



**RADventure Referential Integrity Wizard
Programmers Documentation**

Table of contents

1	Preface.....	3
2	RADventure Referential Integrity Wizard	4
3	Installation	4
4	Referential Integrity Wizard Utility Template.....	5
5	Options in the menu	12
6	Layout log file	17
7	Commandline interface	18
8	RADventure support.....	19
9	Copyright.....	19

1 Preface

Relational database, like most Clarion programs work with, are based upon several files, which are related (linked) to each other. This relation is achieved by the contents of a combination of fields in each record in one file (the child file) always uniquely identifying a specific record in an other file (the parent file). A child file may have many parent files.

Usually it is required that the identified record in the parent file does indeed exist. This is accomplished by validating during the creation of the child record that the identified parent exists, and specifying referential constraints, which determine what action should be taken when a parent records with child records is deleted, or has the contents of its link fields changed.

However, for several reasons (no referential constraints active, child-parent relation only implemented after child records had been created, database corruption) it might be that there are child records with one of more parents missing. Since the existence of those semi-orphans might hamper the normal functioning of the system, it is desirable that they are identified and possibly removed.

This identification and removal can be done by hand, but this can take up a lot of time. Faster and more efficient is to have this done by programs. However, in a medium to large-scale system, with tens or hundreds of files, the creation and testing of those programs is no trivial task.

An even more efficient way would be to have those programs automatically generated. We are using Clarion after all, aren't we? However, the crafting of the necessary templates to accomplish this is even more challenging as the creating of the program itself. This is where the RADventure Referential Integrity Wizard comes into use. It is already build and tested, and based on the data dictionary quickly generates efficient programs. Usually the programs can be used as-is (after compilation), and even have a command line interface to enable them to be automatically started during e.g. night- or weekend processing.

Of course you could try to develop this yourself. You are a smart developer after all, aren't you? However, when you calculate the time it would cost you to build this, you will soon find out that the smartest thing you can do is to save yourself time (and therefore money) and purchase the RADventure Referential Integrity Wizard.

2 RADventure Referential Integrity Wizard

The RADventure Referential Integrity Wizard enables you to automatically create an application, based on a data-dictionary, which checks if all parent-child relations defined in this dictionary are still intact. If not, it can optionally delete the offending child records.

Although the wizard itself makes no difference between SQL - and non-SQL systems, it is especially usefull in non-SQL systems, since SQL systems have often already other tools for this. We advise the toolset of the RADventure Oracle Tools for this.

The files are processed in generational order, and log files are available.

3 Installation

Installation of the RADventure Referential Integrity Wizard template is completely automatic. You can start the RADRIwiz .exe. This installation program will automatically install and register the templates. The templates can automatically register themselves in your template registry.

The files installed are the following:

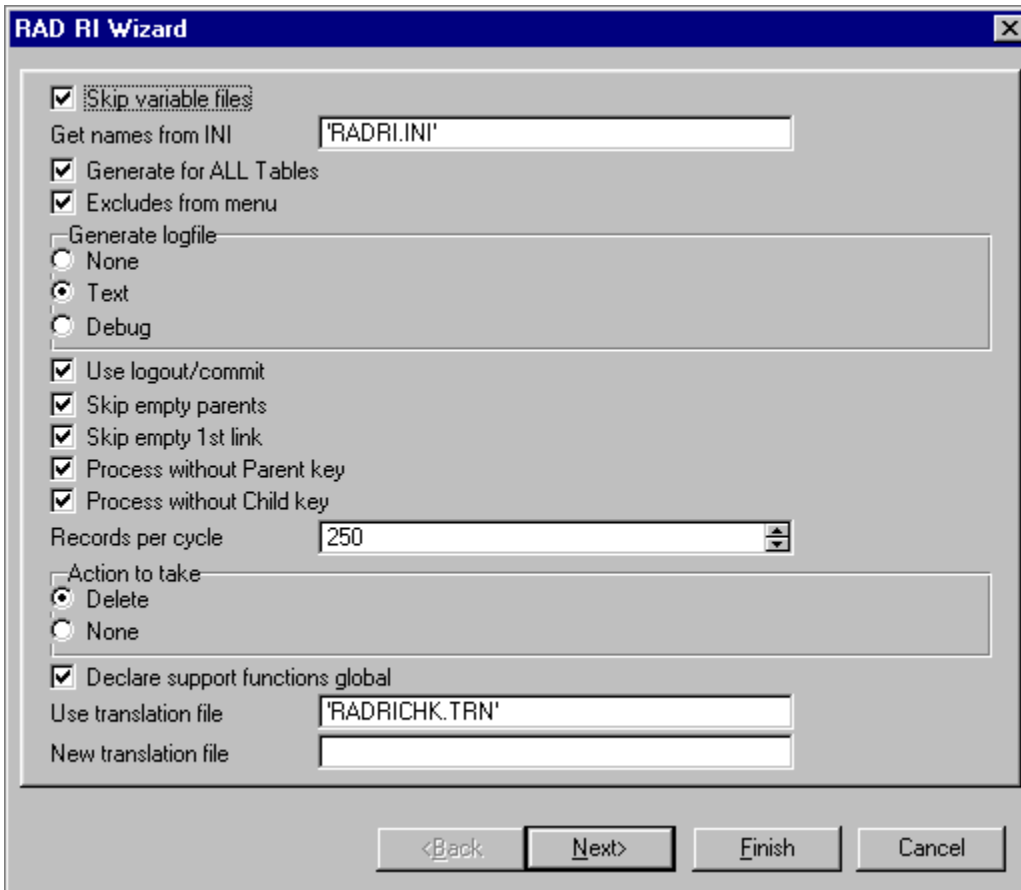
\template	RADRIWizard.tpl
------------------	-----------------

The RADventure Referential Integrity Wizard templates consist of a Utility template which, when executed, will created a application to check the referential integrity of the files in the dictionary it is based upon.

4 Referential Integrity Wizard Utility Template

To use the RADventure Referential Integrity Utility template use the following steps:

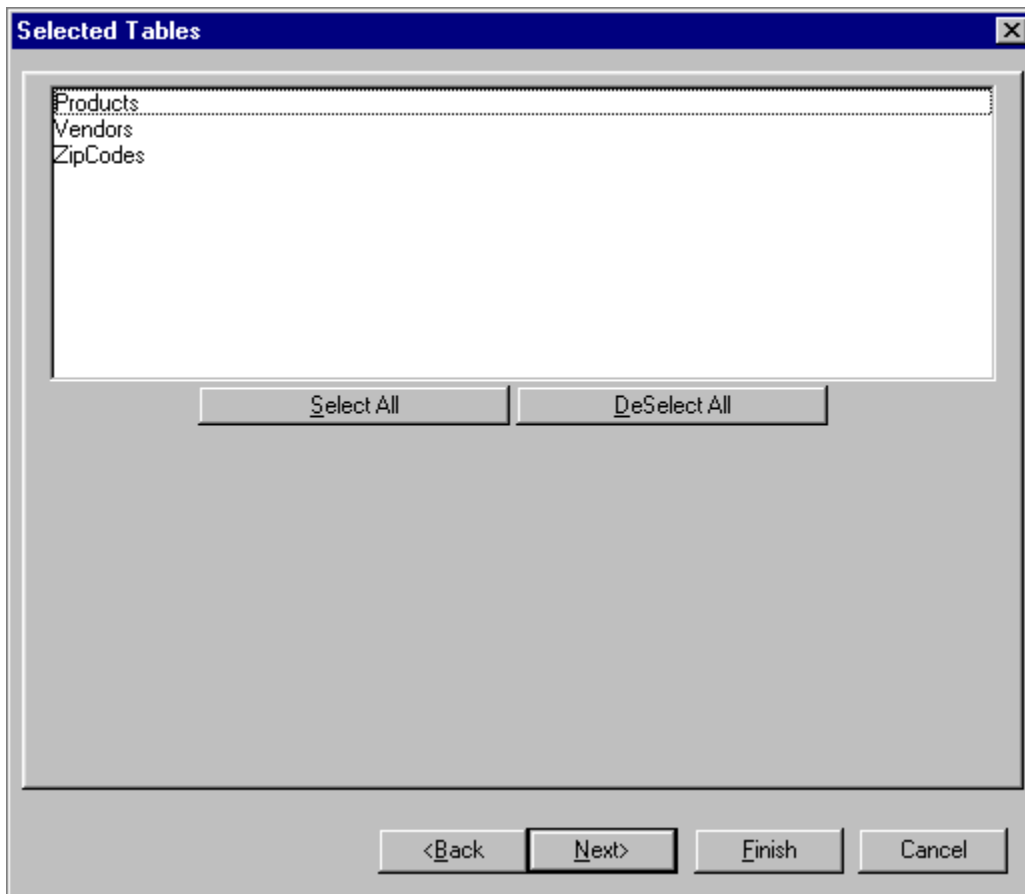
- 1) From the Clarion IDE menu choose File, New, Application.
- 2) In the New window
 - a) Provide the name of the application to create, e.g. RI
 - b) Make sure 'Use Quick Start' is **not** checked
 - c) The filetype should be set to Application (*.app)
 - d) Press enter to save the file and proceed.
- 3) In the application properties screen
 - Choose the dictionary to process. (The example screens are based upon the INVNTORY.DCT from the Clarion 5.5 Examples\Invntory directory.)
 - Make sure the Application Wizard is **not** checked
 - Application Template should be ABC (or ABC based).Click OK to proceed.
- 4) In the application tree screen
 - a) from the menu select Application, Template Utility.
- 5) From the Select Utility screen select RADRiward - RAD Referential Integrity Wizard.
Click on Select.
This window should appear.



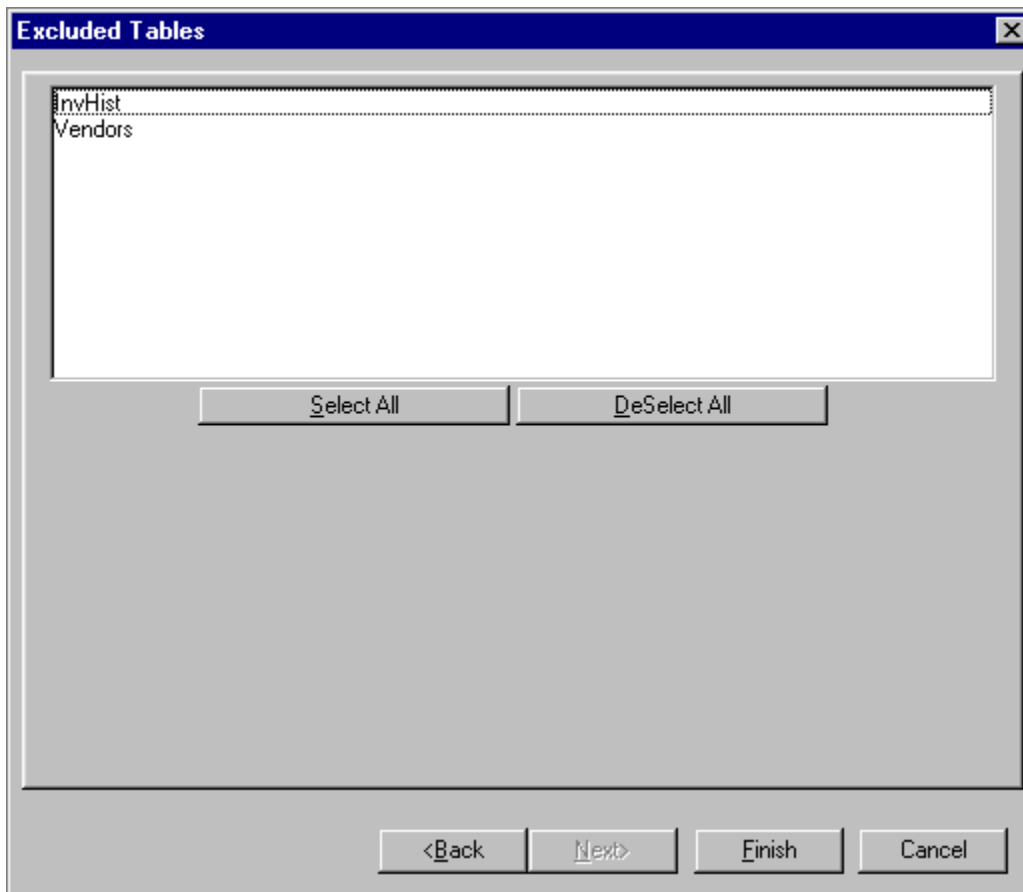
- Skip variable files
 - When a file has a variable file name, this might denote that it is not an intrinsic part of the system. There this option enables the user to skip files with variable filenames.
- Get names from INI
 - When files with variable filenames are present, this option gives the opportunity to load these variables from the specified INI file. The sectionname is always Filename, with the entries being similar to the actual variablenames.
- Generate for ALL Tables
 - This generates procedures for all parent files. If not, only the parent files specified on the next tab are used.

- Excludes from menu
 - This enables the end user to specify runtime child files to exclude.
- Generate logfile
 - This specifies whether or not a logfile should be created and, if so, which kind
 - None: no logfile is created
 - Text: a textfile with the name ri.log is created
 - Debug: a debuglogfile is created. This requires the RADventure Debug Class.
- Use logout/commit
 - When set, this places the deletes in a transaction frame, which provides a speed advantage especially for TopSpeed files.
- Skip Empty Parents
 - When set, an empty parent table is not processed. Apart from this speed advantage, this also is useful when e.g. the parentfiles might be optional.
- Skip empty 1st link
 - When set, childs with the 1st linkfield being empty (CLEAR/NULL) are skipped. This is useful when this denoted an optional link.
- Process without Parent key
 - When set, relations where the parentkey is undefined are skipped. Such a relation is unusual, but might be used to enable subset-selection based on the relation.
 - When not set, the parent record is searched by a sequential search through the file, which has impact on the processing time.
- Process without Child key
 - When set, relations where the childkey is undefined are skipped. Such a relationship might exist when it is only used to validate the childfield.
 - When not set, the child records are processed sequentially, which has impact on the processing time.

- Records per cycle
 - This value determines how many records are processed per cycle. A higher number results in a faster throughput, but gives less opportunity to interrupt.
- Action to take
 - Delete: selected records are actually deleted
 - None: the generated program perform no actual deletes
- Declare support functions global
 - When set, generated support functions like RICheck have the Declare Global set, and are not include in the Process tree of the calling procedures
 - When not set, the generated support functions like RICheck are included in the Process tree of the calling procedures and are not Declared Global
- Use Translation file
 - The specified file is included in the application and compiled for runtime-translation purposes. If specified, this file should be present at compile time.
- New translation file
 - The specified file is generated with the found translation string

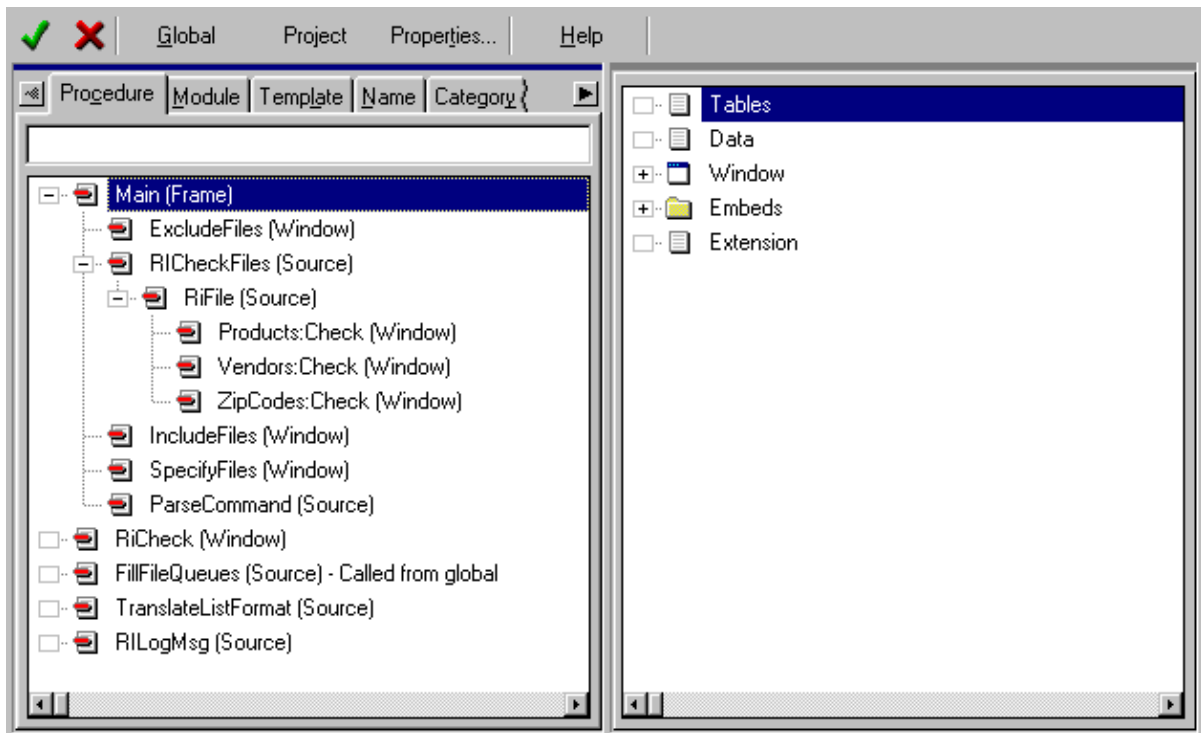


The selected files are included in the list of Parent files to process.



The selected child files are excluded from processing.

After this you press the Finish button and the application is filled with procedures. The resulting Application tree should be similar to the following:

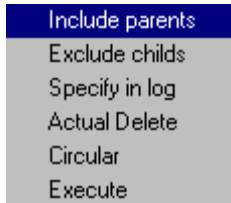


Make and run the application. A screen similar to the following should appear:

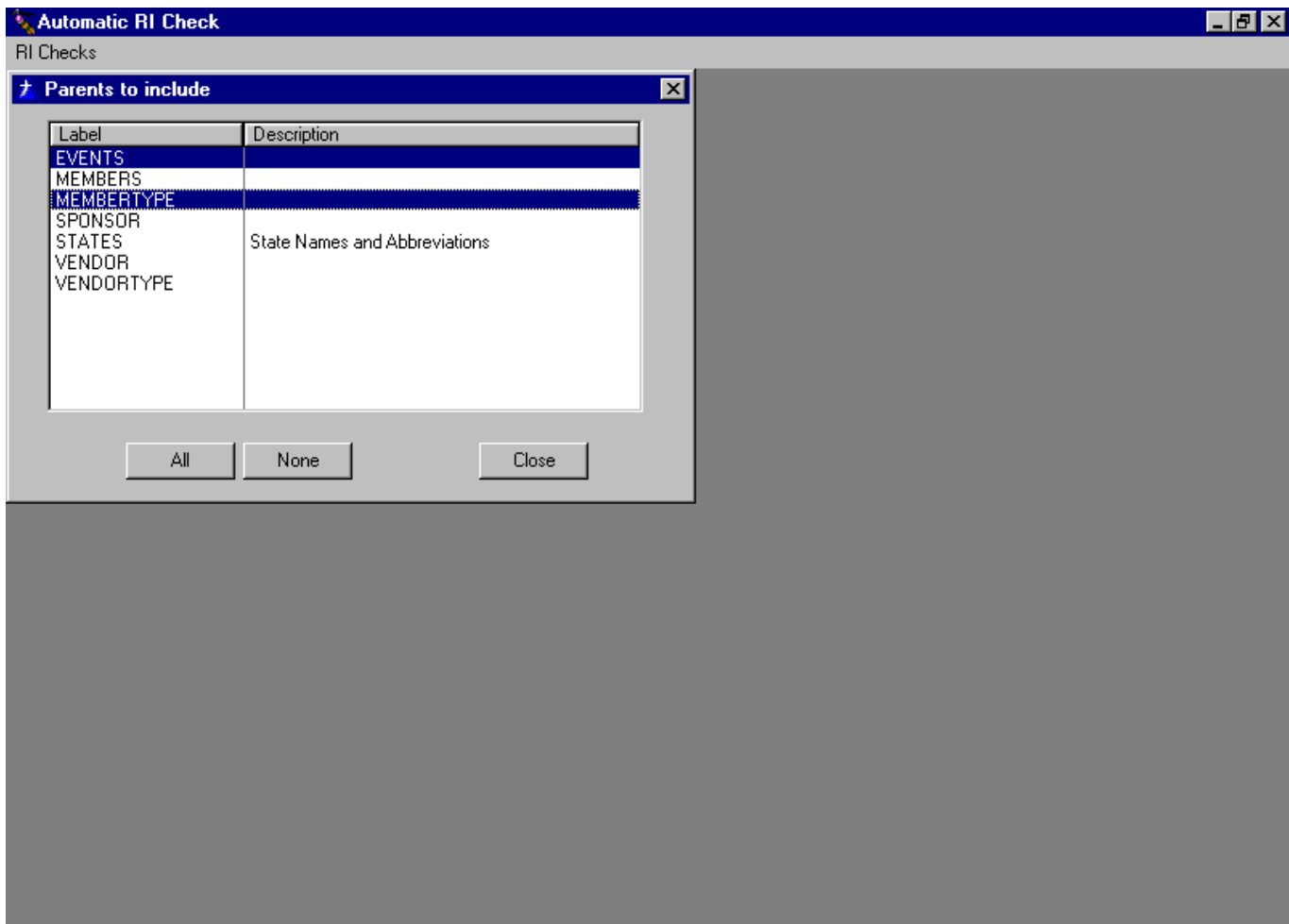


5 Options in the menu

Include parents



This present a screen similar to the following:

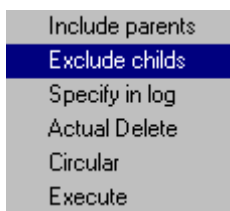


Here you can select the parent files to process. Only files, which have 1:N relationships, defined in the data dictionary are listed here.

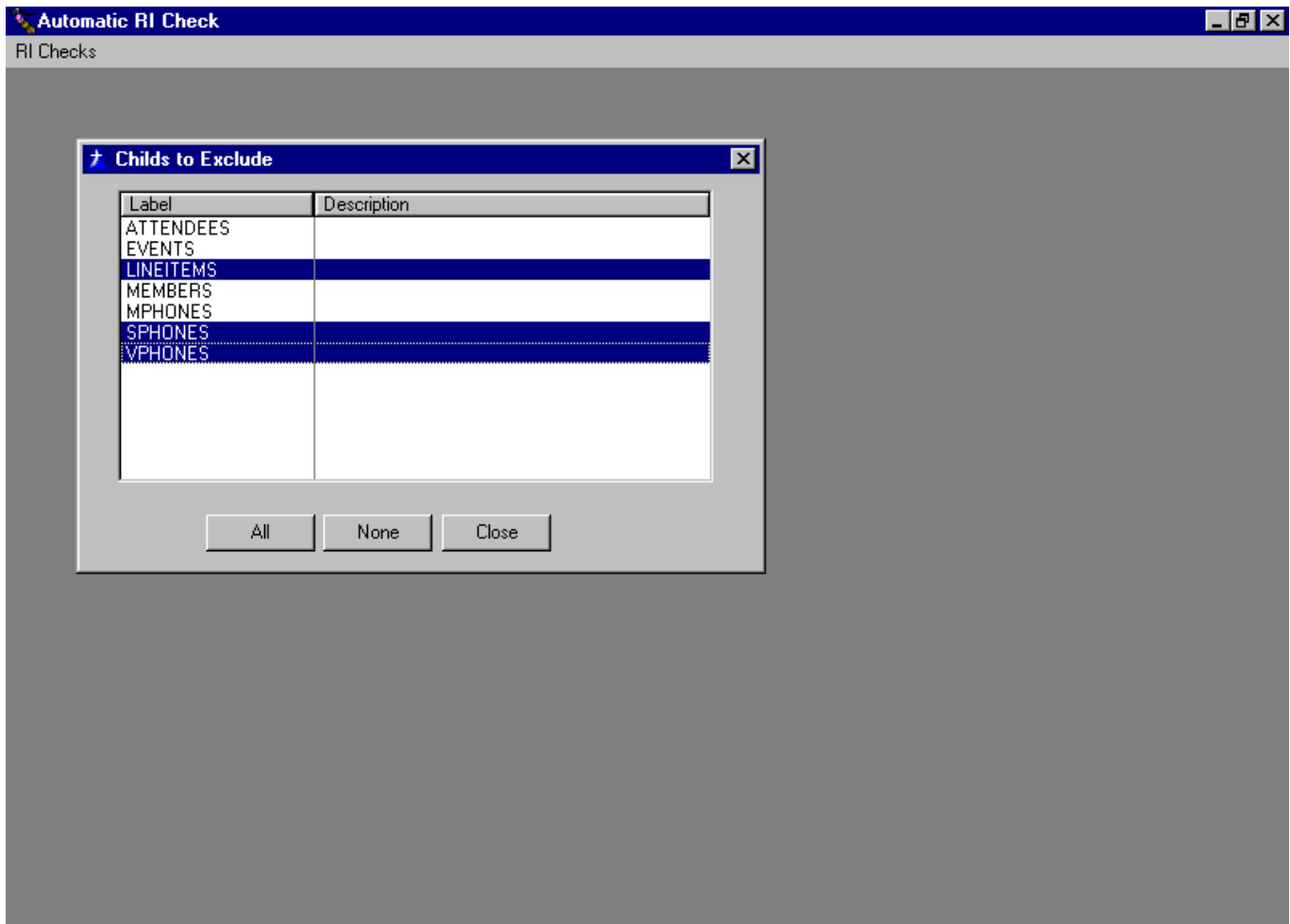
A file is selected/deselected with either the spacebar or a mouse click. With the All-button all files are selected, with the None-button all files are deselected. With the Close button you leave the screen.

In the example screen, based on the EVENTMGR example application of Clarion, the EVENTS and MEMBERTYPE files are selected for process. The descriptions of the files are derived from the data-dictionary. In this example only the STATES file has a description.

Menu option Exclude Childs



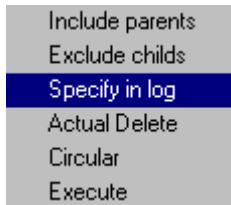
This presents a screen similar to the following:



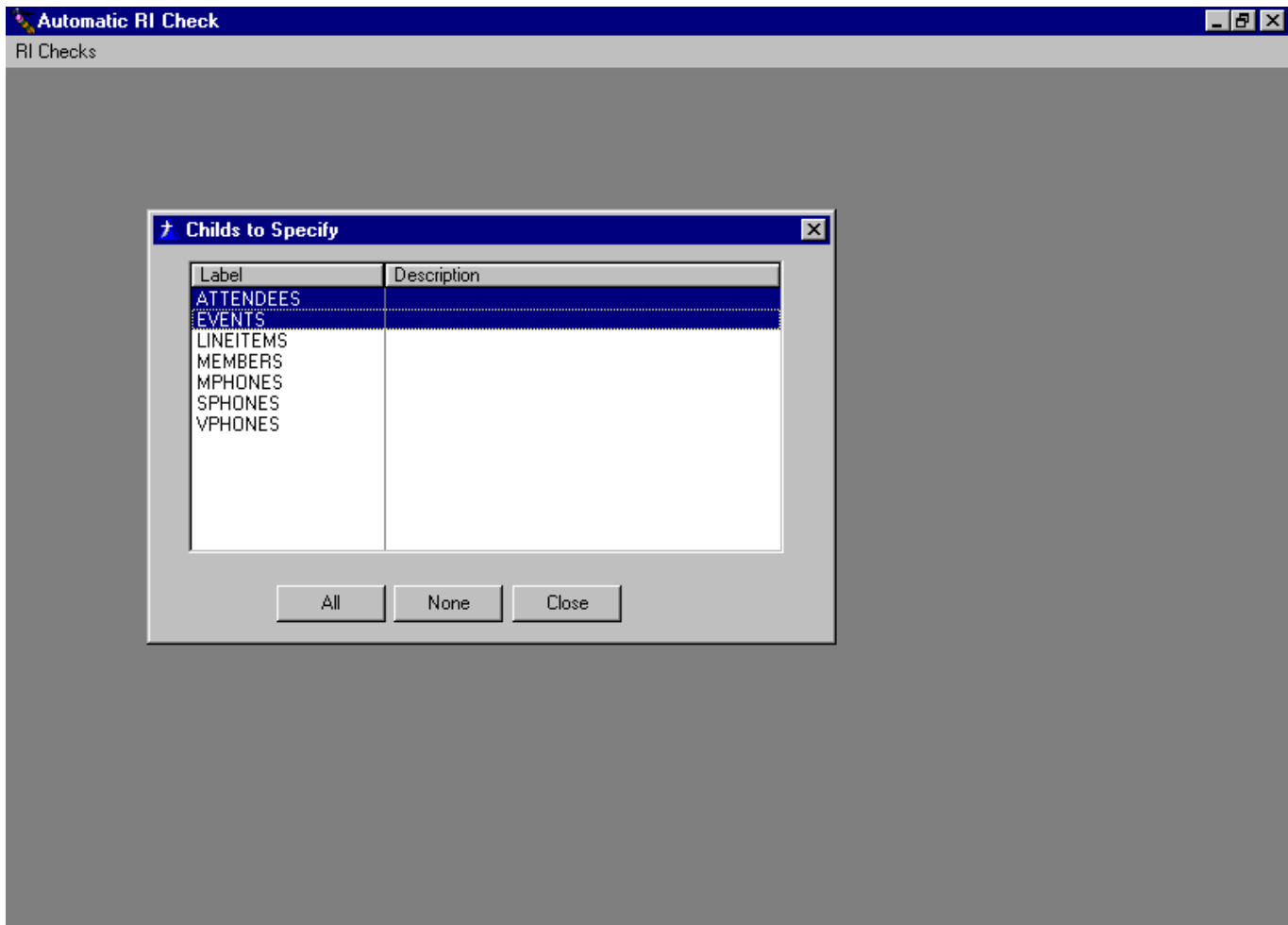
In this screen you can specify which child files should be excluded from processing. In this case the LINEITEMS, SPHONES and VPHONES are selected. The descriptions are initially derived from the data dictionary, which is the case for no files in this example.

Files are selected/deselected with either the spacebar or a mouse click. The All-button is used to select all the files, and the None-button to deselect all the files. The Close-button closes the selection and returns to the menu.

Menu-option Specify in log



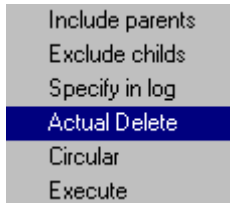
This presents a screen similar to the following:



Here you can select the child files for which the individual records should be specified in the log file, in this example the ATTENDEES and the EVENTS files.

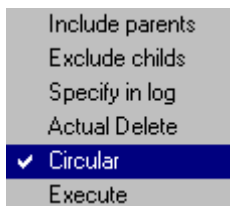
Files are selected/deselected with either the spacebar or a mouse click. The All-button is used to select all the files, and the None-button to deselect all the files. The Close-button closes the selection and returns to the menu.

Menu option Actual Delete



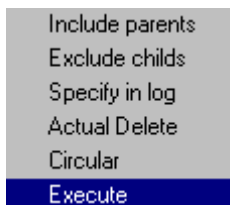
This is a checkbox option which, when set, specifies that the deletions should be actually performed.

Menu option Circular



This is a checkbox option which, when set, specifies that the circular processing should be performed. This is done when files have a circular generation definition. In this case the processing restarts as long as there are deletions done within the group.

Menu option Execute



This starts the actual execution of the RI check.

NB: please make sure there is a recent backup of the datafiles before your start to execute.

6 Layout log file

A log file starts with a header line, which provides headers for the columns Total, Processed and Deleted.

Total represents the total number of records in the processed child file.

Processed represents the number of examined records. This can be less than the Total number of records, when either the child key used is optional, or the processing is interrupted.

Deleted represents the number of records that are deleted, or are selected for deletion.

Check <filename>: this represents the parent file processed. The physical filename is presented in parenthesis.

<Filename>: This represents the child file processed, together with its physical name between parenthesis and the total, processed and deleted number of records.

<Filename> Link: this represents how the child file is linked to the parent. This is only included in the log when this child file should be specified. The line also depicts the link field(s) used, the type of identification key (primary, unique or 1st), the name of the identification key and the composing fields.

In the lines under the Link lines the action taken for the record (Deleted or Selected) is written, followed by the link and identification fields.

When a parent has no records, and empty files should be skipped, this is logged as 'Skipped because of empty file'.

When circular processing occurs, this is logged as 'Circular processing because of mutations'.

7 Commandline interface

To enable the processing to be started automatically, there is a commandline interface defined.

/INCLUDE

Include the immediately hereafter specified labels as parent files to include. The label ALL denotes all files.

/EXCLUDE

Include the immediately hereafter specified labels as child files to exclude. The label ALL denotes all files.

/SPECIFY

Include the immediately hereafter specified labels as child files to specify. The label ALL denotes all files.

/DELETE

Perform actual deletion.

/CIRCULAR

Use circular processing.

/IMMEDIATE

Start immediately with execution.

So the commandline parameters **/INCLUDE VENDOR /SPECIFY ALL /DELETE /IMMEDIATE** denotes that the VENDOR parent file should be processed, with all child files specified in the log, the deletions should be actually performed and the action should start immediately.

8 RADventure support

RADventure Tools Support	
Email:	tools@radventure.nl
Telephone:	+31 (0)346 29 09 80
Fax:	+31 (0)346 29 09 08
Post:	PO Box 1069, 3600 BB Maarsse, The Netherlands

9 Copyright

RADventure Referential Integrity Wizard Template is copyrighted (c) 2002 by RADventure B.V.

RADventure Referential Integrity Wizard is provided as is, and you use it and the programs derived from it your own risk. RADventure B.V. and its employees accept no liability for anything lost, destroyed or damaged because of RADventure Referential Integrity Wizard tools. Use of this product implies acceptance of this condition.

All RADventure files are copyrighted by RADventure B.V. and may not be distributed.