

SCIENTIFIC SOFTWARE & MODELLING SOLUTIONS

# AMBER 4.5 Release Note

---

**Version 1.00e**

**April 2003**

Enviros Consulting Limited  
Building D5 • Culham Science Centre  
Culham • Oxfordshire  
OX14 3DB  
United Kingdom  
Phone (44) 1235 468800  
Fax (44) 1235 468828  
Email [AMBER@enviros.com](mailto:AMBER@enviros.com)  
[www.enviros.com/amber](http://www.enviros.com/amber)

Quintessa Limited  
Dalton House • Newtown Road  
Henley-on-Thames • Oxfordshire  
RG9 1HG  
United Kingdom  
Phone (44) 1491 636246  
Fax (44) 1491 636247  
email [AMBER@quintessa.org](mailto:AMBER@quintessa.org)  
[www.quintessa.org/amber](http://www.quintessa.org/amber)

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Enviros Consulting Limited.

© Enviros Consulting Limited 2003. All rights reserved



# Table of Contents

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>2.</b>	<b>IMPORT/EXPORT.....</b>	<b>3</b>
2.1	<i>Introduction.....</i>	3
2.2	<i>Potential AMBER Uses for a Time-series Data File.....</i>	4
2.3	<i>File Format.....</i>	6
2.4	<i>Creating the File from AMBER.....</i>	11
2.5	<i>Importing the File to AMBER.....</i>	13
<b>3.</b>	<b>PARAMETER COPY AND PASTE .....</b>	<b>21</b>
3.1	<i>Parameters from file.....</i>	22
<b>4.</b>	<b>OTHER NEW FEATURES.....</b>	<b>27</b>
4.1	<i>New Units .....</i>	27
4.2	<i>Switch Transfers On/Off.....</i>	27
4.3	<i>Auto-save after Calculate .....</i>	28
4.4	<i>Search by Indexing .....</i>	28
<b>5.</b>	<b>BUG FIXES AND MINOR CHANGES .....</b>	<b>29</b>
<b>6.</b>	<b>UNDOCUMENTED FEATURES AND FUTURE DEVELOPMENTS.....</b>	<b>35</b>
6.1	<i>Undocumented Features.....</i>	35
6.2	<i>Future Developments.....</i>	36



# 1. Introduction

In AMBER Version 4.5 the capabilities of the program have been extended in a number of ways.

The changes that have been implemented are:

- ▲ Import/Export;
- ▲ Copy/Paste Parameters;
- ▲ New units through GUI;
- ▲ Switch on/off transfers;
- ▲ Auto-save;
- ▲ Search according to index;
- ▲ Bug fixes; and
- ▲ Other minor changes (e.g. receptor mapping).

The following Sections describe these changes in more detail.



## 2. Import/Export

### 2.1 Introduction

A major new feature in Version 4.5 is the ability to Import and Export data between AMBER and external files. In Version 4.4, AMBER did not allow one calculated result to depend on another except through limited built-in relationships (e.g. solubility limitation). In some situations it would be useful to be able to calculate some results and then use these in defining parameters for subsequent calculations. In order to make this new capability as flexible as possible, the following approach has been taken.

- Define a file format for storing time-series data for multi-indexed parameters (designed to be easy to create, e.g. by copying data from a spreadsheet);
- Enable AMBER to output parameters in the defined format;
- Enable AMBER to use files in the defined format to provide values for parameters that could then be used as normal AMBER parameters.

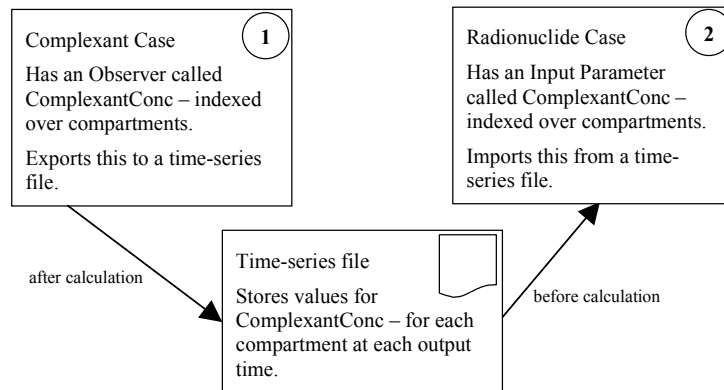
This note therefore discusses: the file format; the AMBER GUI aspects for creating such files; and the AMBER GUI aspects for using such files – including checking for compatibility.

In order to motivate these aspects, an initial discussion is presented of where this new capability could be used.

## 2.2 Potential AMBER Uses for a Time-series Data File

The initial motivation arose from a requirement to use a calculated complexant concentration in defining properties (e.g.  $K_d$ ) for radionuclide transport. Since the complexant calculation does not depend on the radionuclide calculation, the calculations can be performed in sequence with the results from the first passed onto the second.

This is illustrated in the diagram below.



This clearly generalises to any result that can be calculated on the same structure (compartments/transfers) and used subsequently.

By allowing the Time-series file to be generated independently of AMBER, other possibilities are available.

Some possible uses are:

- calculation of a concentration field for one or more “chemicals” that subsequently influence parameter values for a contaminant transport calculation;

- specification of source fluxes in a more convenient way than currently possible;
- specification of disposal history;
- saving the state of the system (amounts) at some time to allow restart (e.g. with different subsequent behaviour and less nuclides or no time dependence or to compare two different futures without recalculating short-term behaviour);
- separating source, geosphere and biosphere parts of an AMBER case (maybe for efficiency, or to allow different models to be interchanged);
- calculating a source flux for input to another program and then using its output in an AMBER biosphere model;
- reading old results and comparing them with new ones;
- comparing against external results for verification;
- taking a flow regime in one AMBER case file for use in another. Although AMBER is not designed to solve for water flows, the equations for transient flow are of the same structure as diffusion equations and so this could in principle be done;
- for weakly non-linear situations, if a parameter depends on the solution, then an iterative scheme could be used by saving the results and re-inputting them in the next iteration.

One issue that arises from the list above is the units of the “Amount” variable. Currently, this can only be chosen from a restricted list (mol, kg, Bq). In some cases (e.g. the flow calculation but also possibly for aspects of the chemical environment) it would be convenient to allow the user to specify the units directly. In cases where decays are involved and the user had specified units (if such a case ever arises) the solver would have to assume that one unit of parent decays to one unit of daughter (as for the mole unit rather

than the Bq unit). This feature has not been included in the current version.

### **2.3 File Format**

This section describes the file format. Users who wish only to use AMBER to create and re-read these files need not be aware of this detail. The AMBER file extension for these files is .aaf (AMBER ASCII file).

There are lots of “standard” file formats around (e.g. CDF, PDS, FITS), but they all tend to be specialised to a particular area (oceanography, astrophysics...). Also, most use a binary file with supplied software to read and write it. We want a simple format that is easy to use – ideally easy to make from a spreadsheet form if required (possibly with the help of specially created macros) or from file-generation applications (e.g. awk).

The approach that has been used is as follows:

- An ASCII file format is used.
- Data items are separated by white space (e.g. a single space).
- Each file may contain several parameters, so that situations where several outputs belong together (e.g. concentrations of several complexants) can be handled in a convenient fashion.
- Only deterministic results can be saved – otherwise the files become too large, moreover the question of how to associate one set of sampled results with another case is unclear.
- Although the usual case is for time-series, static values are catered for.
- Arrays (1D or 2D) or scalar values may be output. There is no single natural layout of data values for

2D arrays and so several layout styles have been allowed.

- In order to ensure proper connection between the data items, it is necessary to store array index information (NameSet item names). Situations where the item list includes unwanted items or where it has missing items have been catered for.
- To protect against future extensions, a version number is included in the file.
- It is possible to store (optional) descriptive text.

So, a file could look like the following.

VERSION: 1 (optional – assumes current version if omitted – first line of file if present)  
 NAME: ComplexantConcentration (first line for each parameter in the file)  
 DESCRIPTION: “Text” (optional – multiline - quote delimited)  
 TYPE: TimeSeries (other choice is Constant)  
 DIMENSIONS: 1 (or 0 or 2)  
 DIMENSION 1 NAME: Compartments  
 DIMENSION 1 SIZE: 42  
 DIMENSION 1 ITEMS: comp1 comp2 ... (whitespace separated list)  
 (same for DIMENSION 2 if relevant)  
 UNITS: kg/m<sup>3</sup> (optional – dimensionless otherwise)  
 NUMBER OF TIMES: 56  
 TIME UNITS: y (optional – dimensionless otherwise)  
 LAYOUT: TV\_TV (choices are discussed later)  
 VALUES-BLOCK (organised according to the layout given)

Note that the format of the header information must be strictly followed for the file to be read back correctly.

The layout of the values block depends on dimensionality. The simplest format would be to have each line as a whitespace-separated list:

```
time value1 value2 ...
```

with the number of values being the product of the dimension sizes and the order for the two-dimensional case being (1,1), (1,2), (1,3) ... (1,N1), (2,1), ...

The only problem with this is that it can lead to very long lines. This can upset some text editors and Excel cannot cope with more than 256 columns. The obvious alternative is to limit the number of values on each line, and so use multiple lines for each time. This has the disadvantage of being harder to follow and to use once imported into Excel.

Another approach would be to output one line per item in the first dimension. In fact, several different layout styles are useful, so the file format supports them all.

The following subsection discusses how this is handled.

### **2.3.1 Allowed Layouts**

Different considerations apply to static (constant) data and to time-series data and the way scalar data is handled is different from 1D and 2D data. There are six combinations to consider. Table 1 gives the valid layouts. The notation  $t_k$  is used to indicate the  $k$ -th time,  $v_m(t_k)$  means the value of the  $m$ -th item in a 1D array, at time  $t_k$  (the  $(t_k)$  is omitted for constants or when it is obvious which time is referred to).  $v_{mn}(t_k)$  means the value of  $(m,n)$ -th item in a 2D array, where the  $m$  refers to dimension 1 and the  $n$  to dimension 2.

Note that for the purposes of reading the files end-of-lines and spaces can be used interchangeably as separators – the choice being made for readability.

For each possibility, a layout name is given – this is intended to reflect the ordering so as to make it easy to remember. The underlined choice is the default for the combination of time-dependency and dimension (i.e. the one that AMBER will use for writing). Note that AMBER does not allow a selection of the output layout through the GUI (but it can be specified in the case file).

Note that in cases where the time is repeated, AMBER re-reads the times and does not check that they are the same. The times must be monotonically increasing.

**Table 1 Allowed Layouts for Time-Series and Static Data**

	Constant	Time-series	
<b>Scalar</b>	Only one value so the block simply reads: v V	Either time then value: t1 v(t1) t2 v(t2) ... TV	Or all times and all values: t1 t2 ... v(t1) v(t2) ...  T^T_VV
<b>1D</b>	List of values, so the block simply reads: v1 v2 ... VV	Either time then all values: t1 v1(t1) v2(t1) ... t2 v1(t2) v2(t2) ... ... TVV	Or time repeated before each value: t1 v1(t1) t1 v2(t1) ... t2 v1(t2) t2 v2(t2) ... TV_TV
<b>2D</b>	Either dimension 1 is held in blocks: v11 v12 v13 ... v21 v22 ... ... VV_12A* or VV_12  Or dimension 2 is held in blocks: v11 v21 v31 ... v12 v22 ... ... VV_21A or VV_21	The following have dimension 1 held - the versions with dimension 2 held exist too:  Simplest: t1 v11(t1) v12 v13 ... v21 v22 ... ... t2 v11(t2) v12 v13 ... v21 v22 ... ... TVV_12A* or TVV_12  Or all values for a block together: t1 v11(t1) v12 v13 ... t2 v11(t2) v12 v13 ... ... t1 v21(t1) v22 ... ... TVV_12B*	Or with time repeated each row: t1 v11(t1) v12 v13 ... t1 v21 v22 ... ... t2 v11(t2) v12 v13 ... t2 v21 v22 ... ... TVV_12C*  Or the "21" versions: TVV_21, TVV_21A, ... TVV_21B, TVV_21C

\* the 12 here means that dimension 1 is held first, but also that v12 is the second value written. (The 'A' version has a line break at the end of each row – when reading back this is irrelevant.)

Notes:

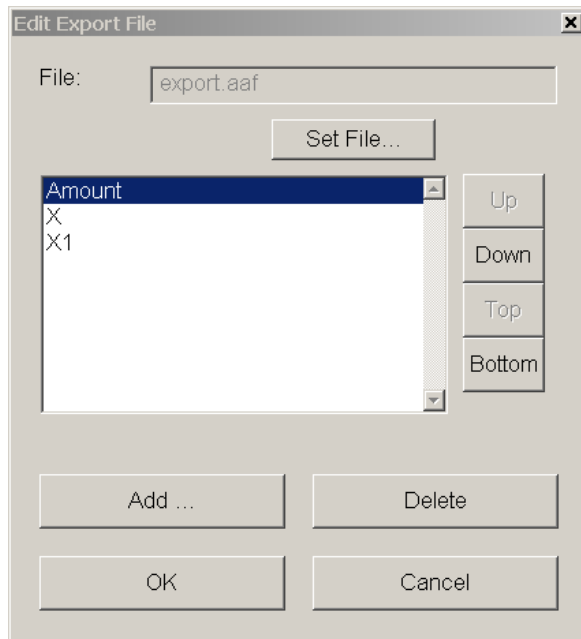
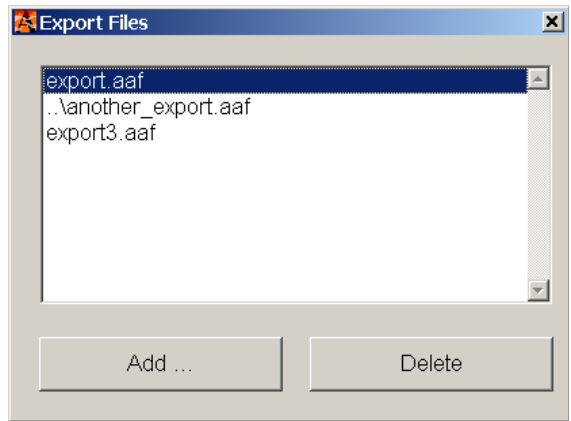
## 2.4 *Creating the File from AMBER*

The simplest approach would be that the file is created by a specific user action (rather than automatically). However, this would mean that if a case is recalculated the file must be re-exported by the user. Thus, an automatic creation has been used, so the file is rewritten after a new calculation, when the case is saved (i.e. at the same time that the adf file is saved).

It is useful to have a standard file extension, to make it easy to recognise these files. The AMBER binary results file is denoted “.adf” for AMBER data file, which might have been a good name for the new files. Instead they will be denoted as “.aaf” – AMBER ASCII file.

The GUI aspect is quite simple – an extra item on the Results menu to “Export Files...” starts the process. This brings up the Export Files Window, which lists the currently defined files.

The Add button allows new files to be defined and double-clicking on a file allows its details to be changed. Either action brings up the Export File Dialog which allows the file name and parameter information to be specified.



The Add button on the dialog allows new parameters to be added. AMBER does not allow a choice of layout, so only the parameter name is required. The file name is generally stored relative to the case file's directory, but may be stored as a full path if it is not below that directory. A prompt will

be issued asking whether a relative or full path name is to be used.

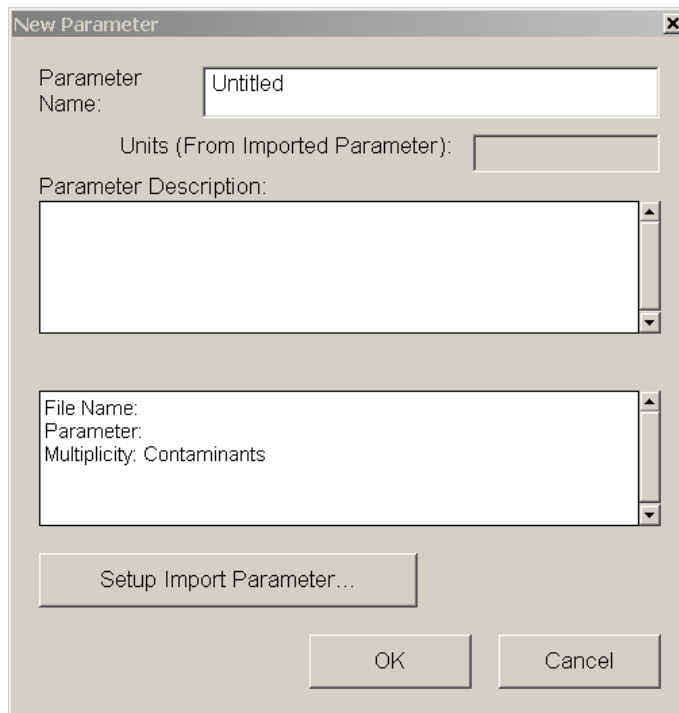
## 2.5 *Importing the File to AMBER*

For importing into AMBER, the file and parameter information is setup on an “Import Parameter”. The actual data is read when the parameter is created (or when the case file is opened) but may be re-read if any relevant features change (such as changes to items in a NameSet that is used).

Once imported, time-series data are linearly interpolated (without any consideration of discontinuities). The user can use SnapShots in the original case to add extra output times at key points. For times outside the range in the imported file, a value of zero is used.

To add a new Import Parameter, use the Add button on the Parameters window and select Import Parameter (note that the From File choice is completely separate).

The multiplicity of the parameter must be selected first. The number of dimensions must be consistent with what is being imported, but the NameSet details may differ (this is set up later). In particular, the order of the dimensions for a 2D parameter is not important. Once the multiplicity has been selected, the Import Parameter Dialog is displayed.



The parameter name is the name in this case file, which may be different from the name in the import file. The units cannot be set, but will be taken from the import file. A description can be entered here, but most of the information required is specified via a “Wizard” that is invoked by the “Setup Import Parameter” button. A summary of the information is displayed in the lower text box.

The wizard can also be used to edit information on an existing Import Parameter, after the dialog is displayed by double-clicking on the Import Parameter name in the Parameter Window. Note that the information is not actually saved until the OK button is pressed on the dialog. When this is done, the parameter is created or modified and the import file data is read (or re-read). Just opening the

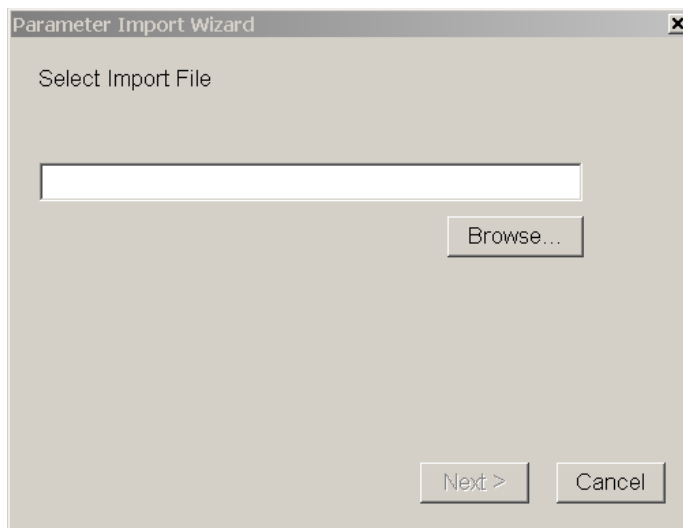
dialog and pressing OK forces the file to be re-read and so can be used in cases where the import file has been changed.

The Wizard goes through the necessary steps to set up the Import Parameter:

- Selecting the file
- Selecting the parameter from the file
- Matching the NameSets (if 2D)
- Matching NameSet items in the case to those in the import file
- Confirming the information.

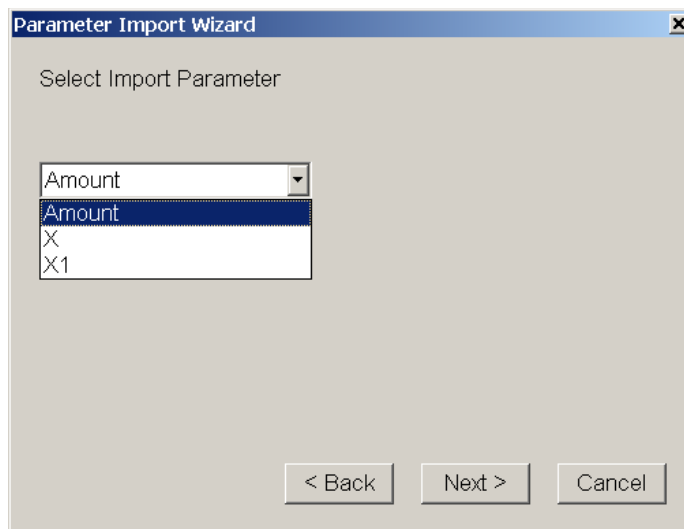
On each step, checks are made on the consistency of the information.

The first step is to provide the filename. This can be typed in or found through browsing. The Next button is inactive until a filename is filled in.



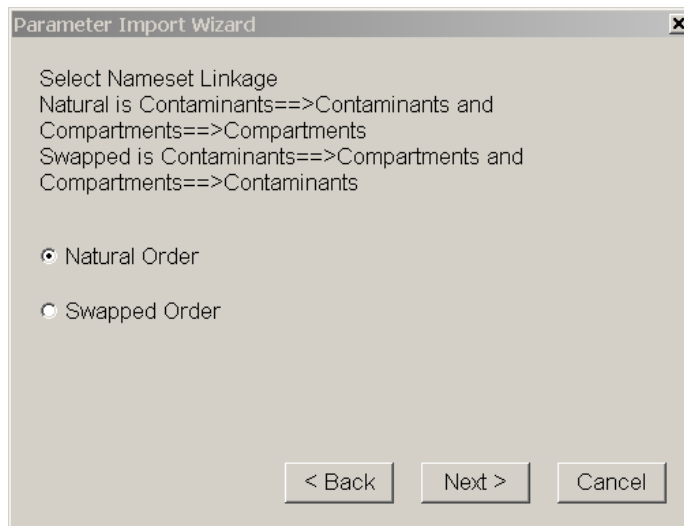
On pressing Next, the file contents are checked to see that this is an Import File.

The next panel allows the particular parameter in the file to be selected:



Having selected the required parameter, pressing the Next button will check that this has the correct number of dimensions. Note that the Back button can be used at any time to go to the previous panel and the Cancel button quits the Wizard without saving any information.

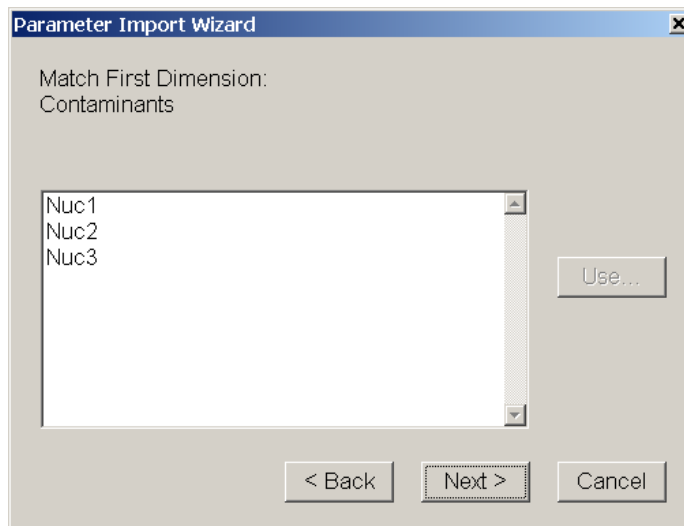
For a 2D parameter, the next panel is to confirm which namesets in the file correspond to those in the open case. Where the names match, the “natural order” will be the obvious choice. Indeed, this and the subsequent panels can be ignored if the file and case namesets match precisely.



The next panel, which is also arrived at for a 1D parameter, is to confirm which nameset item in the file is to be used to provide data for each nameset item in the case.

Each item in the case is listed. If there is no comment after the name, then an exact match exists in the file and no action is necessary.

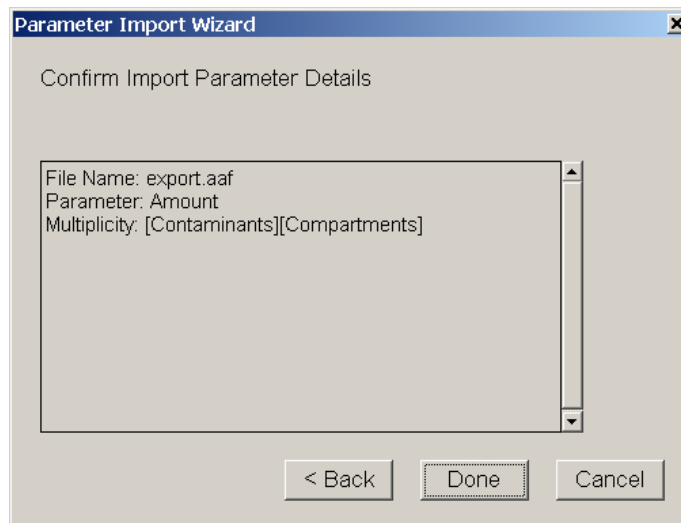
Notes:



Two other situations are possible. If the name reads Item (unmatched), then no item of the same name exists in the file (and no match has been specified). If the name reads Item (use OtherName), then the data will be obtained from the OtherName item in the file. In either case, selecting the name and pressing “Use...” allows the match to be changed. When the data is read, unmatched items will have a value of zero.

For a 2D parameter, a second similar panel is displayed for the second dimension.

Finally, a confirmation screen is displayed.



To accept the specification press “Done”, otherwise go Back or Cancel.

Once the data has been imported, it is kept unless a change occurs to one of the namesets that could invalidate it.

If a relevant NameSet is changed then the following actions will occur:

- if a NameSet is deleted, then the Import Parameter is deleted (this is the same as for any other parameter indexed over it);
- if a NameSet is renamed, then the linkages to the file are automatically updated and the data remain valid;
- if a NameSet item is deleted or renamed, or a new item is added, or the order changes, then the Import Parameter data is marked as out of date and will be re-read when required.

Note that if a nameset item is renamed so that the new name is not in the file, then the Wizard will need to be used to set

## AMBER 4.5 Release Note

up the match. If, however, the new item name is in the file then the data will automatically be matched.

## 3. Parameter Copy and Paste

In Version 4.5, parameters can be copied and pasted within or between cases.

When the parameter window is active, the selected parameter can be copied (Edit|Copy or Ctrl-C). This can then be pasted into the current case or into another open case (in the same AMBER run).

On pasting, a new parameter dialog for the relevant type of parameter is displayed, with information from the copied parameter already filled in. The user can then rename the parameter and change other information before adding it to the case file.

A few parameter types cannot be copied: Time-dependent (piecewise constant or linear); Import Parameters; and Sample Parameters using a Sample File (see Section ).

Built-in parameters become user-parameters when copied. So, it is not possible to copy TransferRate directly over the TransferRate of another case.

Copying to a new case will fail if relevant NameSets do not exist. Where NameSet entries differ, only the expressions for exactly matching items will be pasted. Copying Derived Observers will fail if the parameters that they depend on do not exist.

The copy and paste uses the Parameter File (creating a temporary file for the purpose). The Parameter File capabilities have been enhanced for this, and the new features can be used in general. The following description is an update of the reference guide section (5.4.1.9), which contained some errors.

### **3.1 Parameters from file**

Parameters can be created by reading a file. This has been included to allow parameter information written in another application to be read. The file can also be created manually.

By selecting the “From file” option on the “new parameter” dialog, a filename can be given (\*.apf by default). This file must contain the description for a single parameter. Once read, the user is left with the edit dialog for the parameter specified.

The contents of the file is keyword based. Valid keywords are:

NAME  
DESCRIPTION  
UNITS  
PDF  
LOG-PDF  
CDF-COORDS  
OBSERVER  
MULTIPLICITY  
VALUE  
DERIVED-OBSERVER

Comment lines beginning with # are ignored. Other lines are also ignored, but should be avoided.

NAME is optional. If not included then a default name will be used. If included, then the input line simply reads as follows.

NAME parameter

DESCRIPTION is also optional. If included, the line(s) read

DESCRIPTION “a description  
which can be split across lines”

where the “...” pair delimit the description.

UNITS is also optional. If not included then a dimensionless parameter is implied. If included, then the input line simply reads as follows.

UNITS “units”

for example

UNITS “kg m-3”.

If PDF or LOG-PDF are included, then the parameter is a sampled variable and OBSERVER, DERIVED-OBSERVER, MULTIPLICITY and VALUE cannot be specified. In this case, the full inputs are

PDF type data [BEST best-estimate]  
LOG-PDF type data [BEST best-estimate]

Where the type is Uniform, Gaussian, “Truncated Gaussian”, Triangular, Beta, or “General CDF”.

The data that follows depends on the type.

Uniform	Min Max
Gaussian	Mean Standard-deviation
Truncated Gaussian	Mean Standard-deviation Min Max
Triangular	Min Peak Max
Beta	Min Max A B
General CDF	Min Max (and a CDF-COORDS line)

The best-estimate is optional, and follows the keyword BEST.

For the General CDF, an additional line is required to give intermediate cumulative probabilities. This reads

CDF-COORDS v1 p1 v2 p2 ...

where the vN are values are the pN are cumulative probabilities. The VN must be between the Min and Max and in increasing order. The PN will be between 0 and 1 and in increasing order.

The keyword OBSERVER, on a line by itself, creates an Observer rather than a Standard parameter. In either case the MULTIPLICITY keyword is used to specify the multiplicity. If omitted, a scalar parameter will be created.

The MULTIPLICITY line can indicate a one-dimensional or two-dimensional parameter.

```
MULTIPLICITY nameset
MULTIPLICITY nameset1 nameset2
```

The namesets can be internal or user defined. Internal nameset names are: Contaminants, Decays, Compartments, Transfers, and Sources.

The VALUE keyword specifies numeric values, or expressions. For parameters with multiplicity, a series of VALUE statements can be given.

Some valid VALUE lines are:

```
VALUE 1.234
VALUE "a+b"
VALUE * 0.0
VALUE A Nuc1 "x/2"
```

In general, the indexing (one item per dimension) must be followed and a number or a quoted expression.

For a Derived Observer, the input simply reads

```
DERIVED-OBSERVER master type [atSymbol]
```

and the MULTIPLICITY, VALUE, OBSERVER and PDF entries are not allowed. The UNITS entry is allowed but will be ignored on reading (as the units will be the same as for the master symbol).

In the input line, master is the name of the master symbol, type is AT, MAX or MAXTIME and the atSymbol is required if AT is used – it gives the parameter that provides the time at which the derived observer is evaluated.



## 4. Other New Features

The following features have also been added in Version 4.5:

- new units through GUI
- switch on/off transfers
- auto-save after calculate
- search according to indexing.

These are all fairly self-explanatory.

### 4.1 New Units

New units can be added through the Options|Units menu item. The dialog has Add Base, Add Derived and Delete buttons. Built-in units cannot be deleted. To add a base unit requires just its name and setting the “prefixable” flag (to indicate if SI prefixes can be used with the unit). Adding a derived unit also requires its definition, e.g. to add Newtons, set the name to N, the prefixable flag on and the definition to 1 kg m<sup>2</sup> s<sup>-1</sup> (the 1 can be omitted if desired).

Note that units cannot be edited – just delete and add back the correct version.

### 4.2 Switch Transfers On/Off

Transfers can now be switched off (made inactive) without setting the transfer rate to zero. This is useful for variant calculations when the original is to be retained. There is a new check-box on the Transfer Edit dialog. This disables the other items on the dialog (as a reminder that the transfer is inactive). Inactive transfers are drawn with thin dashed

lines on the model window. Both solvers used in AMBER skip inactive transfers.

### **4.3 *Auto-save after Calculate***

A new check-box on the Calculate dialog allows an auto-save to be requested. With this option selected, the case is saved once the calculation finishes. This is particularly useful for long probabilistic runs in reducing the risk of results being lost due to a power failure or computer crash.

### **4.4 *Search by Indexing***

A new option in the Search dialog allows a search for parameters that include a selected NameSet in their multiplicity. Built-in and user-added NameSets can be selected.

## 5. Bug Fixes and Minor Changes

In applying AMBER Version 4.4 to a range of situations, some minor problems were identified. Version 4.5 addresses many of these...

### *Graphs should include SnapShot times*

Graphs now include any SnapShot times in the times used. This improves the accuracy of the graphing near discontinuities.

### *Adding SnapShot times after a calculation causes a crash*

If a SnapShot time was added after a calculation had been done, the number of result times was incorrectly handled leading to the program crashing. This has now been corrected.

### *Derived Observers not properly deleted*

Deleting a Derived Observer removed the symbol but did not remove the associated definition – so the symbol reappeared when the case was re-opened. The deletion is now properly implemented.

### *Large cases give negative %-done message*

An integer overflow caused the percentage done to be displayed as a negative number. This has now been corrected.

*Use of ESC to close dialogs sometimes causes a crash*

In some situations the use of ESC to close a dialog was handled by the wrong dialog causing AMBER to crash. This has been fixed, along with a potentially similar problem if the Close-box was used to close a nested dialog.

*Scroll bars on Check Parameters and Check Preconditions sometimes disappear*

The scroll bars could become hidden behind the text area. The width of the text area is now set properly.

*Units are not reported on the Check Parameters window*

This has been corrected.

*Drawing large graphs can cause the system to hang*

Graphs of more than 32,000 points cannot be handled by QCharter. AMBER now prevents any attempt to draw a larger graph.

*Cases with literal values used in expressions no longer work*

The units functionality incorrectly tried to scale the literal value. This has now been corrected.

*The "largest" function gives zero if all the values are negative*

AMBER now correctly reports the largest value.

*The syntax for units should allow things like "(Sv/y)/(Bq/y)"*

The units definitions now allow the use of bracketed sub-units as in the example. The bracketed sub-unit acts as a single item, and so can be raised to a power.

*Graphs sometime draw the wrong line when there are Selections*

If a Selection appeared in the list for a graph index but a single item was selected, then the graph was sometimes (depending on the recent history of the dialog) drawn for the wrong item. This has been corrected.

*StepUp and StepDown do not work*

Earlier versions of the reference guide mistakenly gives these functions a capital S. They should be stepUp and stepDown. AMBER has been changed to accept either and convert to the intended form. Note that “And” and “cyclicLookup” were also wrong in the reference guide – they should be “and” and “cyclicLookUp” respectively.

*There should be a more convenient way of getting a smooth transition from 0 to 1*

Four new functions have been added, two for smooth transitions and two for ramps (piecewise linear).

The smooth transitions are provided by smoothUp and smoothDown. Each takes three arguments. They are similar to stepUp and stepDown except that the step is smoothed.

The smoothDown(x, X, s) acts as follows. For small x (and all negative x) it is equal to 1.0. For large x it approaches 0.0. When x=X the value is 0.5. The smoothness of the transition is controlled by the value of s, a larger value of s giving a smoother transition.

The units of x and X must be compatible and s must be dimensionless.

$$\text{smoothDown}(x, X, s) = \frac{1}{1 + (x / X)^{2/s}} \text{ for } x > 0, \text{ and } 1 \text{ otherwise}$$

$$\text{smoothUp}(x, X, s) = 1 - \text{smoothDown}(x, X, s).$$

The first argument will often be t (time), but need not be. The expected range for s is 0.1 to 10.

The ramp transitions are provided by rampUp and rampDown. Each takes three arguments.

The `rampDown(x, start, end)` acts as follows. For `x` less than `start` it is equal to 1.0. For `x` greater than `end` it is equal to 0.0. Between `start` and `end` a linear ramp is used.

The units of `x`, `start` and `end` must all be compatible.

$$\text{rampUp}(x, \text{start}, \text{end}) = 1 - \text{rampDown}(x, \text{start}, \text{end})$$

Note that if the functions are used with `x` as the time, `t`, then `start` and `end` should generally be parameters that are declared as switch parameters.

*The time-stepping solver loses accuracy in some situations after a solubility constraint period ends*

The logic in the solver that sets the accuracy parameters was wrong in some cases. This sometimes led to oscillations in the solution. This has now been corrected.

*Why can deterministic graphs not be drawn for a probabilistic case with only one sample?*

Probabilistic cases with one sample are now treated as deterministic for the purposes of graphing.

*Specified Amounts are sometimes lost in the GUI*

This problem has been corrected.

*When reading library files, if the wrong filename is given AMBER creates an empty file of that name and then gives an odd error message*

This problem has been corrected – if a file does not exist then an error is immediately reported (this applies generally across AMBER).

*When adding a NameSet – cancelling does not prevent an item being added*

This problem has been corrected.

*The Switch Parameter dialog should only list parameters with time units*  
The dialog (and the SnapShot dialog) now only lists parameters with time units if units errors are being fully respected. Otherwise, dimensionless parameters are also included (this is to allow old cases to continue to function).

*Cases with some units set can give different results after reloading*  
If parameter units were changed and new units errors arose, then the old unit conversions were being remembered until the case was reloaded. The behaviour is now consistent – old conversions are forgotten. However, it is recommended that cases are fully converted and then full units checking is turned on.

*The Partial sampling option sometimes forgets which parameters to sample*

If a sampled parameter was edited, its status was reset to be sampled in all cases (even if using the best estimate had been specified). This has now been corrected.

*Why can't switch and snap-shot parameters be indexed?*

They can now – and they act as a set of switch or snap-shot parameters.

*Results from derived observers set at the end time of the run are sometimes incorrectly reported as being zero*

This was due to rounding errors making the end time, derived observer time and possibly a snapshot time be slightly different and was particularly a problem when different internal and result time units were used. Allowance for small rounding errors has now been made to correct the problem.

*Why isn't there a "rename mapping" option like the "rename parameter" one?*

This has now been added. If a mapping is renamed, the user can request that all references to the old name in expressions are updated.

*Why isn't there a receptor mapping like the donor mapping?*

This has now been added. It gives the target compartment for each transfer.

*Expressions used for Availabilities (e.g. solubility limits) are not checked in Check Parameters.*

There are now checked – note that the Availability schemes create temporary parameters, so any errors will be associated with parameters with unfamiliar names. The required corrections should be made in the availability scheme dialog.

## 6. Undocumented Features and Future Developments

### 6.1 Undocumented Features

AMBER Version 4.5 has enhanced “batch mode” capabilities. These allow a control file to be written that runs AMBER cases and saves results to files in various formats.

Version 4.5 also contains a capability for defining Sample Files. These list allowable parameter values and combinations of parameter values that can be used. This gives AMBER users the ability to define arbitrary pdfs, to specify complex correlation structures and to force particular values to be sampled.

A “Viewer” version of AMBER is currently under development. This has no solvers and prevents the user making any changes except for adding or changing observers. The viewer version will require no dongle, so can be distributed more freely.

These capabilities will be documented for the next release, but if you would like to try them out before then, please contact us.

## **6.2 *Future Developments***

AMBER is under continual development. As such we welcome feedback from AMBER users regarding possible future developments – please contact us at the address at the front of this document if you have any suggestions.