b>1</b>	<b>MAPPING OF DATA BETWEEN USER-DEFINED AND NEAR_ADM TIME INTERVALS</b>	<b>3</b>
1	Introduction	7
2	Calculation of an "average wind"	8
3	Time mapping with UMAPIT	11
3.1	Denotations, time intervals, possible constellations	11
3.2	Time mapping of quantities behaving additive or rate-like	12
3.3	Time mapping of continuous quantities connected to a marker event	12
3.4	Time mapping of "obstinate" quantities	13
3.5	Summary of UMAPIT time mapping classes and subroutines	14
4	References	16



## 1 Introduction

The program system RODOS<sup>2</sup> [1] allows for user input of source term data in its interactive and automatic operation mode, and of meteorological data in its interactive operation mode. In earlier versions of RODOS such input could only be made directly for the coherent time intervals of fixed width dictated by the requirement of the RODOS atmospheric dispersion and deposition models for the near range (Near\_ADMs) [2]. Since version PV5.0 onwards the input can be made for a number of time intervals of variable duration defined by the user [3]. The RODOS ADMs, however, do still need as before both source term and meteorology data in coherent time intervals with fixed widths.

This new way of specifying the data has not only the consequence that the duration of the input time intervals is no longer fixed. Release input, for instance, can now also appear for intervals that are not coherent, because time periods not covered by a specification naturally imply that then a release does not take place. Meteorology input, however, can appear only for coherent time intervals, because a meteorological condition "no weather" does not exist. In addition, by technical reasons connected with the RODOS A and B user interfaces and the RODOS-Lite interface, the meteorological time intervals appear always ordered with respect to ascending time, whereas source term time intervals need not necessarily do so.

Chapter 2 (J. Päsler-Sauer) describes the calculation of an average wind vector for each near-range ADM time interval from the wind speeds and directions given by the RODOS user in time intervals defined by the user.

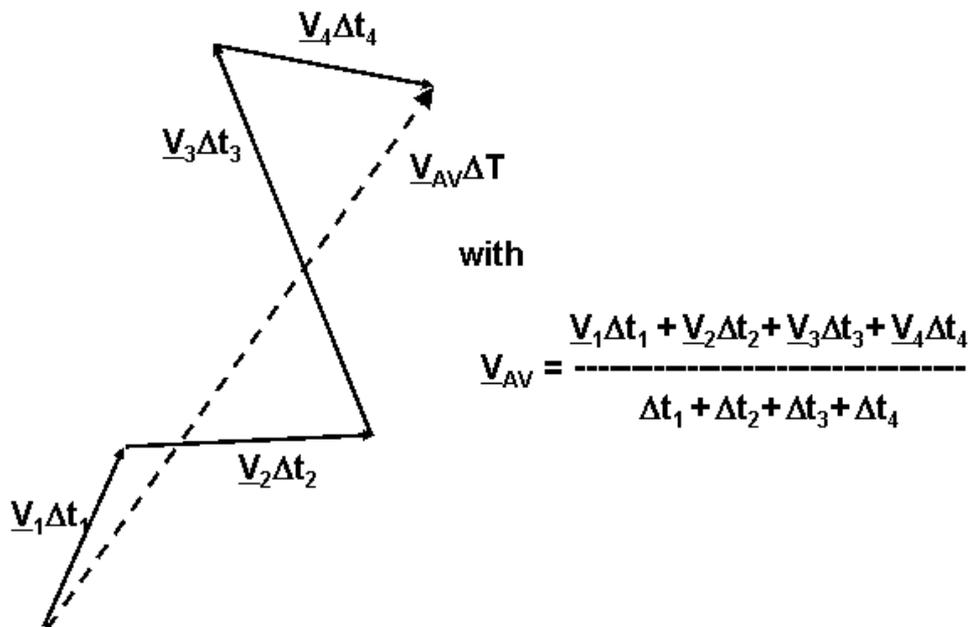
Chapter 3 (C. Landman) describes the mapping with respect to the time variable between the user input side and the near-range ADM side for all source term input and the remaining meteorology items with code package UMAPIT (**U**tility for **M**apping of **I**nput **T**imes) that was originally developed with the only aim to transform various data items from the input time representation to the Near\_ADM time representation. Later, UMAPIT was extended to support other applications such as the RODOS internal conversion of source term data [4].

---

<sup>2</sup> RODOS: **R**eal-time **O**nline **D**ecision Support System

## 2 Calculation of an "average wind"

Consider the transport of some airborne particle with varying wind vectors  $\underline{V}_i$  during a sequence of coherent time intervals  $\Delta t_i$ . The endpoint of the particle's path is obtained by adding the vector path components  $\underline{V}_i \Delta t_i$ . The average wind vector  $\underline{V}_{AV}$  represents a wind that would lead to the same endpoint when blowing with constant speed and direction over the total transport time  $\Delta T = \Delta t_1 + \Delta t_2 + \dots + \Delta t_n$ . The scheme below illustrates this for the case of two-dimensional transport over four time intervals.



In RODOS, the speed and direction of the average wind for each near\_ADM time interval is calculated with FORTRAN subroutine WINDAV from the user wind input in time intervals defined by the user.

WINDAV must be called for each near\_ADM time interval. Input to WINDAV is the number of user intervals covering the ADM interval under consideration, and, for each affected user time interval:

- The length of the part of the user interval lying in the ADM interval,  $\Delta t_i$ , [s].
- The wind speed in the user interval,  $|\underline{V}_i|$ , [m/s].

- The meteorological wind direction in the user interval,  $|\theta_{wi}|$ , [degree].

Output from WINDAV is the speed  $|\underline{V}_{AV}|$  and meteorological direction  $\theta_{WAV}$  of the "average wind in the ADM interval".

In meteorological terms, wind is described by the wind speed  $v$  and the angle  $\theta_w$  describing direction from where the wind blows. The WINDAV calculations are carried out with wind vectors in a two-dimensional x-y Cartesian co-ordinate representations in a sequence of steps described below.

### **Step 1:**

#### **Transport wind direction from meteorological wind direction**

The meteorological wind direction  $\theta_w$  is the direction from where the wind blows ( $\theta_w=0^0 \Leftrightarrow$  north,  $\theta_w=90^0 \Leftrightarrow$  east,  $\theta_w=180^0 \Leftrightarrow$  south,  $\theta_w=270^0 \Leftrightarrow$  west). The direction of the mathematical wind vector, the "transport direction",  $\theta_v$ , is the direction into which the wind blows, and is obtained from  $\theta_w$  with Equation 2-1:

$$\theta_v = \theta_w - 180^0 \quad \text{(Equation 2-1)}$$

If  $\theta_v$  is negative replace it by  $\theta_v + 360^0$ .

### **Step 2:**

#### **Mathematical wind vector from meteorological wind**

The mathematical wind vector  $\underline{V}$  in a 2-dimensional Cartesian x-y coordinate system where the +y-axis pointing to North is described with Equation 2-2:

$$v_x = |\underline{V}| \cos\theta_v$$

$$v_y = |\underline{V}| \sin\theta_v$$

*(Equation 2-2)*

### **Step 3:**

#### **Calculation of speed of average wind**

The x- and y- components of the average wind are obtained in the following way:

$$\langle v_x \rangle = (v_{x1} \Delta t_1 + v_{x2} \Delta t_2 + \dots + v_{xn} \Delta t_n) / \Delta T$$

$$\langle v_y \rangle = (v_{y1} \Delta t_1 + v_{y2} \Delta t_2 + \dots + v_{yn} \Delta t_n) / \Delta T,$$

$$\text{with } \Delta T = \Delta t_1 + \Delta t_2 + \dots + \Delta t_n$$

The resulting average wind speed  $|\underline{V}_{AV}|$  is:

$$|\underline{V}_{AV}| = \text{Squareroot}(\langle v_x \rangle^2 + \langle v_y \rangle^2)$$

*(Equation 2-3)*

**Step 4:**

**Calculation of speed of average wind**

There are several mathematical possibilities to calculate the mathematical angle  $\theta_V$ . WINDAV uses Equation 2-4:

$$\theta_V = \arcsin(\langle v_y \rangle / |\underline{V}_{AV}|) * 180^0 / \pi$$

*(Equation 2-4)*

**Step 5:**

**Meteorological wind direction from transport wind direction**

Finally, the meteorological direction of the average wind is obtained by

$$\theta_{WAV} = \theta_V + 180^0 \quad \text{(Equation 2-5)}$$

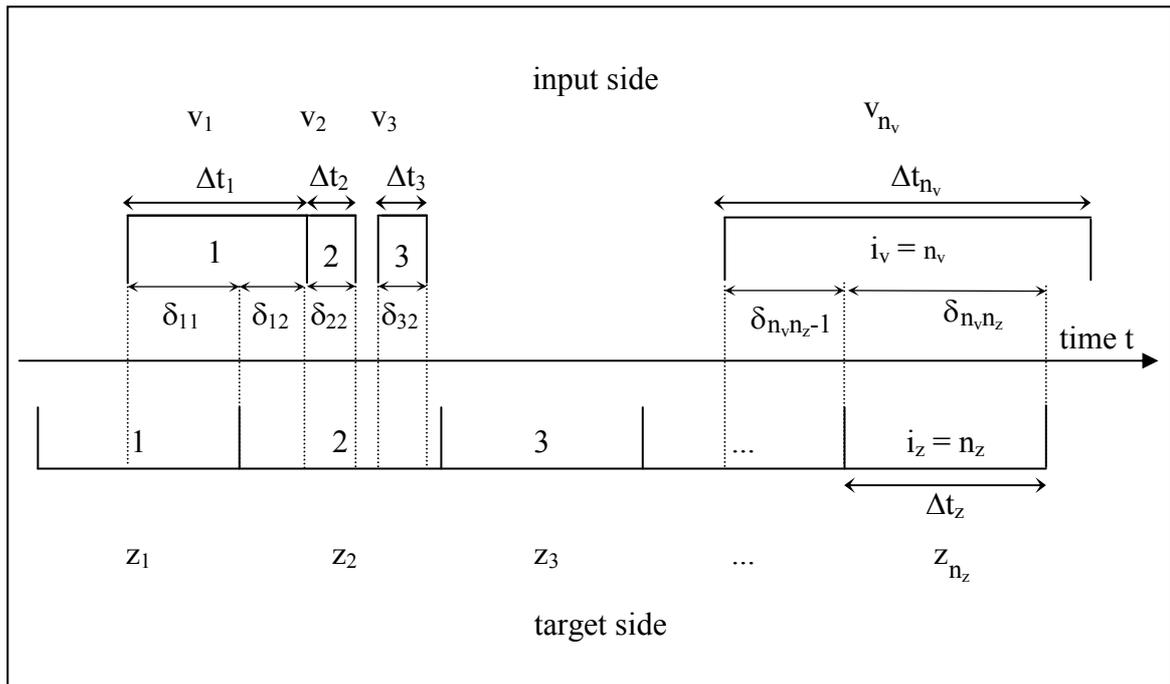
If  $\theta_{WAV}$  is greater or equal  $360^0$  replace it by  $\theta_{WAV} - 360^0$ .

### 3 Time mapping with UMAPIT

#### 3.1 Denotations, time intervals, possible constellations

Figure 1 shows by means of an illustrative example the denotations used in this document for the time intervals and other quantities on the input side and the target side of UMAPIT.

**Figure 1: Illustration of denotations and time intervals**



Character  $t$  represents the time variable. The input side is distinguished by character  $v$ , and the target side by character  $z$ .

On the target side, there are  $i_z = 1 \dots n_z$  coherent and ascending time intervals of fixed width  $\Delta t_z^3$ .

On the input side, there are  $i_v = 1 \dots n_v$  time intervals with, in general, non-equal widths  $\Delta t_{i_v}$ , the boundaries of which must not overlap. For each interval  $i_v$ , the value  $v(i_v)$  of some physical quantity  $q$  is given that holds for the entire interval. The input intervals can but need not be coherent. Also, they need not to appear ordered with respect to ascending time, which means, an interval with a higher index  $i_v$  can have time boundaries preceding those of an interval with a lower index. For the sake of simplicity, however, in Figure 1 the input intervals are shown as being ordered with time.

<sup>3</sup> UMAPIT contains also subroutines for target intervals with non-equal widths  $\Delta t_{i_z}$ .

On both sides, interspersed intervals with zero or negative widths do no harm because they are simply ignored in UMAPIT.

Looking from the target side to the input side, there can be target intervals that are covered by more than one input specification; in the example, this is the case for  $i_z=2$ . Or, a target interval is covered by only one input specification, either partly (e.g. for  $i_z=1$ ) or fully (for  $i_z=n_z$ ). There can also be target intervals not covered by any input specification (for  $i_z=3$ ).

If there is at least one input specification covering a target interval  $i_z$ , the value  $z(i_z)$  of quantity  $q$  in the target side time representation is provided by UMAPIT under one of the assumptions explained in the following Chapters 3.2 to 3.4. If there is no specification,  $z(i_z)$  is left untouched at whatever value it may have had before the call to UMAPIT.

### 3.2 Time mapping of quantities behaving additive or rate-like

Some of the RODOS user input variables represent a physical quantity that behaves additive with respect to time, as for instance the total amount of activity released in a time interval. For such a quantity, the value  $z$  for a given target time interval  $i_z$  is obtained by summing up all contributions of  $v$  to the target interval:

$$z(i_z) = \sum_{i_v} (\delta(i_v, i_z) / \Delta t_{i_v}) * v_{i_v} \quad \text{(Equation 3-1)}$$

Value  $z_2$  in Figure 1 obtained with viewpoint of Equation 3-1 would be:  $z_2 \equiv z(i_z=2) = (\delta_{12}/\Delta t_1) * v_1 + (\delta_{22}/\Delta t_2) * v_2 + (\delta_{32}/\Delta t_2) * v_3$ .

If one divides an additive quantity given in a time interval by the interval duration, the resulting value describes a rate that is constant in the time interval. By interpreting the values  $v(i_v)$  as such rates, the value  $z$  for a target interval  $i_z$  can be derived by summing up all integrated contributions of  $v$  to the target interval, and dividing the resulting sum by the target interval duration  $\Delta t_z$  to obtain a rate again:

$$z(i_z) = [\sum_{i_v} (\delta(i_v, i_z) * v_{i_v})] / \Delta t_z \quad \text{(Equation 3-2)}$$

Value  $z_2$  in Figure 1 obtained with viewpoint of Equation 3-2 would be:  $z_2 \equiv z(i_z=2) = (\delta_{12} * v_1 + \delta_{22} * v_2 + \delta_{32} * v_3) / \Delta t_z$ .

### 3.3 Time mapping of continuous quantities connected to a marker event

Some physical quantities are connected to a special "marker" event in the sense that without that event they lose their meaning. Such quantities do not allow straightforward averaging over time periods with and without the marker event. An example is the release height. When an activity release is going on, the release height has some

definite value  $\geq 0$  m. However, when there is no release, there is simply no release height, and not a release height of 0 m.

If such variables are continuous, weighted averages can be calculated for a target time interval  $i_z$ ; however, all portions of the target interval that are not covered by an input specification must be thereby excluded:

$$z(i_z) = [\sum_{i_v} (\delta(i_v i_z) * v_{i_v})] / \sum_{i_v} \delta(i_v i_z) \tag{Equation 3-3}$$

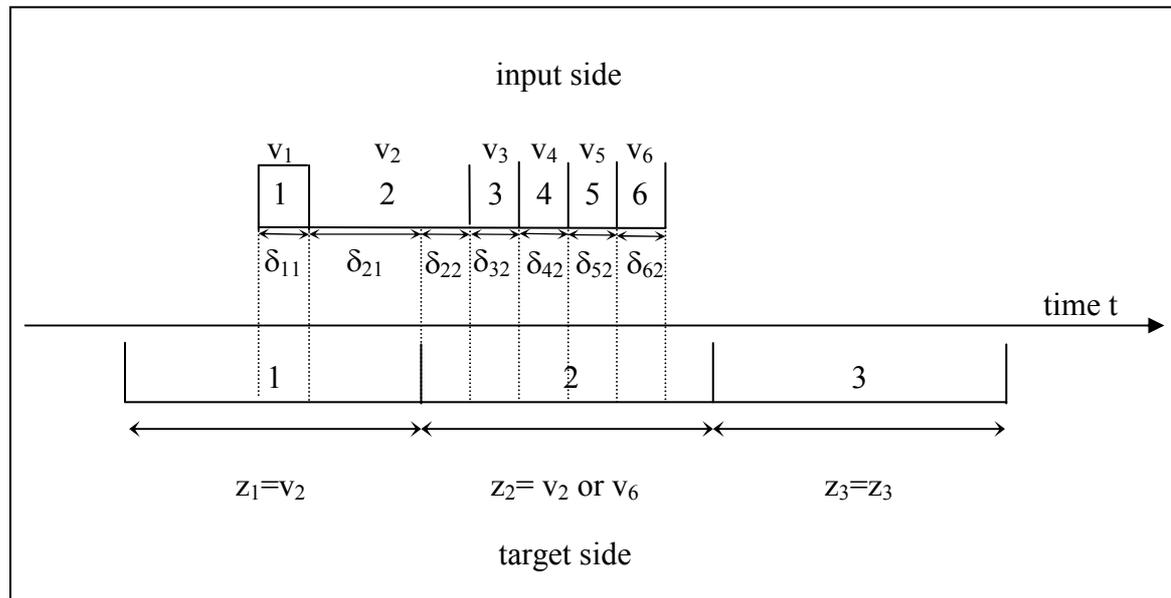
Value  $z_2$  in Figure 1 obtained with viewpoint of Equation 3-3 would be:  $z_2 \equiv z(i_z=2) = (\delta_{12} * v_1 + \delta_{22} * v_2 + \delta_{32} * v_3) / (\delta_{12} + \delta_{22} + \delta_{32})$

### 3.4 Time mapping of "obstinate" quantities

Last but not least, there are quantities that defy an easy treatment with respect to the time mapping because of mathematical or conceptual problems, for example the "Pasquill-Gifford diffusion category".

For tackling the time mapping of such "obstinate" quantities, UMAPIT contains a subroutine UMAPU which uses a simple and straightforward approach illustrated by Figure 2 (explanations follow after the Figure).

**Figure 2: Example for time mapping of an "obstinate" quantity**



Each target interval  $i_z$  gets as value  $z(i_z)$  the value  $v$  of that input time interval  $i_v$  which occupies the largest portion  $\delta(i_v i_z)$  of the target time interval. In case of ambiguities, either the first or the last encountered candidate with equal  $\delta(i_v i_z)$  is taken; this can be steered by the caller of UMAPU. Subroutine UMAPU is currently limited to coherent input

intervals ascending with time  $t$ , and to coherent target intervals with ascending order in  $t$  and equal width.

In the example in Figure 2, the value of variable  $z$  in target interval  $i_z=1$  will be  $v(i_v=2)$ , and in target interval  $i_z=2$  either  $v(i_v=2)$  or  $v(i_v=6)$ , depending on the steering flag mentioned above. As target interval  $i_z=3$  is not covered by any input specification,  $z(i_z=3)$  will contain whatever value it had before the call to UMAPU.

### 3.5 Summary of UMAPIT time mapping classes and subroutines

Code Package UMAPIT contains subroutines for the four variable types with respect to the time mapping behaviour that were described in Chapters 3.2 to 3.4. Table 1 lists the classes and relates all RODOS user source term and meteorology quantities to the classes.

**Table 1: UMAPIT time mapping classes**

UMAPIT class	Behaviour with respect to time mapping	RODOS variable
<vi>	Quantity <vi> represents an additive quantity in each time interval. For time intervals without specification for <vi>, the value is physically existent and zero.	Activity release in percent (STerm) Activity release (Bq)
<vr>	Quantity <vr> represents a rate of an additive quantity in each time interval. For time intervals without specification for <vr>, the value is physically existent and zero.	(Meteo) Rain rate (mm/h) (STerm) Released thermal power (MW) (STerm) Vert. released vol.flux (m**3/s] (STerm) Activity release rate (Bq/s)
<vm>	Quantity of type <vm> represents a continuous quantity connected to some "marker event". For time intervals with a non-zero value of the marker event, the value of <vm> may be zero. For time intervals with a zero value of the marker event, the value of <vm> is undefined.	(STerm) Release height (m). <u>Marker event:</u> Activity release. (STerm) Vent. area of release (m**2). <u>Marker event:</u> Activity release. (STerm) Relative iodine fractions. <u>Marker event:</u> Iodine release.
<vu>	Unknown, not easily definable, or difficult to handle.	(Meteo) Pasquill-Gifford diffusion categ.

Table 2 shows all UMPIT subroutines available for the different time mapping classes together with their input requirements.

All subroutines are written in FORTRAN and completely independent from the RODOS environment. The communication between caller and subroutine takes place via argument lists. The dimensions of the arrays are inferred from the caller. In the header section of each subroutine, detailed instructions are given about the call and the arguments.

Subroutines marked "1D" are for the mapping of 1-dimensional FORTRAN arrays, those marked "2-D" for 2-dimensional arrays. Mapping of FORTRAN arrays with more dimensions can be achieved by calling the 1D-subroutine separately for each array dimension.

The subroutines for continuous quantities with connection to a marker event must first be initialised for the marker event in a separate call. After that, they can be used for the time mapping for all quantities referring to this marker.

**Table 2: Subroutines in code package UMAPIT**

Class	Subroutine	Input requirements (I: Input side, T: Target side) <sup>a)b)</sup>
<vi>	UMAPXI_1D UMAPXI_2D	<ul style="list-style-type: none"> <li>Time intervals: I: No constraint on width, order with time, coherence. T: Equal width. Ascending with time. Coherent.</li> <li>Target result array(s) z: Must be set to zero prior to call by caller.</li> </ul>
<vr>	UMAPXR_1D UMAPXR_2D	
<vm>	UMAPXM_1D	
<vu>	UMAPU	<ul style="list-style-type: none"> <li>Time intervals: I: No constraint on width. Ascending with time. Coherent. T: Equal width. Ascending with time. Coherent.</li> <li>Target result array(s) z: Before calling UMAPU make either sure that the input specifications cover the entire target time range. Or, alternatively, set the target value array to whatever value is reasonable for your problem in the case "target interval not covered by an input specification" (see Chapter 3.4).</li> </ul>
<p><b>Notes:</b></p> <p>a) The overlap of interval boundaries is not allowed. This constraint is not checked in UMAPIT and must be safeguarded by the caller!</p> <p>b) Interspersed intervals with zero or negative widths are skipped in UMAPIT, that is, they do no harm.</p>		

## 4 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 (2000), ISBN 92-828-9773-7.
- [2] C. Landman, J. Päsler-Sauer: User Guide for the Prognosis Model RODOS-PV4.0F02. RODOS internal report (no number).
- [3] I. Hasemann, C. Landman, J. Päsler-Sauer: User Guide for the Prognosis Models of RODOS-PV5.0. RODOS internal report (no number).
- [4] C. Landman: Source Term Treatment in RODOS PV6final. RODOS internal report RODOS(RA2)-TN(04)-04, 2004.

## Document History

Document Title: Mapping of data between user-defined and Near\_ADM time intervals  
RODOS number: RODOS(RA1)-TN(03)-01  
Version and status: Final 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:  
Date of Issue: May 2007  
Circulation: all  
File Name: UMAPIT.doc  
Date of print: August 23, 2007