

**1. Title**

**GRC Tools™  
GRC Country™  
User's guide  
Version 2011-1-1**



**GRC Database Information**

Nieuwe Prinsengracht 80-hs  
1018 VV AMSTERDAM

The Netherlands

[graham@grcdi.nl](mailto:graham@grcdi.nl)

[://www.grcdi.nl](http://www.grcdi.nl)

GRC Tools is a trademark of GRC Database Information. All other trademarks belong to their respective owners

All information provided within this manual and within the program GRC Tools and the lookup tables provided with the program GRC Tools are copyright G.R. Rhind 2008.

---

## 2. End-User Licence Agreement & Disclaimer

# End-User Licence Agreement & Disclaimer

You are granted a single-user licence to use **GRC Tools™/GRC Country™**

The structure and organization of this software are the property of GRC Database Information. You may not make, have made or permit to be made copies of this software, the documentation or any portions thereof except a sole copy for backup purposes. You agree not to modify, adapt, translate, reverse engineer, de-compile, disassemble or create derivative works based on this software

The lookup tables delivered with this software and any delivered in any later versions of this software or delivered separately to later versions of this software remain the property of GRC Database Information. These lookup tables may not be altered in any way. These tables may not be copied, printed or reproduced in any way except through the use of such options within **GRC Tools™/GRC Country™**, nor may they be used or incorporated in any software products or in any other media external to this program.

GRC Database Information retains title, intellectual rights and ownership of this software, the media upon which it is recorded, and all subsequent copies of the software, regardless of the form or media or whichever form the original and other copies may exist.

This program and its associated tables and documentation are provided as is. GRC Database Information makes no warranty that **GRC Tools™/GRC Country™** will meet your requirements and that the operation of the software will be uninterrupted or error-free. No warranties or guarantees of any kind are offered. GRC Database Information is not responsible for any problems or damage caused by the software that may result from its use or misuse. This includes, but is not limited to, computer hardware, computer software, operating systems and any computer- or computing-accessories. You agree to hold GRC Database Information and/or any persons associated with the creation of **GRC Tools™/GRC Country™** harmless for any problems arising from the use of the software.

### 3. About GRC Tools

## About GRC Tools™

Many companies now have address databases containing addresses from many different countries, and many of these are increasingly being used for cross-border mailings. Up to now, however, there have been problems in using and outputting this data given the differences in language, format and expectations within and between these different countries.

**GRC Tools™** is designed to resolve many of these problems. Based on many hundreds of man-hours of work with, and analysis of, international databases, this software is a painless way of assuring accuracy and consistency in address databases for different countries.

**GRC Tools™** works on two premises - that data should be **consistent**, and that data should be **accurate**. Though address formats differ to such an extent between countries that consistency throughout a database are said to be achievable only at the expense of accuracy, **GRC Tools™** ensures consistency and accuracy on a country and/or language region level.

Achieving **consistency** on a country/language level enables you as a database manager to make your data more accessible (easier to find) as you know the format in which the data will be stored. It enables macro-changes to data such as telephone numbers to be made without excess hassle and, very importantly, it will dramatically increase your ability to locate and weed out duplicates, thus saving potentially enormous amounts of money in printing and mailing costs, and increasing the productivity of the database.

By working on a country/language level, consistency need no longer be achieved at the expense of **accuracy**. Nothing is more irritating to the person to whom you are mailing than being addressed in the wrong language, or being addressed in the wrong way. These are highly emotional points for many people and can dramatically reduce the value of any mailing that you do. Furthermore, format the address in the wrong way, and the mailing may never reach its intended recipient and be returned as undeliverable.

**GRC Tools™** ensures consistency and accuracy without losing flexibility.

**GRC Tools™** is a set of modules which gather data from lookup tables and change your data file according to what it finds there. These lookup tables have been formulated, based on a great deal of research, to make the data accurate on the basis of country or language area.



**You are strongly advised to read the section on each module before running it to understand fully what it will do to your data.**

**GRC Tools™** has been thoroughly tested. However, as a highly flexible program unexpected problems can always occur. As needs and requirements differ greatly between users, and modules may be run in combinations not expected by the program, there is a small chance that changes may be made that you do not expect or do not want.



**It is highly advisable to back-up any data files over which you wish to run GRC Tools™. Remember that GRC Tools™ changes data. The author disclaims any responsibility for damaged data caused by the use or misuse of this program.**

Should you have any comments or suggestions for improvements to **GRC Tools™**, we would be delighted to hear of

them.



Countries have been chosen for inclusion in **GRC Tools™** on a pragmatic basis, and on postal and address requirements.

The list excludes countries with a claim to independence or which are *ipso facto* independent and includes some dependent territories. The parameters for inclusion have been that in terms of address systems a clear distinction can be made in some respect for the country or territory concerned.

**4. About GRC Country****About GRC Country™**

**GRC Country™** uses the knowledge embedded within the tables used by **GRC Tools™** to attempt to assign a country code for an address record not already coded. Tests have shown that **GRC Country™** works best for developed countries with postal code systems, and on files which contain well formatted and correctly fielded data. Our tests of this file type have produced between 96 and 100% correct country coding. Tests of files which have a greater number of developing countries, including countries without postal code systems, and with poor data formatting and fielding, have produced results of 82% accuracy.

**5. Technical Requirements**

# Technical Requirements

**GRC Tools™/GRC Country™** requires:

- An IBM compatible computer with 486 50 MHz processor or higher
- Mouse
- 10 MB RAM
- DOS version 3.1 or higher
- Microsoft Windows version 95 or higher, or Microsoft Windows NT 3.51 or higher.
- 50 MB of disk space

For **GRC Tools™** , for each data table being processed, you must have:

- A field containing a country code

and if your table contains diacritical characters, you must know

- The code page in which the table was created.

For **GRC Country™**, for each data table being processed, you must have:

- A field to which to write a country code

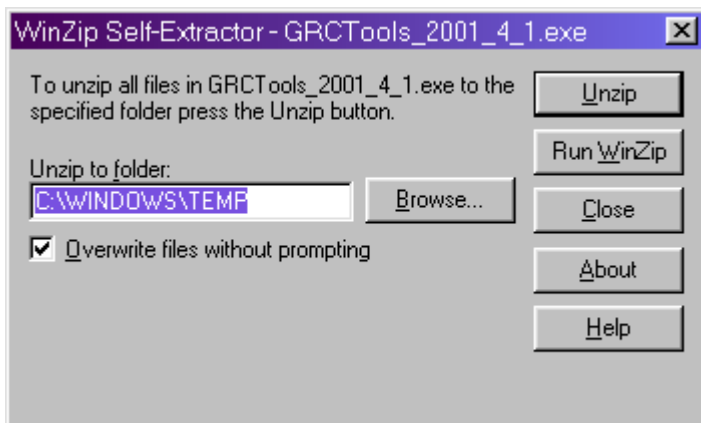
## 6. Installation

# Installation

**GRCCountry™** is installed with **GRC Tools™**. It must reside in the same directory as **GRC Tools™**. It should not be run at the same time as **GRC Tools™** is being run in the same directory.

**For Windows Vista you must install the program with administrator privileges.**

The file that you download from the GRC Database web site is a self-extracting zipped file. Run the file and choose the directory to which to unzip the files.



After unzipping the files, go to the specified directory and run the program SETUP.EXE. Follow the instructions to install **GRC Tools™** on your local drive.

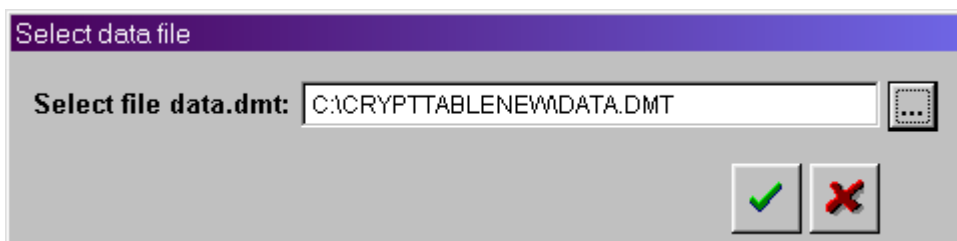
**GRC Tools™** is intended to be installed once only on a single computer.

When you start **GRC Tools™** the first time, you will be informed that you have run out of credits OR that your license has run out and that you need to order a new one.

If you have downloaded the trial version and have not yet been sent a data.dmt file, e-mail your name and company name with a request to receive your credits to [graham@gredi.nl](mailto:graham@gredi.nl).

We shall send you back a file called DATA.DMT, which contains your credits information.

You will either be asked for the location of the DATA.DMT file, or you can access this screen in the menu HELP-> UPDATE CREDITS. Use the browse key to locate the file and press the tick key to complete registration.



**7. Summary of the modules****Summary of the modules**

**GRC Tools™** works on standardising names and addresses on a modular basis. Below is a summary of the modules, in the order in which they should be run for optimal quality, and the type of data upon which they work:

**Data processing**

<i>Process</i>	<i>Field contents</i>
Invalid strings - move field contents	Any alpha-numeric field
Field - split string	Any alpha-numeric field
Field - trim leading spaces	Any alpha-numeric field
Field - trim final string	Any alpha-numeric field
Non-numeric characters - remove	Any alpha-numeric field
Punctuation - remove	Any alpha-numeric field
Double spaces - remove	Any alpha-numeric field
Telephone numbers - burst	Any alpha-numeric field
Postal code country code - remove	Postal code
Thoroughfare types - parse	Any alpha-numeric field
Format postal code	Postal code
Postal codes - assess validity	Postal code
Postal codes - write corrected	Postal code
Postal codes - locate	Postal code, any alpha-numeric field likely to contain a stray postal code
Postal codes (incl. format) - locate	Postal code, any alpha-numeric field likely to contain a stray postal code
Postal codes - write located	Postal code
Language regions - assign	Uses postal code and street address data
Accents - remove	Any alpha-numeric field
Quotation marks - remove	Any alpha-numeric field
Apostrophes (French) - add	Any alpha-numeric field
Casing - to upper case	Any alpha-numeric field
Casing - to mixed case	Any alpha-numeric field
Casing - to mixed case - address	Street address data
Casing - to mixed case - other	Any alpha-numeric field, not street address
Articles - move or parse	Company name
"And" strings - move or parse	Company name
Abbreviations - standardise	Company name
Abbreviations in personal names - standardise	Personal name
Company types - standardise	Company name
Sub-building information - parse & standardise	Any alpha-numeric field
Post office box numbers - parse	Any alpha-numeric field which could contain stray post office box numbers, usually street address fields
Thoroughfare strings - standardise	Street address data
Building numbers - move or parse	Street address data
Building number suffix - parse	House number data

Commas - add or remove	Street address data
Building number/letter - format	House number/street address data
Sorting code - parse & standardise	Any alpha-numeric field which could contain stray sorting codes, usually postal town fields
Place names - parse & standardise	Any alpha-numeric address field likely to contain a settlement name
Place names - standardise	Settlement name
Provinces - parse/assign	Any alpha-numeric field which could contain stray province names
Provinces/regions - assign	Uses postal code data
Regions - assign	Uses postal code data
University strings - parse	Any alpha-numeric field
Attention strings - move field contents	Any alpha-numeric field
Department strings - move field contents	Any alpha-numeric field
Department strings - move field contents	Any alpha-numeric field
Department strings - standardise	Any alpha-numeric field
Personal name prefixes - parse & standardise	Any alpha-numeric field which could contain stray forms of address data, usually personal name fields
Personal name middle name(s)/initial (s) - parse	Any alpha-numeric field
Job titles - standardise	Job title

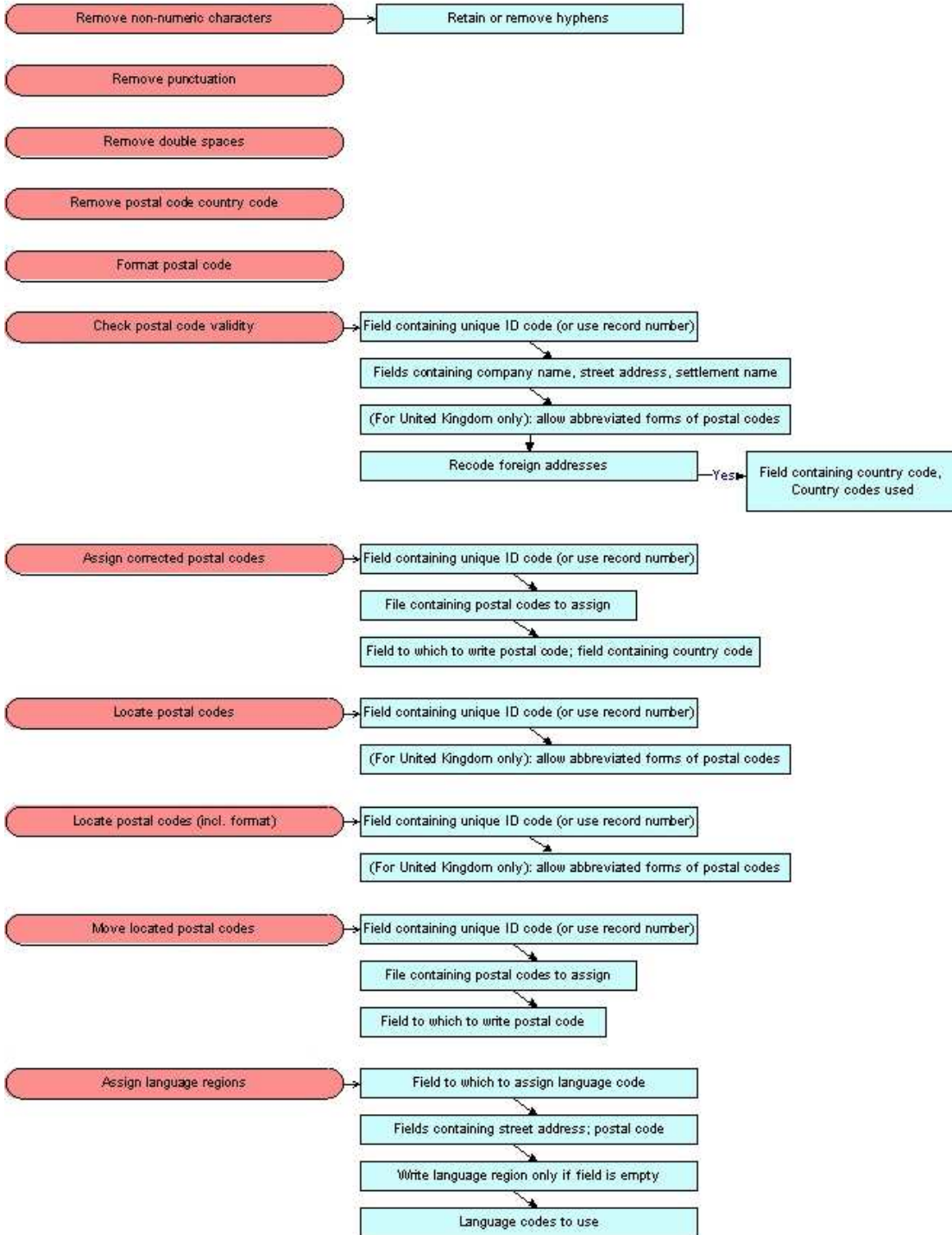


**IMPORTANT:** Address data should not normally be stored in numeric fields, and **GRC Tools™** will give an error if asked to process a numeric field. Please ensure that all fields being processed are of character/alphanumeric type

This graphical representation of the modules shows what information and parameters some of the modules require through **GRC Tools'** user interface:

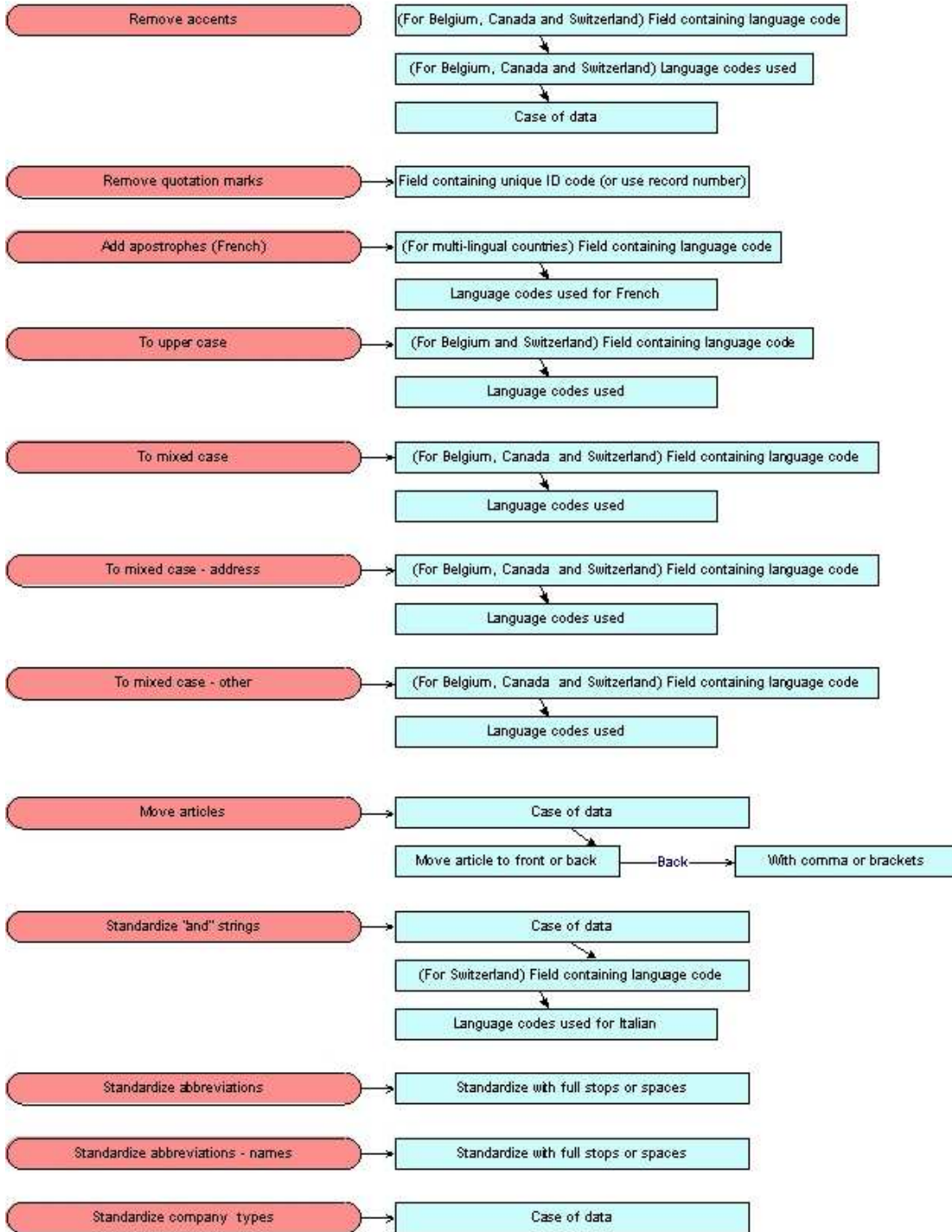
**Procedure**

**Parameters**



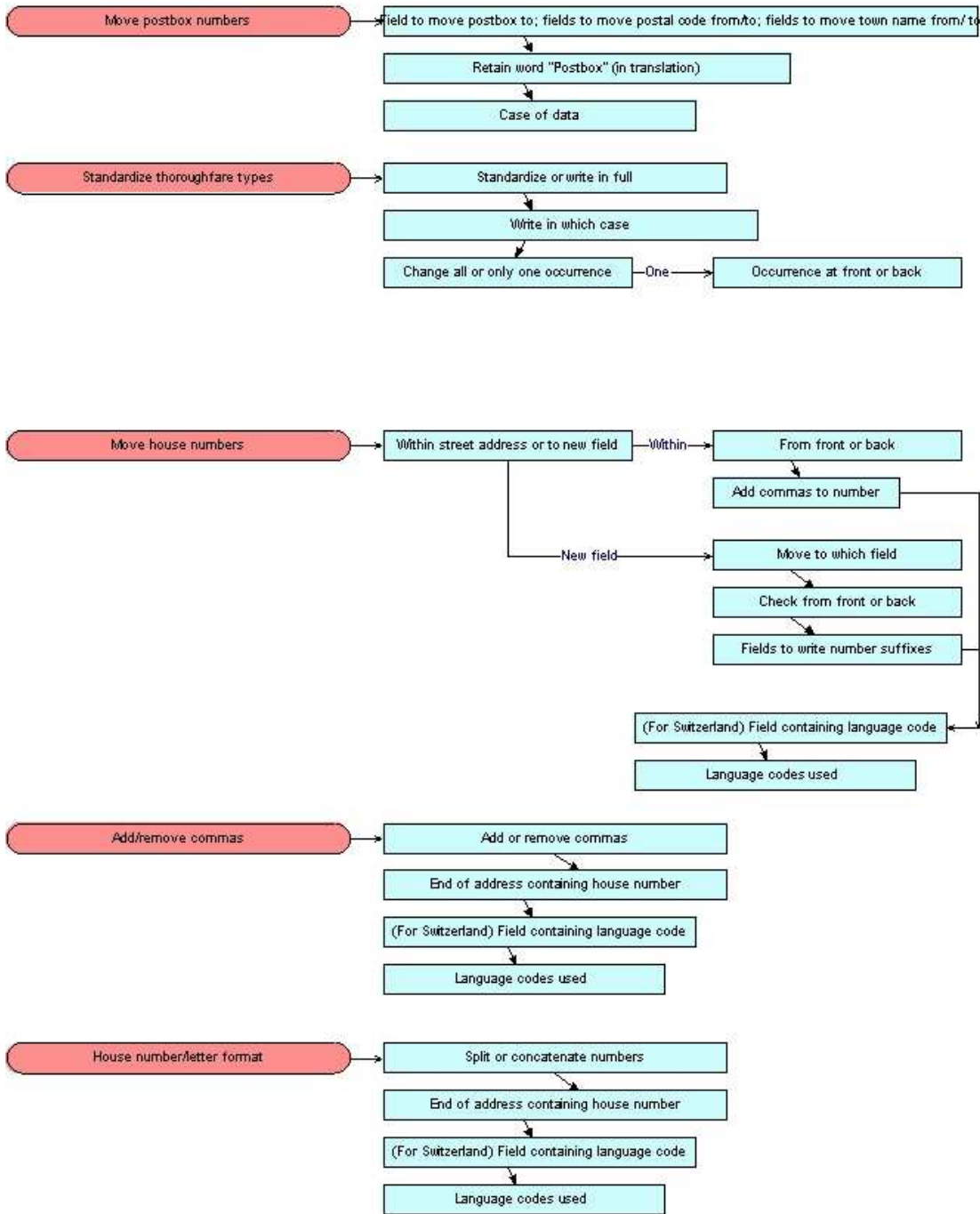
**Procedure**

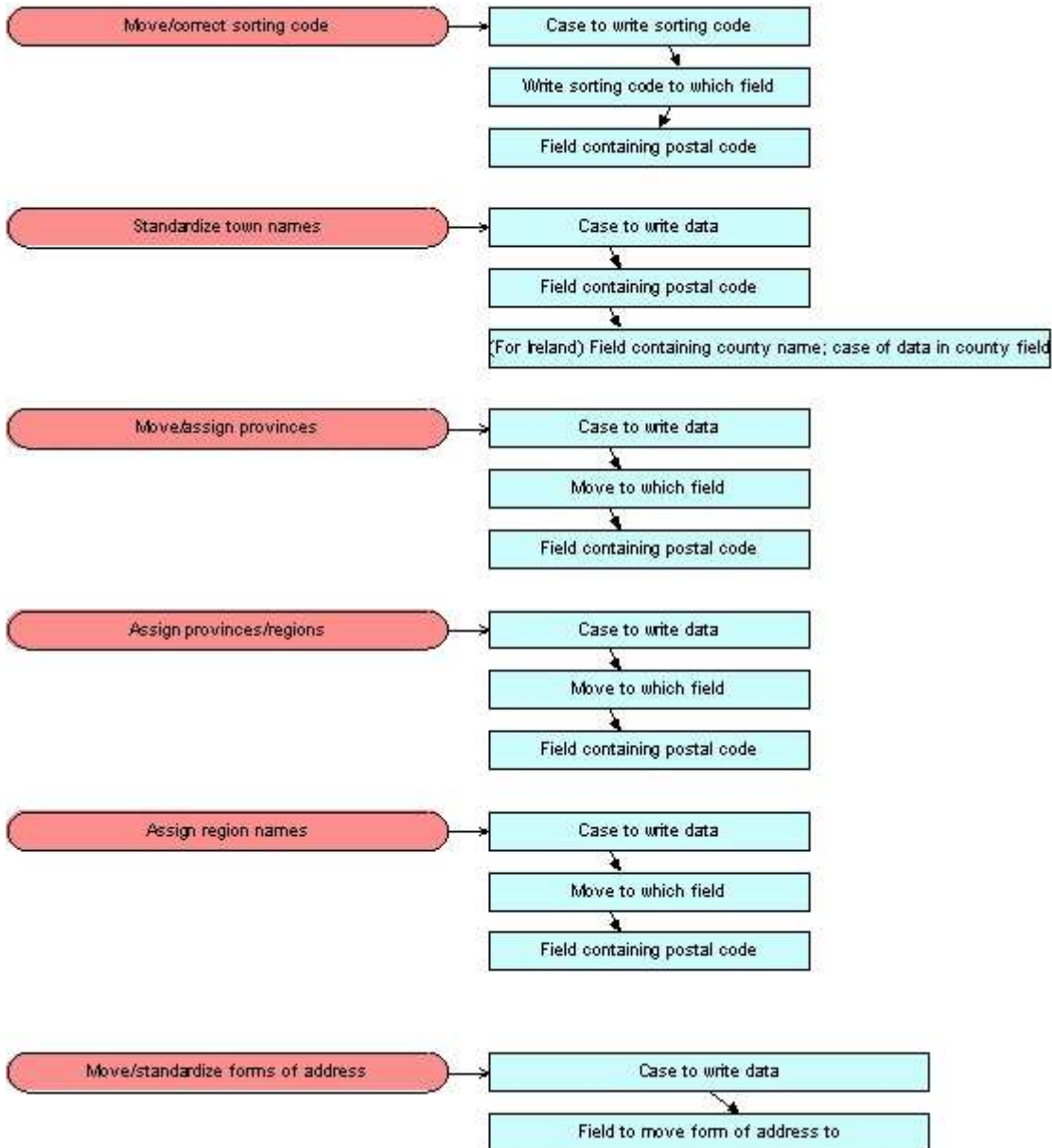
**Parameters**



**Procedure**

**Parameters**



**Procedure****Parameters**

## **8. Running GRC Country**

### **Running GRC Country™**



Run **GRC Country™** by double clicking on the **GRC Country™** icon, or by double clicking on GRCCOUNTRY.EXE in the GRCTools directory.

## 8.1. GRC Country - The menu

### **The Menu** Quit About

The **GRC Country™** menu has two main choices, and these will be discussed in order.

#### **Quit**

Exits the program

#### **About**

Brings up the **About** screen.

**8.2. GRC Country - Processing a file**

# GRC Country - Processing a file

On the main screen of **GRCCountry™**, choose the file type of the file to be opened.

These file options are currently available:

dBase III+  
 Lotus 1-2-3, 1-A  
 Lotus 1-2-3, 2.x  
 Lotus 1-2-3, 3.x  
 Lotus Symphony 1.01  
 Lotus Symphony 1.1  
 Microsoft Access  
 Microsoft Excel 2/3/4  
 Microsoft Excel 5.0  
 Microsoft Excel 97  
 Microsoft FoxBASE+  
 Microsoft FoxPro 2.x  
 Microsoft Multiplan 4.01  
 Microsoft Visual FoxPro  
 SQL Server

The native formats of **GRCCountry™** are Visual FoxPro/dBase III+. A data file which is in one of those formats will usually be processed without problems because no data translation has to be done. For all other data formats, **GRCCountry™** will export the file to Visual FoxPro format for processing and then export it again to the input format. If **GRCCountry™** has problems in exporting the data to Visual FoxPro format, it may produce errors. Please check on the list below to ensure that your file will process without problems:

- **All formats - file names** - an input file name must commence with a letter. A file commencing with a number or other character will return an error.
- **All formats - field lengths** - **GRCCountry™** supports field names of 10 characters or less. If you data contains field names which are longer, they are truncated when exported to Visual FoxPro format for processing. If this truncation produces duplicate field names, **GRCCountry™** will give an error. For example, if your file contains the field names:  
     address\_line\_1  
     address\_line\_2


these fields become truncated to

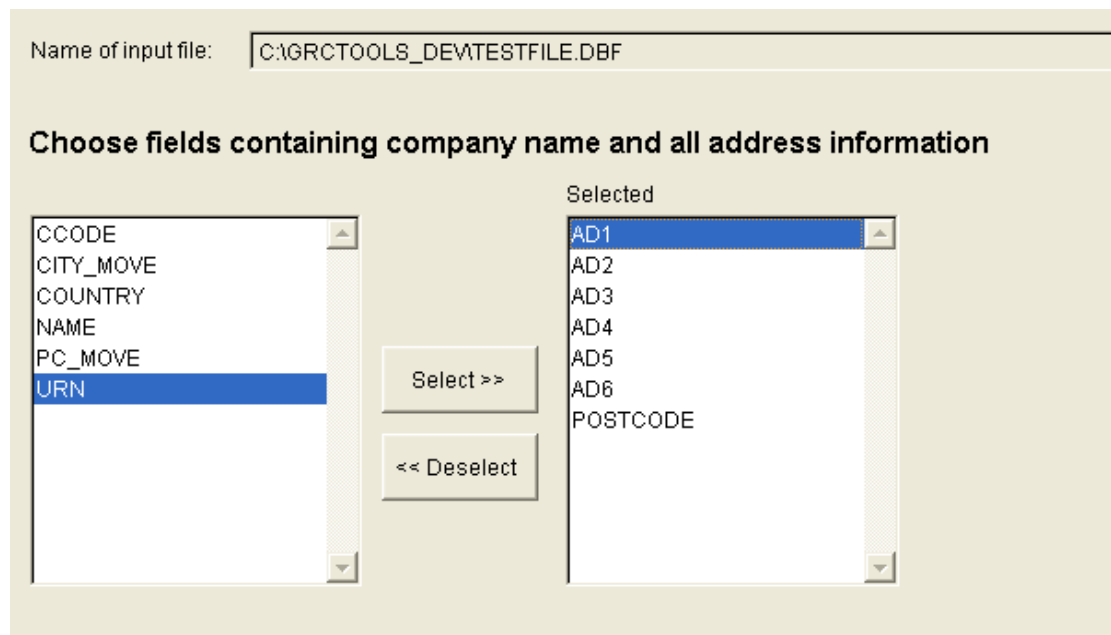
address\_li  
address\_li

The file then contains duplicate field names and **GRCCountry™** will return an error. Furthermore, no field may be longer than 250 characters.

- **Microsoft Excel** - Importing and exporting from Excel will usually corrupt diacritical marks (accented characters) in your data. This is a characteristic of Excel and not of **GRCCountry™**. If your data contains diacritical marks, you should not use Excel as an import format for **GRCCountry™**. Also, Excel may export fields which should be character fields (such as a postal code field) as a numeric field. Not only will **GRCCountry™** not process numeric fields, Excel will remove any zeros at the front of the data in numeric field, damaging your data.
- **Microsoft Access** - There are some major compatibility problems between Access and FoxPro, and the use of Access input files is best avoided. In any case, fields in Access files must have the "Allow zero length" option set to 'yes' to allow **GRCCountry™** to write data back to the file after processing. Furthermore, no field may be longer than 250 characters long, otherwise it is exported as an unexpected field type and cannot be processed.

Choose the file to be processed using the browse button , then press the  button to read the file.

 Next, choose the fields tab.



Name of input file:

**Choose fields containing company name and all address information**

Available Fields	Selected Fields
CCODE	AD1
CITY_MOVE	AD2
COUNTRY	AD3
NAME	AD4
PC_MOVE	AD5
URN	AD6
	POSTCODE

Buttons: Select >>    << Deselect

You are next required to specify the fields in the chosen data source which contain company name, address and postal code information. Include any fields which may include this data:

- Company name
- Street address
- Settlement name
- Region name
- Postal code
- Country name

As a minimum, **GRCCountry™** requires field(s) containing postal code and settlement name to process data.

To choose fields, double click on the field names in the left-hand column or highlight them and press the right arrow

button to select. To unselect a field, double click on the field name in the right-hand column or highlight it and press the left arrow button.

## Field contents

Next, choose the *Field Contents* tab.

The fields you have specified (may) contain this data: (please check all relevant)

- Country name
- Region/Province name
- Place name
- Postal code
- Address
- Company name
- E-mail address/URL

**GRCCountry™** needs to know what data to check for. As **GRCCountry™** requires as a minimum place name and postal code information, these boxes are checked and cannot be changed. If your data contains in some or all cases any of the other information, check the relevant boxes.

**IMPORTANT: E-mail address identification works on the basis of the two-letter Internet domain code, i.e. .de for Germany. In certain circumstances, this may be mistaken for other strings, such as the DE state code of Delaware. If in doubt, do not choose to identify countries on the basis of e-mail address/URL.**

## Coding type

Next, choose the *Coding type* tabs.

Choose country code system to use

- ISO 3166 - 2 letters
- ISO 3166 - 3 letters
- GRC Database Information

Write country code to which field?

Differential to accept as correct

Write differential to which field?

Write results token to which field?

This screen requires a number of parameters.

You need firstly to choose a country coding system for **GRCCountry™** to use. The choice is the ISO 3166 2 or 3 letter code, or the GRC Database Information internal code. The codes are:

Country name	ISO 2	ISO 3	GRC code
Afganistan	AF	AFG	AFG
Albania	AL	ALB	ALB

Algeria	DZ	DZA	ALG
American Samoa	AS	ASM	AMS
Andorra	AD	AND	AND
Angola	AO	AGO	ANG
Anguilla	AI	AIA	ANU
Antigua & Barbuda	AG	ATG	ANT
Argentina	AR	ARG	ARG
Armenia	AM	ARM	ARM
Aruba	AW	ABW	ARU
Australia	AU	AUS	AST
Austria	AT	AUT	AUS
Azerbaijan	AZ	AZE	AZE
Bahamas	BS	BHS	BAH
Bahrain	BH	BHR	BAR
Bangladesh	BD	BGD	BAN
Barbados	BB	BRB	BAB
Belarus	BY	BLR	BEO
Belgium	BE	BEL	BEL
Belize	BZ	BLZ	BEI
Benin	BJ	BEN	BEN
Bermuda	BM	BMU	BER
Bhutan	BT	BTN	BHU
Bolivia	BO	BOL	BOL
<b>Bonaire</b>			<b>BON</b>
Bosnia-Herzegovina	BA	BIH	BOS
Botswana	BW	BWA	BOT
Brazil	BR	BRA	BRA
British Virgin Islands	VG	VGB	BVI
Brunei Darussalam	BN	BRN	BRU
Bulgaria	BG	BGR	BUL
Burkina Faso	BF	BFA	BUK
Burma	MM	MMR	BUR
Burundi	BI	BDI	BUU
Cambodia	KH	KHM	CAM
Cameroon	CM	CMR	CAE
Canada	CA	CAN	CAN
Cape Verde Islands	CV	CPV	CAP
Cayman Islands	KY	CYM	CAY
Central African Republic	CF	CAF	CEN
Chad	TD	TCD	CHA
Chile	CL	CHL	CHI
China	CN	CHN	CHN
Christmas Island	CX	CXR	CHR
Cocos (Keeling) Islands	CC	CCK	COC
Colombia	CO	COL	CLO
Comoros	KM	COM	COM
Congo (Brazzaville)	CG	COG	CNG
Congo (Kinshasa)	ZR	ZAR	ZAI
Cook Islands	CK	COK	COO
Costa Rica	CR	CRI	COS
Croatia	HR	HRV	CRO
Cuba	CU	CUB	CUB
<b>Curaçao</b>			<b>CUR</b>
Cyprus	CY	CYP	CYP
Czech Republic	CZ	CZE	CZE
Denmark	DK	DNK	DEN
Djibouti	DJ	DJI	DJI

Dominica	DM	DMA	DOI
Dominican Republic	DO	DOM	DOM
East Timor	TL	TLS	ETI
Ecuador	EC	ECU	ECU
Egypt	EG	EGY	EGY
El Salvador	SV	SLV	ELS
Equatorial Guinea	GQ	GNQ	EQA
Eritrea	ER	ERI	ERI
Estonia	EE	EST	EST
Ethiopia	ET	ETH	ETH
Faeroe Islands	FO	FRO	FAE
Falkland Islands	FK	FLK	FAL
Fiji	FJ	FJI	FIJ
Finland	FI	FIN	FIN
France	FR	FRA	FRA
French Guiana	GF	GUF	FGU
French Polynesia	PF	PYF	FPO
Gabon	GA	GAB	GAB
Gambia, The	GM	GMB	GAM
Georgia	GE	GEO	GEO
Germany	DE	DEU	GER
Ghana	GH	GHA	GHA
Gibraltar	GI	GIB	GIB
Greece	GR	GRC	GRE
Greenland	GL	GRL	GRN
Grenada	GD	GRD	GRA
Guadeloupe	GP	GLP	GUD
Guam	GU	GUM	GUM
Guatemala	GT	GTM	GUA
Guernsey	GG		GUE
Guinea	GN	GIN	GUI
Guinea-Bissau	GW	GNB	GUB
Guyana	GY	GUY	GUY
Haiti	HT	HTI	HAI
Holy See	VA	VAT	VAT
Honduras	HN	HND	HON
Hong Kong	HK	HKG	HOK
Hungary	HU	HUN	HUN
Iceland	IS	ISL	ICE
India	IN	IND	IND
Indonesia	ID	IDN	INO
Iran	IR	IRN	IRA
Iraq	IQ	IRQ	IRQ
Ireland	IE	IRL	IRE
Isle of Man	IM		ISL
Israel	IL	ISR	ISR
Italy	IT	ITA	ITA
Ivory Coast	CI	CIV	IVO
Jamaica	JM	JAM	JAM
Japan	JP	JPN	JAP
Jersey	JE		JER
Jordan	JO	JOR	JOR
Kazakhstan	KZ	KAZ	KAZ
Kenya	KE	KEN	KEN
Kiribati	KI	KIR	KII
Kosovo			KOS
Kuwait	KW	KWT	KUW

Kyrgyzstan	KG	KGZ	KIR
Laos	LA	LAO	LAO
Latvia	LV	LVA	LAT
Lebanon	LB	LBN	LEB
Lesotho	LS	LSO	LES
Liberia	LR	LBR	LIR
Libya	LY	LBY	LIB
Liechtenstein	LI	LIE	LIE
Lithuania	LT	LTU	LIT
Luxembourg	LU	LUX	LUX
Macau	MO	MAC	MCA
Macedonia	MK	MKD	MCE
Madagascar	MG	MDG	MAD
Malawi	MW	MWI	MAW
Malaysia	MY	MYS	MAA
Maldives	MV	MDV	MAV
Mali	ML	MLI	MAI
Malta	MT	MLT	MAL
Marshall Islands	MH	MHL	MAR
Martinique	MQ	MTQ	MAN
Mauritania	MR	MRT	MAU
Mauritius	MU	MUS	MAT
Mayotte	YT	MYT	MAY
Mexico	MX	MEX	MEX
Micronesia	FM	FSM	MIC
Moldova	MD	MDA	MOL
Monaco	MC	MCO	MON
Mongolia	MN	MNG	MOG
Montenegro			MOE
Montserrat	MS	MSR	MOT
Morocco	MA	MAR	MOR
Mozambique	MZ	MOZ	MOZ
Namibia	NA	NAM	NAM
Nauru	NR	NRU	NAU
Nepal	NP	NPL	NEP
Netherlands Antilles	AN	ANT	NAN
Netherlands, The	NL	NLD	NET
New Caledonia	NC	NCL	NCA
New Zealand	NZ	NZL	NEW
Nicaragua	NI	NIC	NIC
Niger	NE	NER	NIE
Nigeria	NG	NGA	NIG
Niue	NU	NIU	NIU
Norfolk Island	NF	NFK	NOF
North Korea	KP	PRK	NKO
Northern Mariana Islands	MP	MNP	NMI
Norway	NO	NOR	NOR
Oman	OM	OMN	OMA
Pakistan	PK	PAK	PAK
Palau	PW	PLW	PAL
Panama	PA	PAN	PAN
Papua New Guinea	PG	PNG	PAP
Paraguay	PY	PRY	PAR
Peru	PE	PER	PER
Philippines	PH	PHL	PHI
Pitcairn Islands	PN	PCN	PIT
Poland	PL	POL	POL

Portugal	PT	PRT	POR
Puerto Rico	PR	PRI	PUE
Qatar	QA	QAT	QAT
Romania	RO	ROM	ROM
Russia	RU	RUS	RUS
Rwanda	RW	RWA	RWA
Réunion	RE	REU	REU
Saba			SAB
Saint Christopher & Nevis	KN	KNA	STC
Saint Helena	SH	SHN	STH
Saint Lucia	LC	LCA	STL
Saint Pierre & Miquelon	PM	SPM	SPM
Saint Vincent & The Grenadines	VC	VCT	STV
Samoa	WS	WSM	WSM
San Marino	SM	SMR	SAN
Sao Tome & Principe	ST	STP	SAO
Saudi Arabia	SA	SAU	SAU
Senegal	SN	SEN	SEN
Serbia	CS	SCG	YUG
Seychelles	SC	SYC	SEY
Sierra Leone	SL	SLE	SIE
Singapore	SG	SGP	SIN
Sint Eustatius			STE
Sint Maarten			SMA
Slovakia	SK	SVK	SLO
Slovenia	SI	SVN	SLV
Solomon Islands	SB	SLB	SOL
Somalia	SO	SOM	SOM
Somaliland			SOA
South Africa	ZA	ZAF	SAF
South Korea	KR	KOR	SKO
Spain	ES	ESP	SPA
Sri Lanka	LK	LKA	SRI
Sudan	SD	SDN	SUD
Suriname	SR	SUR	SUR
Swaziland	SZ	SWZ	SWA
Sweden	SE	SWE	SWE
Switzerland	CH	CHE	SWI
Syria	SY	SYR	SYR
Taiwan	TW	TWN	TAI
Tajikistan	TJ	TJK	TAJ
Tanzania	TZ	TZA	TAN
Thailand	TH	THA	THA
Togo	TG	TGO	TOG
Tokelau	TK	TKL	TOK
Tonga	TO	TON	TON
Trinidad & Tobago	TT	TTO	TRI
Tunisia	TN	TUN	TUN
Turkey	TR	TUR	TUR
Turkmenistan	TM	TKM	TUK
Turks & Caicos Islands	TC	TCA	TCI
Tuvalu	TV	TUV	TUV
Uganda	UG	UGA	UGA
Ukraine	UA	UKA	UKR
United Arab Emirates	AE	ARE	UAE
United Kingdom	GB	GBR	UNI
United States	US	USA	USA

United States Virgin Islands	VI	VIR	VIR
Uruguay	UY	URY	URU
Uzbekistan	UZ	UZB	UZB
Vanuatu	VU	VUT	VAN
Venezuela	VE	VEN	VEN
Vietnam	VN	VNM	VIE
Wallis & Futuna	WF	WLF	WAL
Western Sahara	EH	ESH	WSH
Yemen	YE	YEM	YEM
Zambia	ZM	ZMB	ZAM
Zimbabwe	ZW	ZWE	ZIM

You then need to specify the field to which the code is to be written. This must be a character/alphanumeric type field.

**GRCCountry™** processes addresses by giving a different weighting to the data in each record that it finds, and adding this up to reach a final assessment. For example, an address:

Bonnstrasse 17  
11705 BERLIN

may have weights assigned as follows:

Germany: City name and postal code combination: 100, street type: 5, postal code pattern: 15. Total: 120

France: Postal code pattern: 15. Total: 15

Austria: Street type: 5. Total: 5

And so on. The *differential* is the difference between the number assigned to the best fit country and the next one. In the example above, the difference between the best match (Germany, 120 points) and the second (France, 15) is 105 points.

However, if the same address is written with an incorrect postal code as follows:

Bonnstrasse 17  
1705 BERLIN

the assessment is much less definite. In this case the weights might become:

Germany: City name: 15, street type: 5. Total: 35

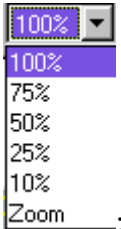
Austria: Postal code pattern: 15, Street type: 5. Total: 20




The *differential* is now only 15 points – the assessment is less certain.

The *Differential to accept as correct* spinner allows you to specify how much of a difference there should be between the top two possible matches to assign a code. We have found that a differential of 10 will give good results for good quality addresses, with full data including largely correct postal codes. For more definite assignments, a differential of 20 may be assigned – in this case fewer countries are found, but equally fewer errors are made. A differential of 30 may be used for very poor quality data. Higher than 30 or lower than 10 should not be used.

You may find it useful to know what differential **GRCCountry™** found in its processing. If you choose a field in the *Write differential to which field* dropdown, the differential number will be written to that field. **NOTE: this must be a numeric field. Specifying this field is optional – you may leave it blank.**



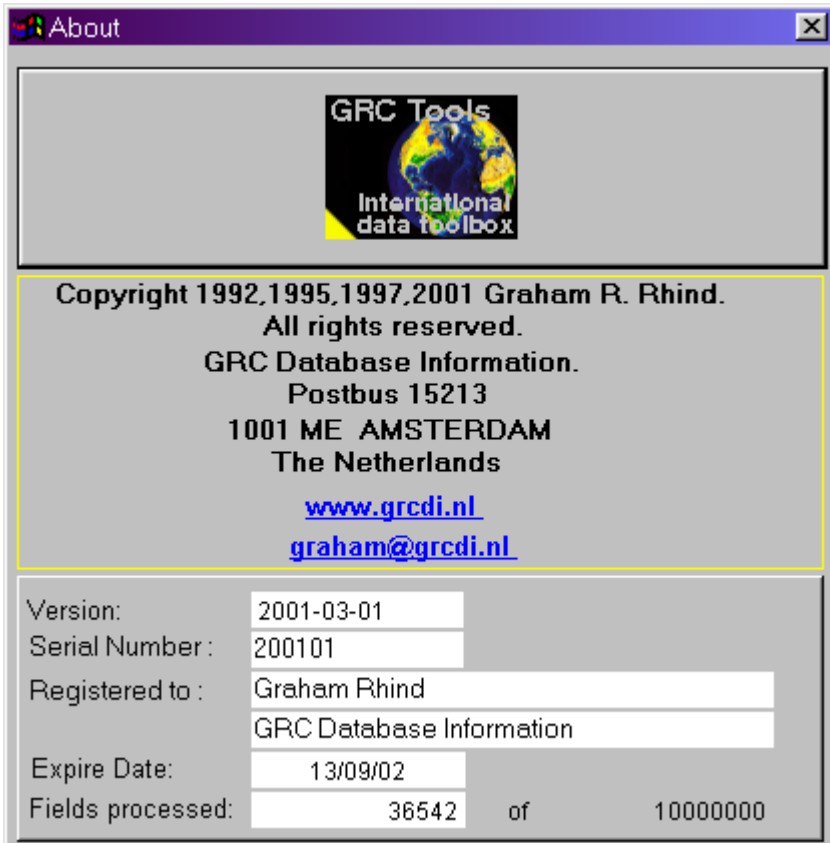


(the middle button allows you to skip to a specified page number); zoom in and out using this option: ; Exit the report without printing it ; or print the report .

**9. Running GRC Tools****Running GRC Tools™**

Run **GRC Tools™** by double clicking on the **GRC Tools™** icon.

Remember that **GRC Tools™** alters data. You should make a backup before starting any **GRC Tools™** processes.



When you start **GRC Tools™** you are shown the *About* screen. This screen shows you copyright information, the version number, the serial number, your registration number, the number of credits which you have used up to this point (*Fields processed*) and the upper number of credits which you may use. A software expiry date is also shown. This is, by default, 1 year from the date of purchase. However, you are not obliged to use you credits within this period - we simply want to encourage you to keep your software updated. When this period runs put, you can increase this period by further periods simply by e-mailing this request to [graham@grcdi.nl](mailto:graham@grcdi.nl).

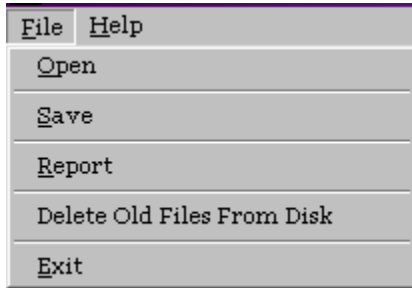
Press the button with the **GRC Tools™** logo to continue.




## 9.1. GRC Tools - The menu

### The Menu

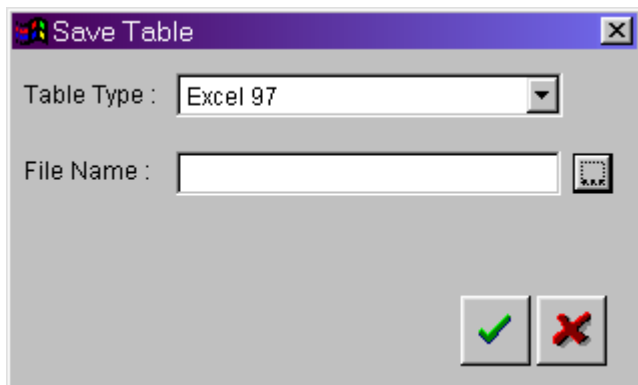
The **GRC Tools™** menu has two main choices, and these will be discussed in order.




### File > Open

This is the option, which may also be run using the open icon: , to choose to process a file. For further details, please refer to the next chapter: “Processing a file”.

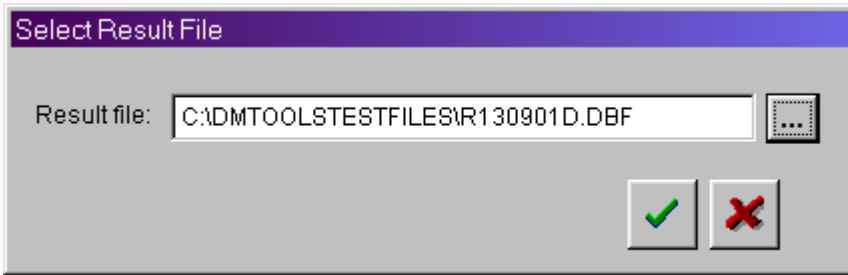
### File > Save




By default, **GRC Tools™** imports your data into its native format and then exports it again into the same format as it was originally. This option saves the last file processed, so you should choose this option only after you have processed the file you want to save in a different format.

This option, which is also available using the save button , allows you to save the last processed file in a different format. Specify the format in which you want to save the file, and the name to which to save it. This option should be used with caution if your file contains diacritical characters, as some file formats will corrupt diacritical characters upon import and export - Excel is a good example of this.

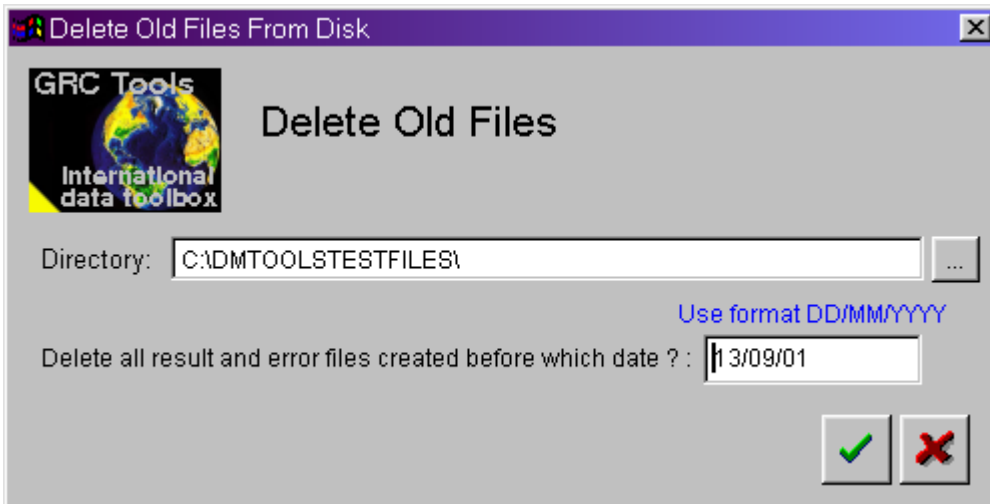
### File > Report





**GRC Tools™** writes results to report files for each run. If you wish to run a report from a previous run, this option

(also available using the print button ) can be chosen. Choose the results file using the browse button. Remember, results files are those with the format *RnnnnnnX.dbf*, where *n* indicates a number and *X* a letter, for example R130901D.DBF, this being the fourth results file of 13<sup>th</sup> September 2001. Choosing any other file type will result in a fatal program error.

## File > Delete Old Files From Disk

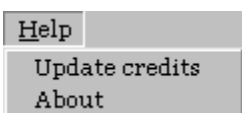


This option can also be chosen using the delete button . As old processing and results files take up valuable disk space, **GRC Tools™** allows you to delete them. Simply choose the directory using the browse button (remember, processing and results files are by default written to the directory where the input file is) and specify a date before which you would like all files deleted and press the tick button. Today's date is shown as the default. If you would like to add a different date, type it in the format show (dd/mm/yyyy).

 Files created on the date chosen will NOT be deleted

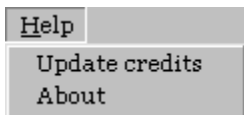
## File > Exit

Exit **GRC Tools™**.



## Help > Update credits

When your **GRC Tools™** are running out, you can purchase new credits by contacting us at [graham@grcdi.nl](mailto:graham@grcdi.nl). When you have made the purchase, we will send you a file called *data.dmt* which contains your new credits details. Choose this option and use the browse button to locate the *data.dmt* file. Press the tick button to update your credits information. This new information will be shown on the *About* screen.



## Help > About

Shows the about screen, with contact and credits details.

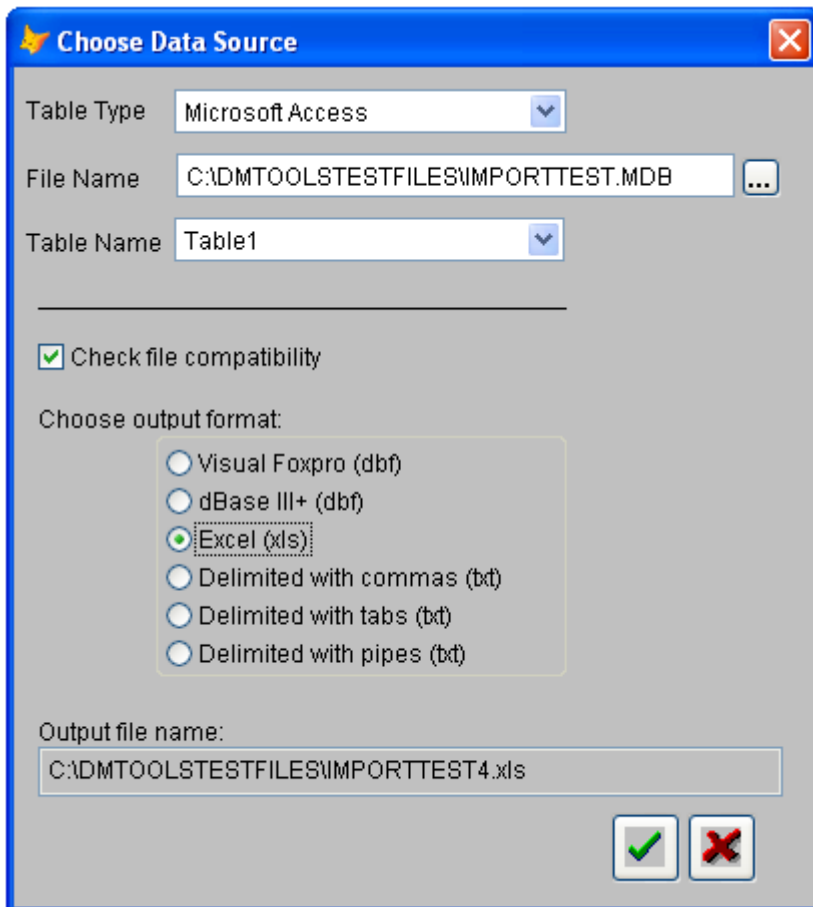
## 9.2. GRC Tools - Processing a file

# GRC Tools - Processing a file



When you choose the menu option **File > Open**, you are asked to specify which file you wish to process, and in which format the file is in. These file options are currently available:

- dBase III+
- Lotus 1-2-3, 1-A
- Lotus 1-2-3, 2.x
- Lotus 1-2-3, 3.x
- Lotus Symphony 1.01
- Lotus Symphony 1.1
- Microsoft Access
- Microsoft Excel 2/3/4
- Microsoft Excel 5.0
- Microsoft Excel 97
- Microsoft FoxBASE+
- Microsoft FoxPro 2.x
- Microsoft Multiplan 4.01
- Microsoft Visual FoxPro
- SQL Server



Choose the file to be processed using the browse button .

The native formats of **GRCTools** are Visual FoxPro/dBase III+. A data file which is in one of those formats will usually be processed without problems because no data translation has to be done. For all other data formats, **GRCTools** will export the file to Visual FoxPro format for processing and then export it again to the specified output format. There are some basic incompatibilities when some programs export to dbf format. Access, for example, may export character fields with a length of 256 (which cannot then be indexed), or as memo fields; whilst Excel may export columns as numeric fields, which **GRCTools** will not process. Unless you are working natively in Foxpro we therefore recommend that you choose the "Check file compatibility" option to change numeric and memo fields to character fields and to reduce the length of character fields to 100. For larger files this check may take some time, but it prevents errors further on in the program.

Choose your output format. Input in .dbf format will be output with the same name (i.e. **GRCTools** will work directly on the input file). For other formats a new file is created (named after the original file with a number between 1 and 99 added to the file name).

If **GRCTools** has problems in exporting the data to Visual FoxPro format, it may produce errors. Please check on the list below to ensure that your file will process without problems:

- **Windows NT** - An error is known to occur on some Windows NT systems where the c:\temp\ directory is a hidden system directory. **GRCTools** needs to write temporary files to this directory during processing. If the directory is hidden, **GRCTools** cannot locate the directory so it tries to create it, which it cannot do, leading to this error. To resolve this, the c:\temp\ directory needs to be given the same properties as other "standard" directories, either by changing its properties, or by re-naming it and creating a new, "normal" c:\temp\ file. If this is not possible, install **GRCTools** on a computer using a different Windows version.
- **All formats - file names** - an input file name must commence with a letter. A file commencing with a number or other character will return an error.
- **All formats - field lengths** - **GRCTools** supports field names of 10 characters or less. If your data contains field names which are longer, they are truncated when exported to Visual FoxPro format for processing. If this truncation produces duplicate field names, **GRCTools** will give an error. For example, if your file contains the field names:

```
address_line_1
address_line_2
```

these fields become truncated to

```
address_li
address_li
```

The file then contains duplicate field names and **GRCTools** will return an error. Furthermore, no field may be longer than 250 characters.

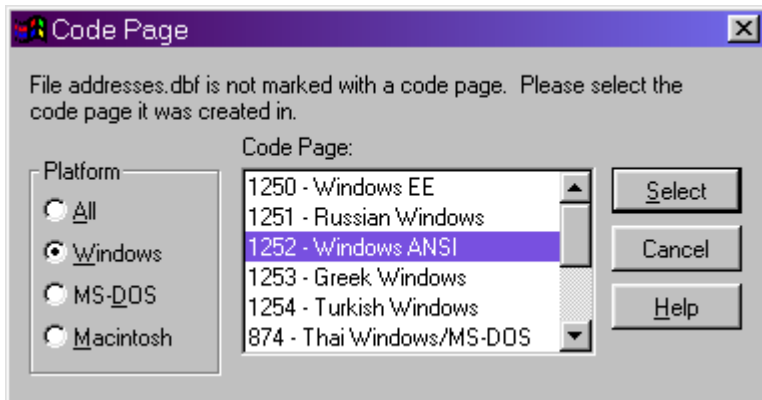
- **SQL Server:** Ensure that field names in the original SQL Server database are 10 characters or less. As longer field names are truncated in **GRCTools**, the SQL insert commands to return data after processing to SQL Server format fail.
- **Microsoft Excel** - Importing and exporting from Excel will usually corrupt diacritical marks (accented characters) in your data. This is a characteristic of Excel and not of **GRCTools**. If your data contains diacritical marks, you should not use Excel as an import format for **GRCTools**.
- **Microsoft Access** - There are some major compatibility problems between Access and FoxPro, and the use of Access input files is best avoided. In any case, fields in Access files must have the "Allow zero length" option set to 'yes' to allow **GRCTools** to write data back to the file after processing.

Press the tick button to continue.

Most of the world's languages contain diacritical characters, and data is added to databases which are marked with a particular code page, depending on the language covered. Not all languages can be covered with a single code page. The lookup tables for **GRC Tools™** have been based on ANSI code page 1252. This means that the following accented characters have been used to build the lookup tables:

Ä, Å, Á, Â, Ã, Ä, Å, â, ä, à, á, ã  
 Ç, ç  
 È, É, Ê, Ë, È, é, ê, ë, è  
 Í, Î, Ï, Ì, ï, î, í, í  
 Ñ, ñ  
 Ö, Ø, Ó, Ô, Ò, Õ, ô, ö, ò, ø, ó, õ,  
 ß  
 Ð, ð, Þ, þ  
 Û, Ú, Û, Ù, ü, û, ù  
 Ý, ÿ, ý,  
 Æ, æ

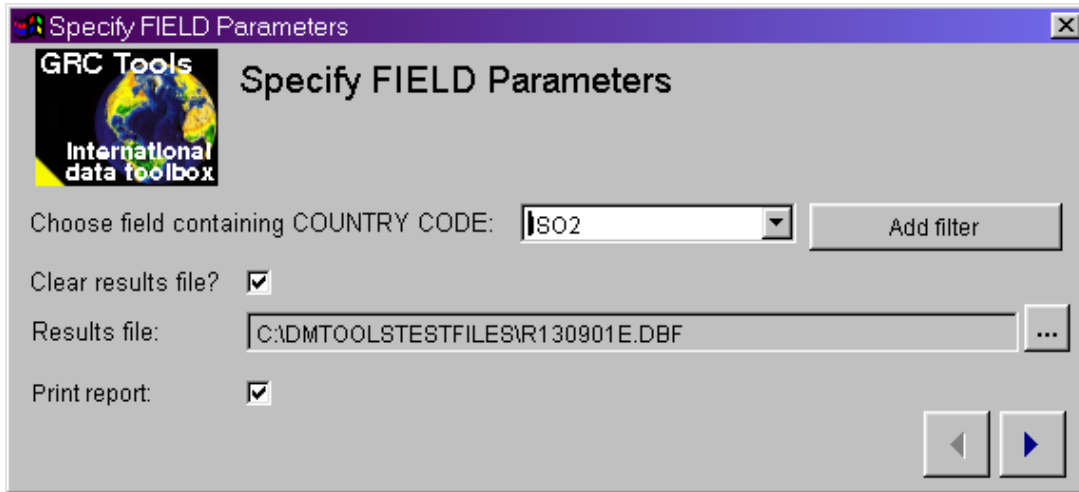
In order to correctly assign information with accented characters and to prevent corruption of incoming data, it is necessary to specify at this point the code page in which the original data was created:



Highlight the appropriate code page information and press *Select* to continue.

**Important:** **GRC Tools™** lookup tables are in Windows ANSI 1252 code page format. Once your file has been processed once by **GRC Tools™**, its data will be stored also in Windows ANSI 1252 format. This being the case, if you process a file more than once in **GRC Tools™**, you need to specify that the data is in Windows ANSI 1252 format for each processing from the second time onwards. If you specify the original file code page after the first time of processing, you will corrupt diacritical characters written in previous processing by **GRC Tools™**.

**GRC Tools™** now asks for certain parameters to be able to format your data correctly. These parameters will be remembered by **GRC Tools™** so you rarely have to change them if your database structures and naming conventions remain constant. You are required to specify the field in which the country code is situated. You must have a filled country code field in your database, even if all the data in the database is from the same country.



A results file called *RddmmyyX.dbf*, where *dd* is the date, *mm* the month and *yy* the year, and *X* is an incremental letter is automatically created by **GRC Tools™**. For every session run on that date, a new report file is automatically created. You may alternatively request that the data is written to a different results file (e.g. one from a previous date or previous session) by choosing this file from the *Results File* browse button. In this case, results are appended to this file. If you wish to delete the contents of this file before commencing a session, check the *Clear Results File?* box. Checking *Print report* will give you an option at the end of the processing to print a report.

! If you should do more than 26 runs in a single day and have thus created results files ending in all letters between A and Z, A is reused for all remaining sessions for that date.

! The results file is created in Visual Foxpro format and in Excel format.





Countries have been listed in English. For the sake of clarity, please note that:

- *Kampuchea* is listed as *Burma*
- *Khmer* is listed as *Cambodia*
- *Congo* (north of the river Congo) is listed as *Congo - Brazzaville*
- *The Democratic Republic of Congo* (south of the river Congo, ex-Zaire) is listed as *Congo - Kinshasa*

You are next required to specify from which countries the data in your input database comes (or which countries you wish to process). Double click on the country names in the left-hand column or highlight them and press the right arrow button to select a country. If you are unsure of all of the countries within the database, then check *All Countries*. **GRC Tools™** will then identify all unique country codes within your input database.

By default, **GRC Tools™** provides you with the flexibility to define which code in your database agrees with which country. However, to save time for users who use the ISO3166 2- or 3-character country codes in their data, choosing “ISO 3166-2” or “ISO 3166-3” will make **GRC Tools™** automatically assign these codes, saving you time. You will be able to review the links made in the next window. If you do not use an ISO coding system, maintain the choice as “Define”.

If you have previously saved a file containing country codes for a specified project, you can import those codes by using the *Load country code file* button. These files must be stored in the /lu/ directory under the **GRC Tools™** main program directory, and must have a name commencing *dmtollu\_*.

For each country chosen, **GRC Tools™** needs to make a link between the country code used within the input database and the country names, so that the correct processing is done per country. **GRC Tools™** remembers links made, so if your country codes remain stable, this process will only have to be done for each country once. Known links are presented in the bottom box. Country codes not yet known are presented in the top left-hand box.

Highlight each code, highlight the country name to which it should be linked in the top right-hand box, and choose

Link

To clear any highlighting you have done in error, choose

Clear

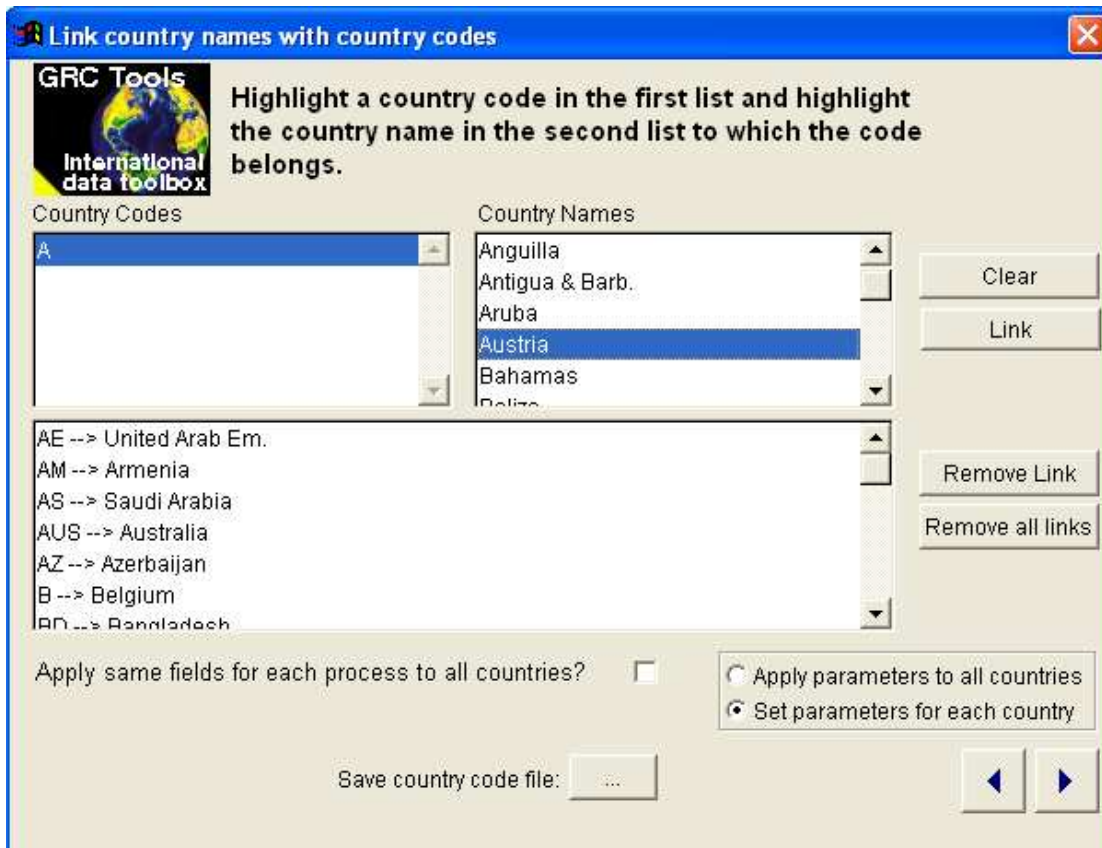
To re-link an incorrect link (or to prevent **GRC Tools™** from processing a given country during this session), highlight a made link in the bottom box and choose

Remove Link

Choose

Remove all links

to clear all made links and re-start the linking process.



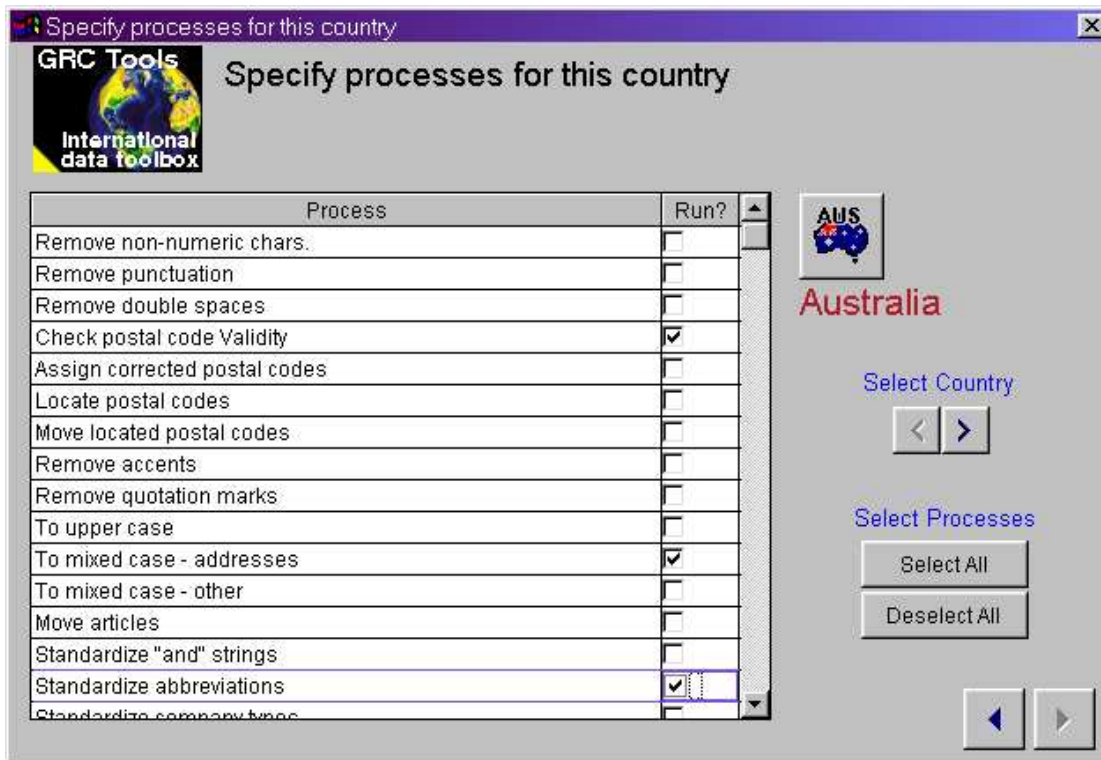
Uncheck the box *Apply same fields for each process to all countries* if the fields for each process differ by country. If you check this box, each process will be automatically assigned the same fields for processing as the first country for which you assigned fields for this country and this session, though you have the chance to change them. Thus, if you choose to run processes over **Belgium, France** and **Germany**, and this box is checked, then if you choose the fields *Address\_1* and *Address\_2* for Belgium for a process, these fields will be automatically filled in for this same process for France and Germany, though you may alter them as you wish.

Choose *Apply parameters to all countries* if the same process is to be run over the same fields for each country. If you choose this option, you will only be asked for process parameters once. However, there are a number of processes where choosing this option may affect quality of processing, as described in the individual process chapters below. To set parameters separately for each country within the database, choose instead *Set parameters for each country*.

To save these country codes for this project, use the *Save country code file* button. The file must be saved to the `/lu/` directory under the main **GRC Tools™** program directory, it must have a name which commences with *dmtollu\_*, and it may not be called *dmtollu* (i.e. without the final underscore).

For each country chosen (or for *All* if the *Apply parameters to all countries* option was chosen), you need now to specify which processes to run. The processes that you previously run for this country are already checked. This is asked per country in alphabetical order, only those processes available for the given country being listed, and the processes are listed in the order that they are run. This order has been carefully constructed for each country to increase accuracy. For example, as many processes require a correct postal code to run, the postal code validity would need to be checked before these processes are run. Be aware of what each process will do to your data, as this may affect parameters specified for processes to be run later. I.e., if your data is in upper case, and you choose to change this to mixed case, then any processes run after the mixed casing option that need to know the case of the data in the field will need to have *mixed case* specified instead of *upper case*.

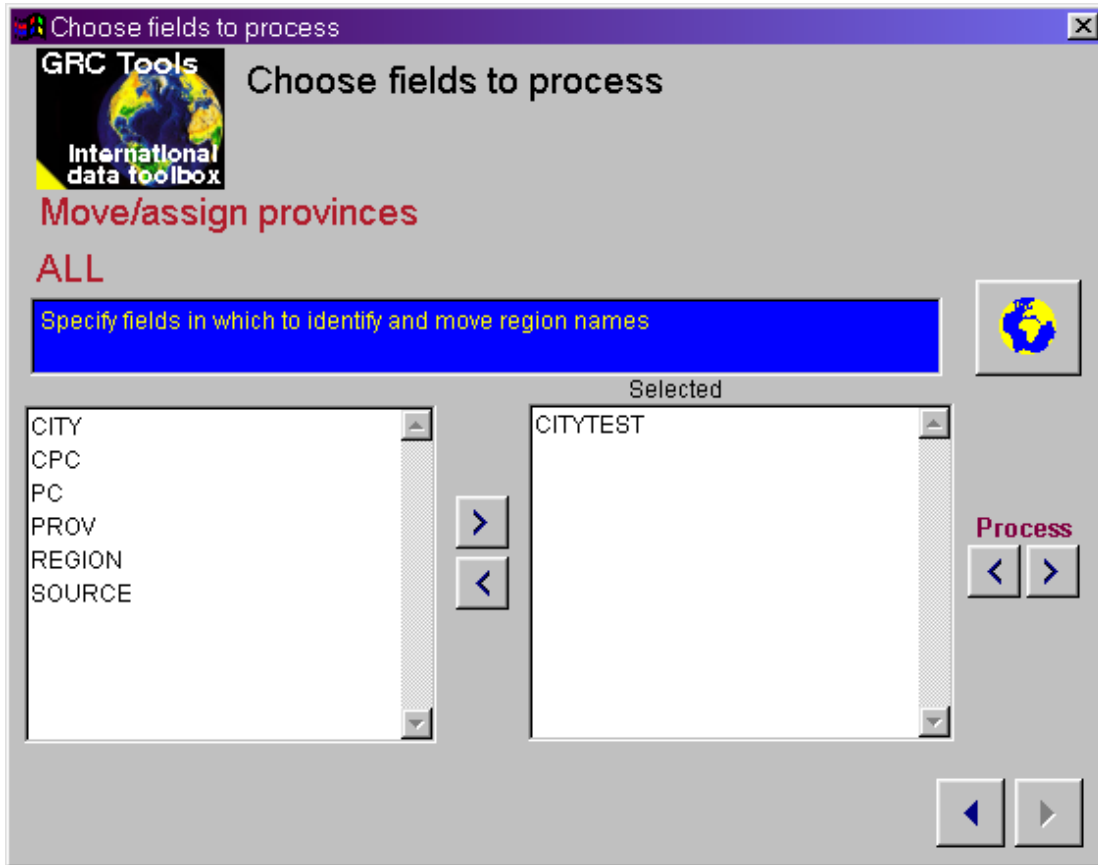
Choose as many processes as you need. Choose *Select all* to select all processes, *Deselect all* to clear all current selections. Move from country to country using the arrow keys under the text *Select country* until all countries are completed.



The processes **Write Located Postal code**, **Assign Language Region** and **Assign Provinces** do not require a field to work on, as they are assigning information rather than changing the contents of a field. For this reason, you are not asked to select fields for these processes.

For each process chosen, you can now specify the names of the fields over which each process is to be run. Double click on the field name in the left-hand column to select a field, moving it to the selected (right-hand) column. Alternately, highlight the field and press the right arrow key between the selection columns. To un-select a field, double-click it in the right-hand column to move it back to the un-selected (left-hand) column, or highlight the field and press the left arrow button between the selection columns.

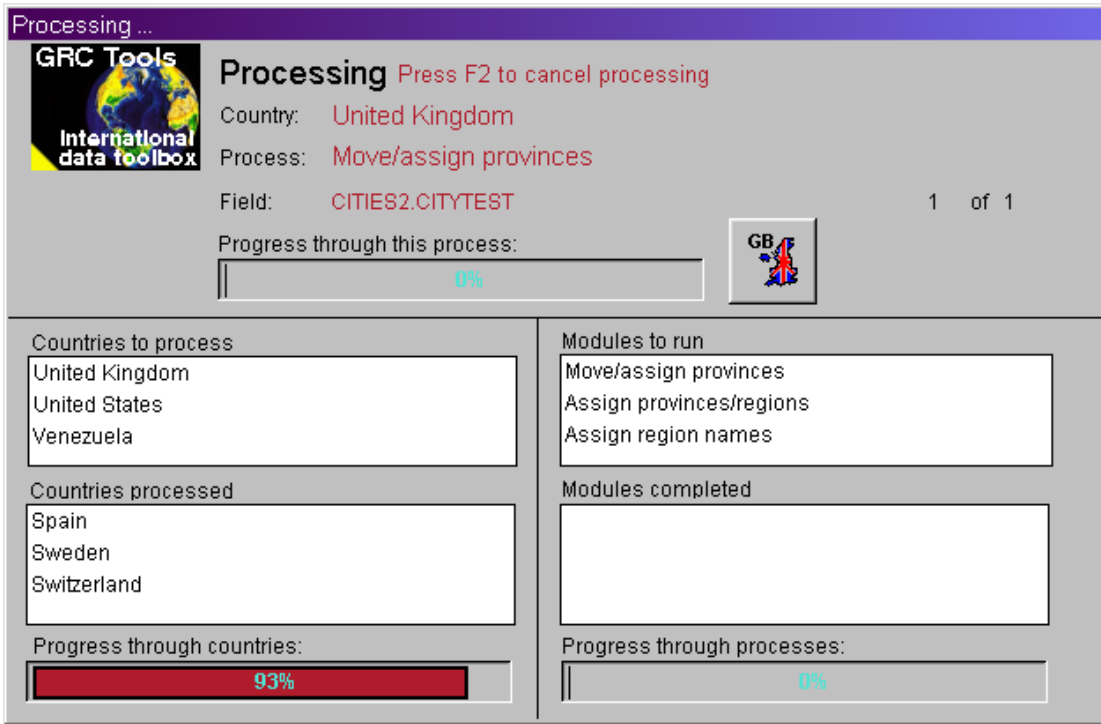
Using the arrow keys to the right of the columns, move between the processes selecting fields until all processes have been completed. As you move between the processes, depending on the processes chosen, a number of extra parameter questions may be asked - these are described in detail in the process chapters.



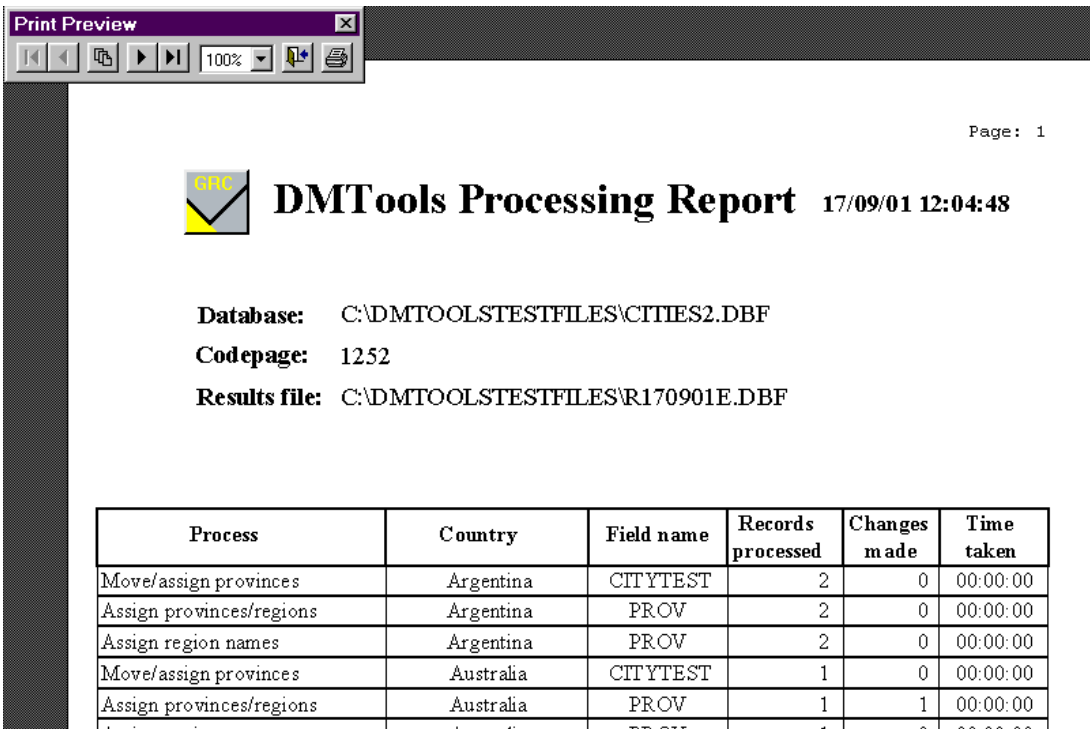
At the beginning and end of each process by country, **GRC Tools™** processes its built-in lookup tables. For processes and countries where these lookup tables are large, **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

**GRC Tools™** will now start processing your data. This screen keeps you updated on progress. You can see which process is running over which field for which country, which countries have been completed (and which are still to be done), and which processes have been completed for this country (and which are still to be done).

Press F2 at any time to interrupt processing. It is important to avoid interrupting **GRC Tools™** in any other way, for example by turning off your computer during processing. This can cause corruption of the lookup tables and can prevent **GRC Tools™** from working.

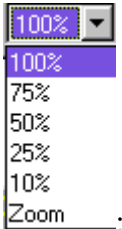





After processing, if you requested a report, **GRC Tools™** will present the report to you on the screen.



This report lists the processes fulfilled per country, the field(s) processed, the number of records processed, the number of records which were changed and the (rounded) time taken to run each process over each field for that country.

Using the floating menu bar, you can scroll through the pages of the report using these buttons



(the middle button allows you to skip to a specified page number); zoom in and out using this option: ; Exit the report without printing it ; or print the report . If you choose not to print the report at this time, **GRC Tools™** allows you to print it another time using the **File > Report** menu option. e number of records which were changed and the (rounded) time taken to run each process over each field for that country.

**9.3. The results file**

## The results file

A results file called *RddmmyyX.dbf*, where *dd* is the date, *mm* the month and *yy* the year, and *X* is an incremental letter is automatically created by **GRC Tools™**. For every session run on that date, a new report file is automatically created.

This file is created in Visual Foxpro and in Excel formats.

process	country	date_start	date_end	time_start	time_end	field_name	changed	records
Parse university strings	United Kingdom	13-Dec-07	13-Dec-07	11:45:38	11:45:49	GRCUNI.COMP1	0	2053
Parse university strings	United Kingdom	13-Dec-07	13-Dec-07	11:45:49	11:45:53	GRCUNI.TEST1	30	3094
Parse university strings	United Kingdom	13-Dec-07	13-Dec-07	11:45:53	11:45:59	GRCUNI.TEST2	69	2555
Parse university strings	United States	13-Dec-07	13-Dec-07	11:46:01	11:54:45	GRCUNI.COMP1	7264	22868
Parse university strings	United States	13-Dec-07	13-Dec-07	11:54:45	11:56:06	GRCUNI.TEST1	398	26894
Parse university strings	United States	13-Dec-07	13-Dec-07	11:56:06	11:57:10	GRCUNI.TEST2	250	17319

The fields contain this information:

**PROCESS:** The name of the process run.

**COUNTRY:** The country for which the process was run.

**DATE\_START:** The date upon which the main processing loop for this country and process commenced.

**DATE\_END:** The date upon which the main processing loop for this country and process exited.

**TIME\_START:** The time at which the main processing loop for this country and process commenced.

**TIME\_END:** The time at which the main processing loop for this country and process exited.

**FIELD\_NAME:** The name of the field over which the process was run.

**CHANGED:** The number of fields where changes were made by the process.

**RECORDS:** The number of fields containing data over which the process was run. Fields without data are not counted.

**CHANGES:** The number of changes made within the fields. Not counted for all processes.

**PCLONG:** For processes checking postal code validity, this indicates the number of postal codes which have more characters than they should.

**PCCOUN:** For processes checking postal code validity, this indicates the number of foreign postal codes identified.

**PCSTRINGS:** For processes checking postal code validity, this indicates the number of postal codes containing invalid strings.

**ASSIGNED:** For processes which assign data, this indicates the number of strings assigned.

**IDENT:** For processes parsing postal codes, this indicates number of fields with codes identified.

**EMPT:** Not used

**NOIDENT:** For processes parsing postal codes, this indicates number of fields with no codes identified.

**NOLU:** For processes checking postal code validity, this indicates that no lookup table exists for this country.

**DATABASE:** Name of database table being processed.

**FILTER:** Filter, if set.

**10. The Processes****Summary of the modules**

**GRC Tools™** works on standardising names and addresses on a modular basis. Below is a summary of the modules, in the order in which they should be run for optimal quality, and the type of data upon which they work:

**Data processing**

<i>Process</i>	<i>Field contents</i>
Invalid strings - move field contents	Any alpha-numeric field
Field - split string	Any alpha-numeric field
Field - trim leading spaces	Any alpha-numeric field
Field - trim final string	Any alpha-numeric field
Non-numeric characters - remove	Any alpha-numeric field
Punctuation - remove	Any alpha-numeric field
Double spaces - remove	Any alpha-numeric field
Telephone numbers - burst	Any alpha-numeric field
Postal code country code - remove	Postal code
Thoroughfare types - parse	Any alpha-numeric field
Format postal code	Postal code
Postal codes - assess validity	Postal code
Postal codes - write corrected	Postal code
Postal codes - locate	Postal code, any alpha-numeric field likely to contain a stray postal code
Postal codes (incl. format) - locate	Postal code, any alpha-numeric field likely to contain a stray postal code
Postal codes - write located	Postal code
Language regions - assign	Uses postal code and street address data
Accents - remove	Any alpha-numeric field
Quotation marks - remove	Any alpha-numeric field
Apostrophes (French) - add	Any alpha-numeric field
Casing - to upper case	Any alpha-numeric field
Casing - to mixed case	Any alpha-numeric field
Casing - to mixed case - address	Street address data
Casing - to mixed case - other	Any alpha-numeric field, not street address
Articles - move or parse	Company name
"And" strings - move or parse	Company name
Abbreviations - standardise	Company name
Abbreviations in personal names - standardise	Personal name
Company types - standardise	Company name
Sub-building information - parse & standardise	Any alpha-numeric field
Post office box numbers - parse	Any alpha-numeric field which could contain stray post office box numbers, usually street address fields
Thoroughfare strings - standardise	Street address data
Building numbers - move or parse	Street address data
Building number suffix - parse	House number data

Commas - add or remove	Street address data
Building number/letter - format	House number/street address data
Sorting code - parse & standardise	Any alpha-numeric field which could contain stray sorting codes, usually postal town fields
Place names - parse & standardise	Any alpha-numeric address field likely to contain a settlement name
Place names - standardise	Settlement name
Provinces - parse/assign	Any alpha-numeric field which could contain stray province names
Provinces/regions - assign	Uses postal code data
Regions - assign	Uses postal code data
University strings - parse	Any alpha-numeric field
Attention strings - move field contents	Any alpha-numeric field
Department strings - move field contents	Any alpha-numeric field
Department strings - move field contents	Any alpha-numeric field
Department strings - standardise	Any alpha-numeric field
Personal name prefixes - parse & standardise	Any alpha-numeric field which could contain stray forms of address data, usually personal name fields
Personal name middle name(s)/initial (s) - parse	Any alpha-numeric field
Job titles - standardise	Job title

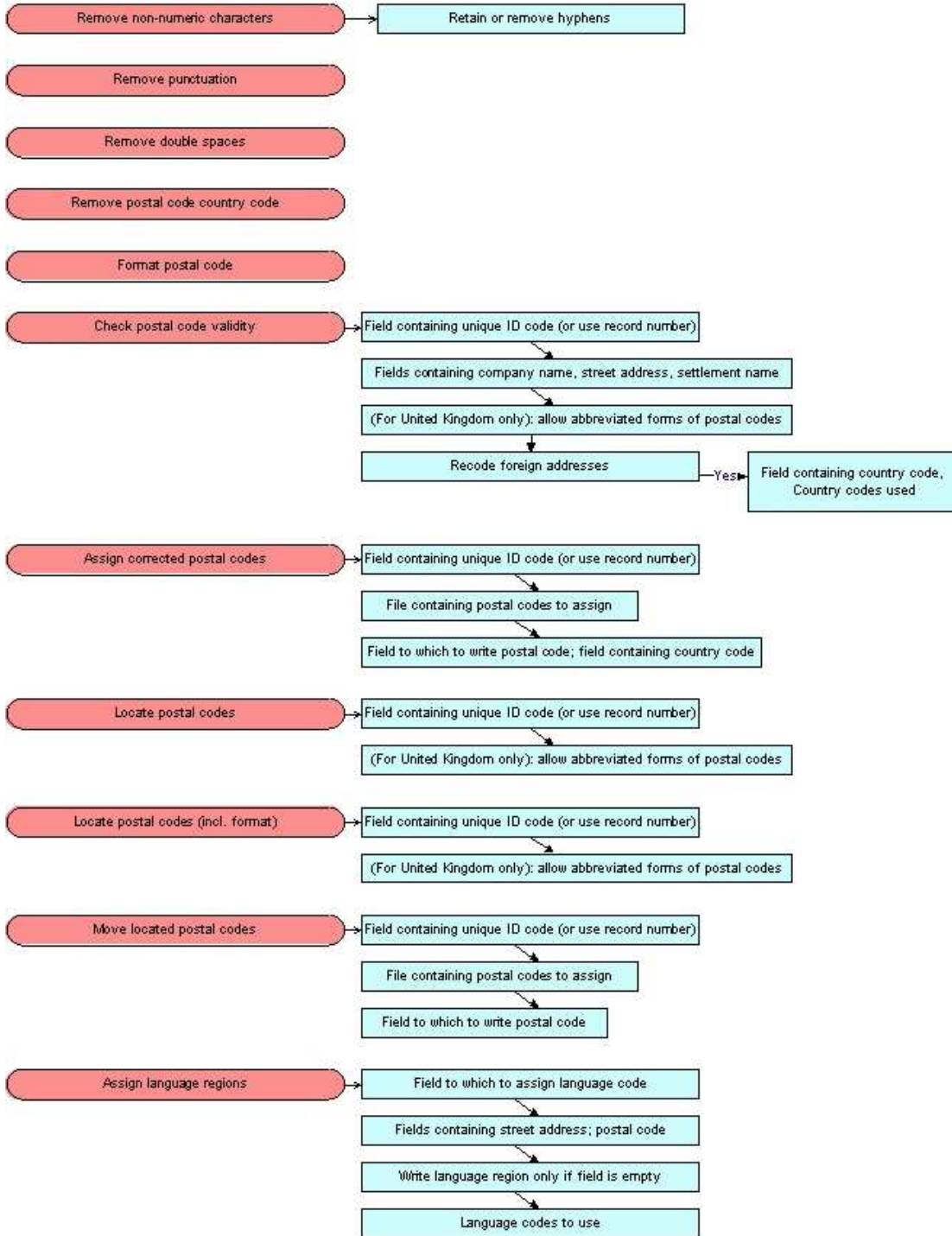


**IMPORTANT:** Address data should not normally be stored in numeric fields, and **GRC Tools™** will give an error if asked to process a numeric field. Please ensure that all fields being processed are of character/alphanumeric type

This graphical representation of the modules shows what information and parameters some of the modules require through **GRC Tools'** user interface:

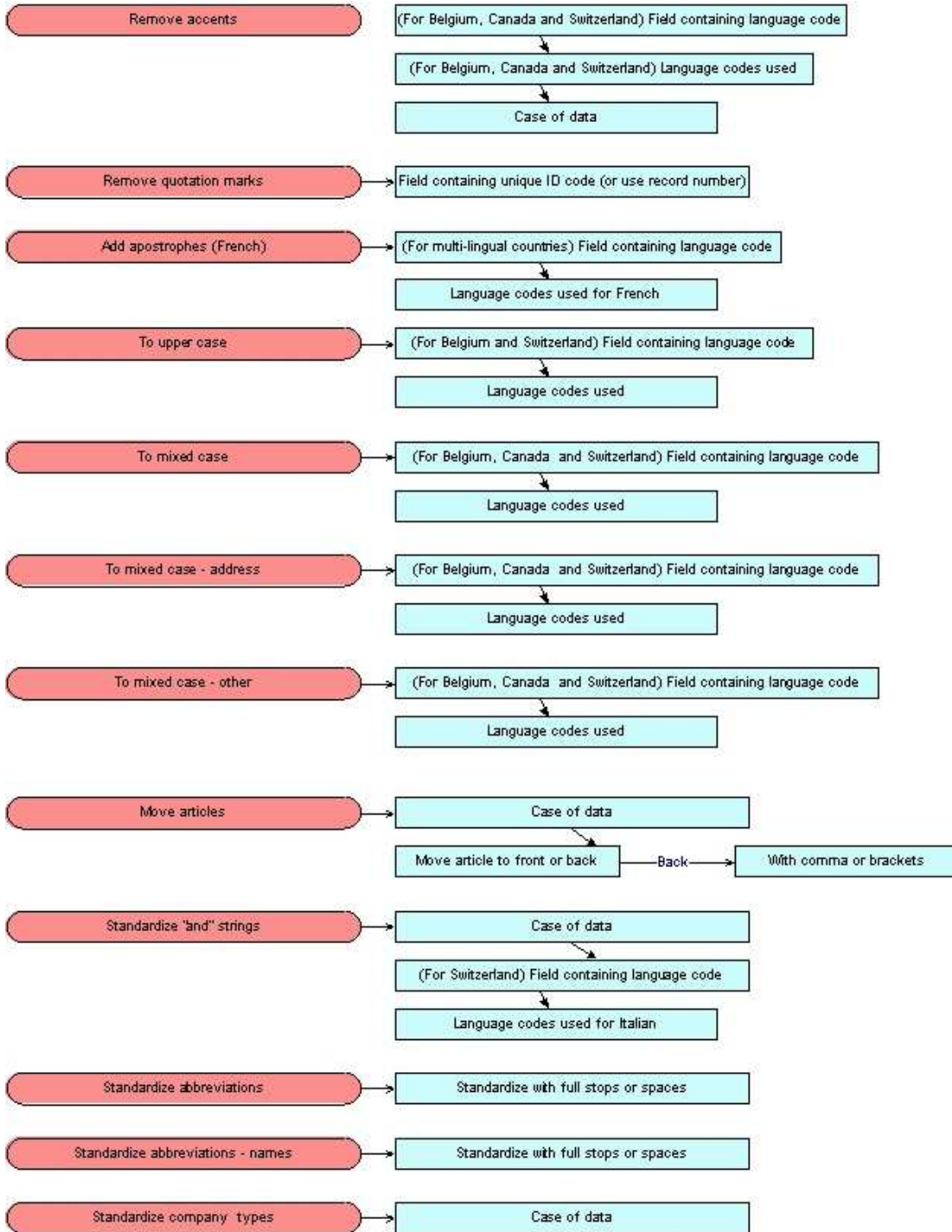
**Procedure**

**Parameters**



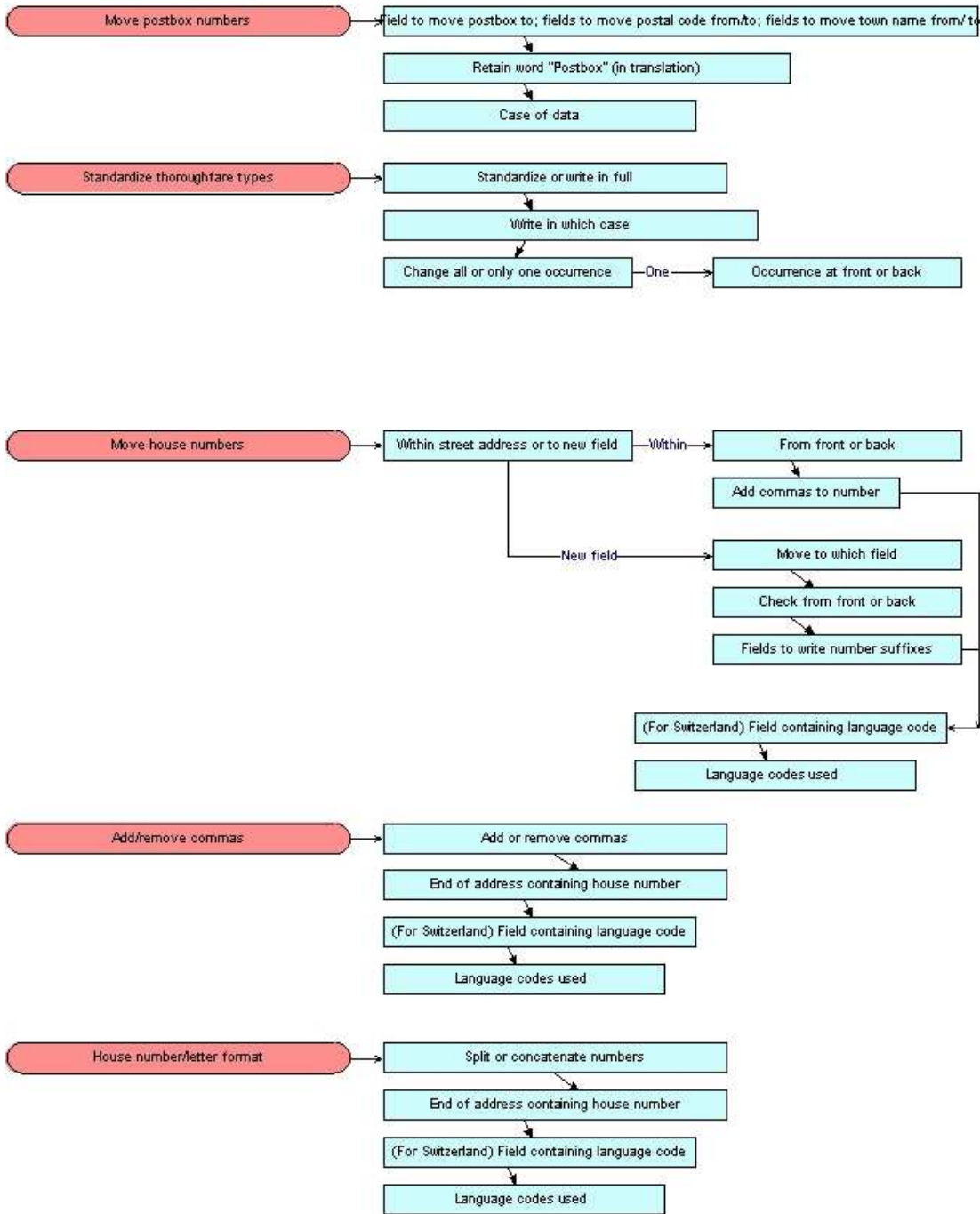
**Procedure**

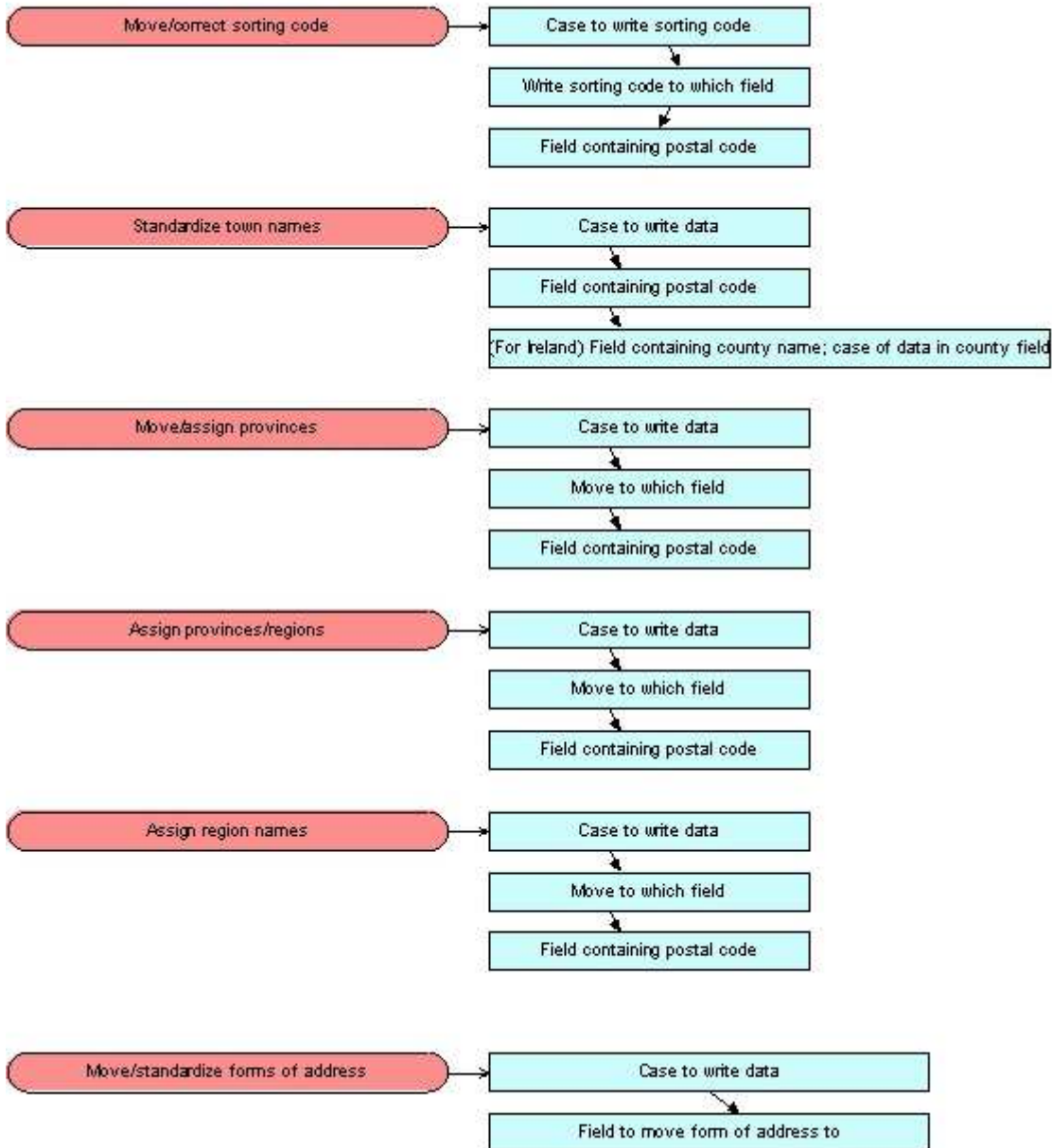
**Parameters**



**Procedure**

**Parameters**



**Procedure****Parameters**

### 10.1. Invalid strings - move field contents

## Invalid strings - move field contents

**Purpose:** to move the contents of fields containing only invalid strings, such as punctuation, "asdasdads", obscenities and so on

**Information required:** For each field chosen, specify the field to which the string will be parsed.



To be parsed an invalid string must be on its own in a field.

**10.2. Field - split string****Field - split string**

**Purpose:** to split a field string at a given sub-string.

This is a general process that can be run on any character field. It splits a field at a specified sub-string. The data before the given string stays in the specified field, the data after is moved to a new field.

**Example:** **Station House High Street**, split after **House** produces **Station House** and **High Street**

**Information required:** For each chosen field, specify the field to which the split data is to be moved.

Indicate the sub-string at which the field is to be split. Note: To indicate a leading or trailing space in this sub-string, use the "\$\_ \$" wildcard. Thus,

house

will split at all occurrences, including, for example *Porterhouse*.

\$\_house

indicates a leading space, so *Porterhouse* will not be split but *houseman* will.

\$\_house\$\_

indicates a leading and trailing space, so neither *Porterhouse* nor *houseman* will be split.

You do not need to use this wildcard for spaces within the string.

The casing of the string is taken into account by the procedure.

Specify whether the split is to be made before or after the given sub-string. If the sub-string is *\$\_house\$\_*, then *Front* will split **Station House High Street** into **Station** and **House High Street**. *Back* will split **Station House High Street** into **Station House** and **High Street**.

You are also required to define whether the split string is to be appended to the front of any existing data in destination field; to the back of any existing data in the destination field; to overwrite any data in the destination field; or to do nothing if the destination field already contains data.



If you specify multiple fields to split, you are asked for these parameters for each field. However, apart from the field name to which to write the split data, all other parameters are fixed. Thus, if you choose a sub-string to split for the first field, and a different one for the second field, the second sub-string is used for all fields as it overwrites the first sub-string. If you wish to use different sub-strings or different parameters for this process for different fields, run the process multiple times.

**Split string****Split string**

Split string from field SPLITTEST1 to which new field?:

SPLIT1

String to use as split (use \$\_\$ for leading/trailing spaces)?

\$\_\$House\$\_\$

Split at front or back of this string?:

 Front Back

Write to new string in which manner?:

 Append to front Append to back Overwrite Only if empty

New string Existing string

Existing string New string

New string

Existing string



**10.3. Field - Trim leading spaces****Field - trim leading spaces**

***Purpose:*** remove spaces at the front of a string within a field

This is a general process that can be run on any character field. It removes leading spaces from the string within a field.

***Example:*** “ **Station House, High Street**” becomes “**Station House, High Street**”

## 10.4. Field - trim final string

# Field - trim final string

**Purpose:** to remove a user-defined character or string at the final position(s) of the string within a field, if it occurs there.

This is a general process that can be run on any character field. It removes a user-defined character or string at the final position(s) of the string within a field, if it occurs there.

**Example:** **Station House High Street /** , remove **/** , produces **Station House High Street .**

**Information required:** For each chosen field, specify the character or string to be removed.



If you specify multiple fields to process, you are asked for this parameter for each field. However, the parameter is fixed within the program. Only the final choice is taken. Thus, if you choose a string to remove from the first field, and a different one for the second field, the second string is used for all fields as it overwrites the first string. If you wish to use different strings for this process for different fields, run the process multiple times.

Split string

Remove final string


String to remove from end of string:

## 10.5. Non-numeric characters - remove

# Non-numeric characters - remove

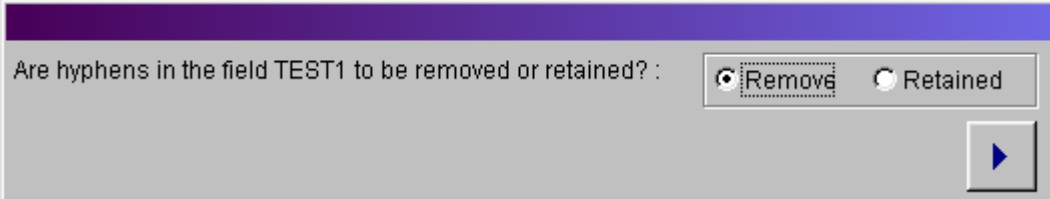
**Purpose:** to remove non-numeric characters from the user-specified fields.

This is a general process that can be run on any character field intended only to hold numeric data. It should be the first process to be run over address data for those countries where the postal code consists only of numbers, as the postal code is required for **GRC Tools™** in its correct format (without addition code or punctuation) for many other processes.

 This process also removes spaces. As spacing may be needed in the postal codes of certain countries (e.g. 999 99 in Sweden) this spacing should be added after this process has been run and before running other **GRC Tools™** processes that require the postal code.


**Example:** 23.45.67.89 becomes 23456789

**Information required:** hyphens (-) are often found in telephone numbers in certain countries like Germany, where they indicate a direct-dialled internal extension number. In other countries it may indicate a number of telephone lines - **33-37** for example, may indicate that lines ending in 33, 34, 35, 35 and 37 belong to this company. **GRC Tools™** gives you the choice of removing or retaining hyphens for each field chosen.



Are hyphens in the field TEST1 to be removed or retained? :

Remove  Retained



**10.6. Punctuation - remove**

## **Punctuation - remove**

***Purpose:*** to remove punctuation from the user-specified fields.

This is a general process that can be run on any character field. It removes punctuation from the field.

***Example:*** **I.C.B. Watford - Bakers Ltd.** becomes **I C B Watford Bakers Ltd**

***Information required:*** none

**10.7. Double spaces - remove**

## **Double spaces - remove**

***Purpose:*** to remove double spaces from the user-specified fields.

This is a general process that can be run on any character field. It removes double spaces from the field.

***Example:*** **John W. Smith** becomes **John W. Smith**

***Information required:*** none

## 10.8. Telephone numbers - burst

# Telephone numbers - burst

**Purpose:** to burst a telephone number into its component parts: international dialling code, trunk dialling code (the digit which may be dropped when calling a telephone number from abroad), area code, subscriber number and extension information. The original data is not altered in this process. The new numbers are written without punctuation. Telephone numbers from countries which are incorrectly coded cannot be burst.

**Example:** 0044(20)625-526 becomes +44 and 0 and 20 and 625526

**Information required:** For each field containing a telephone number to be burst, you need to specify the names of the existing (alpha-numeric) fields to which to write the number components. ALL fields except that to contain a trunk code or an extension must be assigned in order to successfully burst a telephone number.

Telephone numbers - burst

Telephone numbers - burst

Field: TELEPHONE

Write international code to which field: INTL

Include +

Assign?

Write trunk code to which field: TRUNK

Assign?

Write area code to which field: AREA

Write subscriber's number to which field: PHONE

Write extension to which field: EXT

You may choose to assign an international code and/or a trunk code if it does not exist within the number to be burst; and you may choose to include the '+' symbol in front of the international dialling code.

**10.9. Postal codes / Postal codes (incl. format) - locate**

## Postal codes - locate

### Postal codes (incl. format) - locate



Only run this process and *Move Located Postal codes* in the same session of **GRC Tools™** if you are confident that **GRC Tools™** has made no errors in identifying postal codes strings within your data file. Checking the intermediate file first is always advisable.

**Purpose:** to search for postal codes in user-defined fields. The process *Locate postal codes* searches for postal codes that are correctly formatted, whilst *Locate postal codes (incl. format)* searches also for postal codes that may be incorrectly formatted (where a postal code would normally contain a space or a hyphen). Thus, the former process for The United Kingdom will locate the postal code *TW1 IAA*, whilst the latter will locate *TW1 IAA and TW1IAA*.



Exporting from some formats (e.g. Excel) may cause data to be centred within fields. As these processes check data from the left of the field, all fields are left-aligned before processing. Credits are charged for left-aligning the data even if no postal code is found, as **GRCTools™** registers that the field after processing is different from the field before processing. To prevent this, assure that data is left-aligned in fields before processing.

Where a postal code is preceded by a postal country code (e.g. **CH-1726**) then the postal country code is also parsed with the postal code. This country code can be stripped from the postal code using the process *Remove postal code country code*. Postal codes followed by certain punctuation marks are also recognised and parsed.

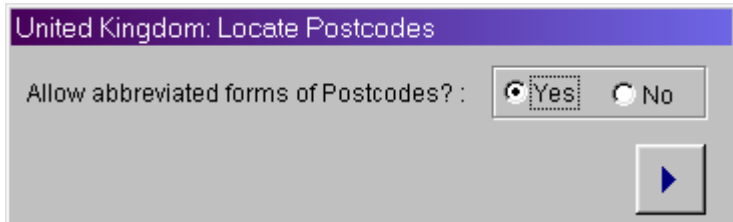
**Information required:** You are required to specify the field in your database containing its unique identification code, or check the *Check to use RECORD NUMBERS instead of ID* box to store the record number to the file created instead of an ID.

Belgium: Locate Postcodes

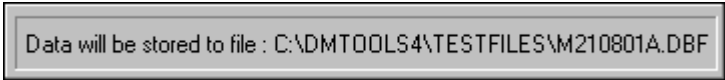
Choose field containing UNIQUE ID CODE:

Check to use RECORD NUMBERS instead of ID:

For the **United Kingdom** only, you have the option of allowing abbreviated forms of postal codes. Especially in London, addresses are often given with only the outward (first section) of the postal code, e.g. **SW1** instead of **SW1 2AB**. Though these postal codes are technically incomplete, this option allows you to search also for these abbreviated postal codes.



A file called *MddmmyyX.DBF*, where *ddmmyy* is the date and *X* is an incremented letter is created. **GRC Tools™** will tell you at the top right of the screen the name of the file created for this session.



! If you should do more than 26 runs in a single day and have thus created files ending in all letters between A and Z, A is reused for all remaining sessions for that date.

! The *MddmmyyX.DBF* file is created in Visual Foxpro format. To view this file you will need Visual Foxpro or a program which can import Visual Foxpro files.

The *MddmmyyX* file created will contain the unique identification code of the record (or the record number if you so choose by checking the “*Check to use Record Numbers instead of ID*” box), the country code, the name of the field which was checked, the original field contents, the postal code (if **GRC Tools™** has identified one), a message and a position. For each record in the user-defined database a corresponding record is created in the *MddmmyyX* file containing one of four messages:

- Postal code identified
- No Postal code Found
- Search Field is Empty
- Lookup Table not Found

<i>id</i>	<i>coun</i>	<i>field_name</i>	<i>field_cont</i>	<i>pc</i>	<i>message</i>	<i>posn</i>
1	FRA	TOWN	75000 PARIS	75000	Postal code Identified	1
2	FRA	TOWN	PARIS		No Postal code Found	
3	FRA	TOWN			Search Field is Empty	
...	...	...	...	...	...	...
17	FRA	TOWN	03001 RENNES	03001	Postal code Identified	1
...	...	...	...	...	...	...
23	AFG	TOWN	KABUL		Lookup Table Not Found	...
56	UK	COUNTY	OXON, OX9 6RY	OX9 6RY	Postal code Identified	7
93	ICE	TOWN	235 Reykjavik	235	Postal code Identified	1
...	...	...	...	...	...	...

If **GRC Tools™** thinks it has found a postal code, it will fill the *pc* field and the *posn* field. If you would like **GRC Tools™** to assign the postal codes found to a new field using the *Move located postal codes* option, you need to check this file and to delete those records where a postal code has been incorrectly identified.

! If upon checking the *MddmmyyX* file you locate a postal code not found by GRC Tools™, you need only add the postal code to the *pc* field and the position of the start of

the postal code to the *posn* field. Do not change the *field\_cont* data or add a *message*.

**10.10. Postal codes - write located****Postal codes - write located**

The process *Locate postal codes* or *Locate postal codes (incl. format)* must have been run in a previous **GRC Tools™** session to be able to run this process

**Purpose:** to move postal codes located using the *MddmmyyX* file created by the **GRC Tools™** process *Locate postal codes* or *Locate postal codes (incl. format)*

**Information required:** You are required to specify the field in your database containing its unique identification code, or check the *Check to use RECORD NUMBERS instead of ID* box if record numbers were used in the *Locate postal codes* process. You must specify to which field the located postal code is to be used, and which *MddmmyyX* file is to be used.

Note: if you chose to check using record numbers instead of an ID, then you must not have changed the order of the records within the original file before running this process, otherwise the corrected data will be written back to the wrong record.



The drop-down file box contains all databases in the data file directory which begin with an 'M'. Make sure you only choose a file with the pattern any other file will cause the process to fail.

*MddmmyyX* - choosing



If the located postal code ends in a comma or a full stop, this character is not moved to the new field by this process. It is retained in the old field, unless this would mean that it would come into the first position of the field, in which case it is removed. Thus **Oxford OX9 6RY, Oxon** becomes **Oxford, Oxon** and **OX9 6RY** whilst **OX9 6RY, Oxon** becomes **Oxon** and **OX9 6RY**.

**10.11. Postal code country code - remove**

## **Postal code country code - remove**

***Purpose:*** to remove the postal country sorting code which sometimes precedes a postal code.

This process can be run on characters fields containing postal code data.

***Example:*** **GB-TW1 1AA** becomes **TW1 1AA**


**B-1000** becomes **1000**

**CH1017** becomes **1017**

***Information required:*** none

**10.12. Thoroughfare types - parse****Thoroughfare types - parse**

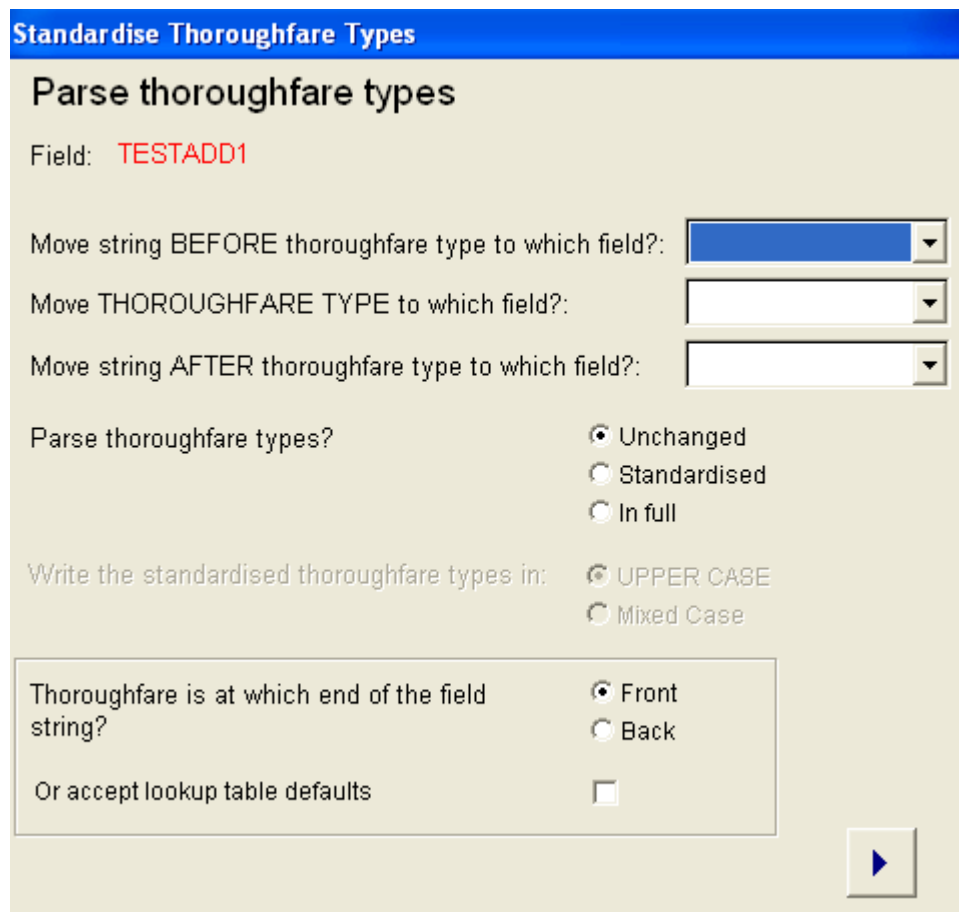
**Purpose:** to parse strings indicating thoroughfare types and other address type strings (“Street”, “strasse”, “rue”, “house”, “apartment”, “zone” etc.) to user-defined fields.

 The lookup tables used by **GRC Tools™** for this process are very large. **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

**Example:** **R. de Paris** becomes **rue|de Paris**; **Kölnstr.** becomes **Köln|straße** etc.

**Information required:** For each field chosen, **GRC Tools™** needs to know to which fields to move the data. Three fields are required: one to which to move the thoroughfare string, one to which to move the string BEFORE the thoroughfare string, and one to which to move the string AFTER the thoroughfare string.

When the thoroughfare string is parsed, you can choose to leave it unchanged, to write it in its standardised form, or in its fullest form. For example, if the string being processed is *HIGH STR*, then its unchanged form is *STR*, its standardised form is *ST* and its full form is *STREET*.



**Standardise Thoroughfare Types**

**Parse thoroughfare types**

Field: **TESTADD1**

Move string BEFORE thoroughfare type to which field?:

Move THOROUGHFARE TYPE to which field?:


Move string AFTER thoroughfare type to which field?:

Parse thoroughfare types?  Unchanged  Standardised  In full

Write the standardised thoroughfare types in:  UPPER CASE  Mixed Case


Thoroughfare is at which end of the field string?  Front  Back

Or accept lookup table defaults



If you choose to standardise the thoroughfare types or write them in their fullest form, then you may choose the case in which to write the data.

Finally, you need to specify at which end of the string **GRC Tools™** should start looking for the thoroughfare type. Unless your data is formatted in an unusual way, for example *de Paris*, *rue*, you should choose to accept the lookup table defaults.

 **Known issue:** due to the nature of the lookup tables used, for some thoroughfare types where the final character is a punctuation mark, such as *AV.*, **GRC Tools™** may take the punctuation mark as being part of the following string, thus parsing to: *AVENUE/*.

**10.13. Postal code - format****Postal code - format**

***Purpose:*** to add spaces or hyphens to postal codes which require them, or to remove spaces and hyphens where not required. |

This process can be run on characters fields containing postal code data.

***Example:*** 10100 becomes 10 100  
TW11AA becomes TW1 1AA  
100 205 becomes 100205

***Information required:*** none

**10.14. Postal code - assess validity****Postal code - assess validity**

**Purpose:** to check the validity of postal codes by analysing them for correct length, correct character type, the presence of invalid characters and the presence of foreign postal codes.

! This process requires the postal code to be written in the user-specified field in the correct format and without the addition of country codes such as **F-75000** or **GB-W1A 4ZA**. I.e. these postal codes need to be written **75000** and **W1A 4ZA**.

A file called *PddmmyyX.DBF*, where *ddmmyy* is the date and *X* is an incremented letter is created to store any records containing problem postal codes. **GRC Tools™** will tell you at the top right of the screen the name of the file created for this session:

Data will be stored to file : C:\DMTOOLS4\TESTFILES\200801A.DBF

! If you should do more than 26 runs in a single day and have thus created files ending in all letters between A and Z, A is re-used for all remaining sessions for that date.

! The *PddmmyyX.DBF* file is created in Visual Foxpro format. To view this file you will need Visual Foxpro or a program which can import Visual Foxpro files.

United Kingdom: Check Postcode Validity

Choose field containing UNIQUE ID CODE:

Check to use RECORD NUMBERS instead of ID:

You are required to specify the field in your database containing its unique identification code, or check the *Check to use RECORD NUMBERS instead of ID* box to store the record number to the file created instead of an ID. You also need to specify the names of the fields containing the company name, street address (the second line is optional) and the settlement name.

United Kingdom: Check Postcode Validity

Field containing COMPANY NAME code :

Field containing STREET ADDRESS (1) :

Field containing STREET ADDRESS (2) :

Field containing TOWN NAME :

For the **United Kingdom** only, you have the option of allowing abbreviated forms of postal codes. In London particularly, addresses are often given with only the outward (first section) of the postal code, e.g. **SW1** instead of **SW1 2AB**. Though these postal codes are technically incomplete, this option allows you not to mark these abbreviated postal codes as incorrect.

Some countries embrace other, smaller, countries within their postal systems (such as Switzerland with Liechtenstein, Italy with San Marino and Holy See, etc.), and the addresses from the smaller countries are often mixed up in databases of addresses from the bigger country. This process can identify these addresses and assign a new country code to them. If you choose to do so, you must specify the field which contains the country code, and the codes that you wish to use for these countries. In some cases, the countries listed are not officially in a different country. For example, the French Overseas Territories such as Guadeloupe and Martinique are constitutionally part of France; and the crown dependencies of the Channel Islands and the Isle of Man are not entirely independent, though they are not a part of The United Kingdom. They have, however, been included for completeness, as many users want to code these areas differently. If this is not the case, simply assign the same code for, for example, the Guadeloupe, as you have assigned for - in this case - France itself.

The *PddmmyX* file created will contain the unique identification code of the record (or the record number if you so choose by checking the *Check to use RECORD NUMBERS instead of ID* box), the name of the field which was checked, the incorrect postal code, the country in which it belongs, a description of the problem, and, because this information can often give a clue to the correct postal code, the company name, address and settlement names. For example, for France:

id	field_name	postal code	country	message	corr_pc	corr_coun	company	address	address_2	city
19	testfile.pc	790231	France	Postal code must be 5 digits long			...	...	...	...
38	testfile.pc	99001	France	String 99 not allowed in postal code position 1						
47	testfile.pc	1043	France	Postal code must be 5 digits long						

---

47	testfile.pc	1043	France	Digit 5 in postal code must be NUMERIC						
...	...	...	...	...						

This file can then be used to correct postal codes in the original file by entering the corrected postal code and/or country code into the fields *corr\_pc* and *corr\_coun* and running the process ***Assign corrected postal codes***. Please refer to ***Assign corrected postal codes*** for details

**10.15. Postal codes - write corrected**

## Postal codes - write corrected

**Purpose:** to assign to the user-defined data file any corrected postal codes that the user has written to the file *PddmmyyX.dbf*, created when running *Check postal code validity*. This process should not be run unless *Check postal code validity* has already been run on the same file, and corrections made to the *PddmmyyX.dbf* file

The process *Assess postal code validity* creates a database containing postal codes which are considered by **GRC Tools™** to be, in some way, invalid. This file contains 2 empty fields: *corr\_pc* and *corr\_coun*. In *corr\_pc* you can add a corrected value for the postal code, and in *corr\_coun* a corrected value for the country code, in this way:

id	field_name	postal code	country	message	corr_pc	corr_coun	company	address	address_2	city
19	testfile.pc	790231	France	Postal code must be 5 digits long	79023		...	...	...	...
38	testfile.pc	99001	France	String 99 not allowed in postal code position 1	97001					
47	testfile.pc	1043	France	Postal code must be 5 digits long		Belgium				
47	testfile.pc	1043	France	Digit 5 in postal code must be NUMERIC		Belgium				
...	....	...	...	...						

If this is done, this process can be run to write the corrected data back to the data file.

**Information required:** You are required to specify the field in your database containing its unique identification code, or check the *Check to use RECORD NUMBERS instead of ID* box if this was the method used to create this correction file. You need to specify the name of the *PddmmyyX.dbf* file which contained the corrections, and also the names of the fields to which to write the corrected postal code and the corrected country code. Note: if you chose to check using record numbers instead of an ID, then you must not have changed the order of the records within the original file before running this process, otherwise the corrected data will be written back to the wrong record.

Assign corrected postal codes

Choose field containing UNIQUE ID CODE:

Check to use RECORD NUMBERS instead of ID:

Choose file from which to assign postal codes :

Field to which to assign corrected postal code :

Field to which to corrected country code :

## 10.16. Language Regions - assign

# Language Regions - assign

**Purpose:** To assign a language to each address within the database file specified in multi-lingual countries.

Although it is often sufficient to know the country in which an address is located in order for **GRC Tools™** to run its processes, in certain countries where more than one language is spoken, there is a clash between certain aspects of addressing between each language, and therefore **GRC Tools™** needs to work on the basis of language areas. This process assigns language areas. Please note that this language area assignment neither relates to the language spoken by the person/people living or working at the address, nor the language that you might want to send information in. It specifies either the predominant language of the area where the address is situated, or, for bilingual areas or for addresses without a postal code, the language in which the address is written.

**Information required:** For each country, this process needs to know:

- the field to which the language code is to be written.
- the fields containing the street address (maximum two - one may be left empty).
- the field containing the postal code
- the codes which need to be written to the language field for each language area.

The process also gives you the choice of checking every address, even if a language code is already assigned, and assigning a new code (leave the “overwrite” check box unchecked); or, by checking the “overwrite” check box, to ignore those addresses where a language has already been assigned, and checking only those where no language is yet assigned.

Belgium: Assign Language Regions

Field to which to assign Language Region code : LANGUAGE

Field containing Street Address (1) : ADDRESS1

Field containing Street Address (2) : ADDRESS2

Field containing Post Code : ZIPCODE

Overwrite Language Region ONLY if EMPTY :

Code indicating (majority) DUTCH speaking : nld

Code indicating (majority) FRENCH speaking : fra

Code indicating (majority) GERMAN speaking : deu

**10.17. Quotation marks - remove**

## Quotation marks - remove

**Purpose:** to remove paired quotation marks from the user-specified field.

**Example:** “Dun Roamin” becomes **Dun Roamin**

This process searches for, and removes, double quotation marks.



Single quotation marks, found in addresses as apostrophes, are not removed.

If more than two double quotation marks are located within the same field, the process will not make changes to the field but will write the data to a file. A file is created called *QddmmyyX.dbf*, where *ddmmyy* is the date and *X* is an incremented letter is created. **GRC Tools™** will tell you at the top right of the screen the name of the file created for this session.

Data will be stored to file : C:\DMTOOLS4\TESTFILES\Q220801B.DBF



If you should do more than 26 runs in a single day and have thus created files ending in all letters between A and Z, A is reused for all remaining sessions for that date.



The *QddmmyyX.DBF* file is created in Visual Foxpro format. To view this file you will need Visual Foxpro or a program which can import Visual Foxpro files.

You are further required to give either the name of the field within the database table which contains a unique ID (which is written to the *QddmmyyX* file so that the user can re-locate the problem record); or, by checking the *use Record Numbers instead of ID* box, to write the record number of the problem record instead of a unique string. The file *QddmmyyX* will contain in a single field called **ID** containing the unique ID or record number of any record where the field specified has an uneven number of double-quotation marks.

**10.18. Apostrophes (French) - add**

## Apostrophes (French) - add

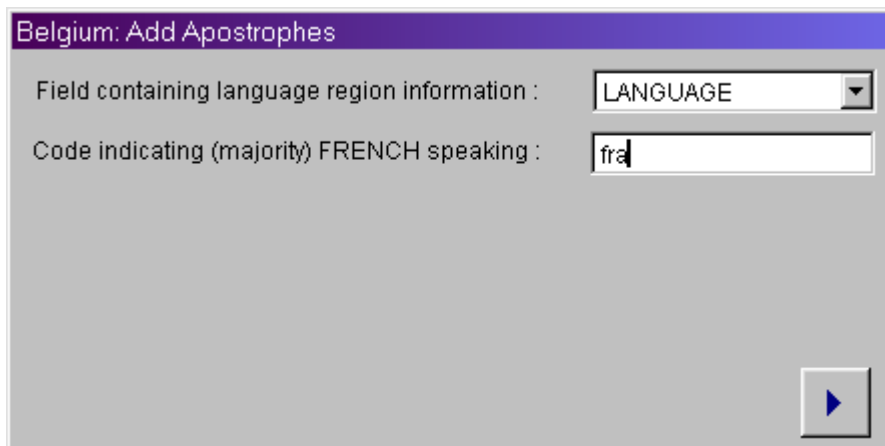
**Purpose:** To add omitted apostrophes into data strings.

This process searches for stand-alone occurrences of the letters 'L' and 'D' (in both upper- and mixed-cases) and, if followed by a vowel, 'H' or 'Y', an apostrophe is added.

**Example:** L HUYSMAN D AMIENS becomes L'HUYSMAN D'AMIENS

! Although this process cannot differentiate between an aspirated and unaspirated aitch ('H'), the aitches found preceded by a stand-alone occurrence of a 'D' or 'L' are assumed to be unaspirated and therefore requiring the apostrophe. Note also that, except where occurring in the first 2 places of the string, the 'L' or 'D' preceded by another standalone character are assumed to be part of an abbreviation and are not given apostrophes. Thus S A R L AMIENS will not become S A R L'AMIENS.

**Information required:** For countries where more than one language is spoken, and where this process is allowed, the field containing the language region information (assigned using *Assign language region* process or already existing in the database table) and the codes for the languages affected are required.



Belgium: Add Apostrophes

Field containing language region information : LANGUAGE

Code indicating (majority) FRENCH speaking : fra

**10.19. Casing - to upper case**

## Casing - to upper case

**Purpose:** to correctly translate the contents of the user-defined field from mixed- or lower-case to upper case.

*This process costs one credit per altered field.*



This process checks also each accented character in order to assign a correct upper-case equivalent, depending on the country or language concerned.

**Example:** **Rue du Récif** in France becomes **RUE DU RECIF**  
**Kölnstraße** in Germany becomes **KÖLNSTRASSE**

**Information needed:** There is a clash in the casing rules for Belgium between Dutch/German and French; and in Switzerland between German, French and Italian. For this reason, this process handles each language area differently. It is therefore necessary to have already written the language areas to the database table using the procedure *Assign language region* (unless the language region is already coded). When running this process for Belgium or Switzerland, you are required to specify which field contains the language region and which codes are used for each language.

Belgium: To Upper Case

Field containing language region information : LANGUAGE

Code indicating (majority) GERMAN speaking : deu

▶

**10.20. Casing - to mixed case / to mixed case - addresses / to mixed case - other**

## Casing - to mixed case

### Casing - to mixed case - addresses

### Casing - to mixed case - other

**Purpose:** to correctly translate the contents of the user-defined field(s) from upper case to mixed (upper and lower) case.

The process *To mixed case* can be used for all fields. However, certain countries require two processes. These countries are those where there are *single character* prepositions within the language. For example, in French, a freestanding *A* could be an abbreviation but might also be the preposition *A* meaning *TO*. As the computer cannot distinguish these differences in usage, the process *To mixed case - addresses* is used for fields where abbreviations are unlikely to occur, such as the street address, and this assumes that free-standing characters such as *A* are prepositions and puts them into lower case. The process *To mixed case - other* assumes that free-standing characters such as *A* are abbreviations and leaves them in upper case.

Running this process over street addresses for countries where a single letter indicates a proposition, such as **a** in France, will result in letters which are part of the house number being put into lower case. Thus in France, **21 A** would become **21 a** whilst **21 B** would remain unchanged. If this is undesirable, run the process *To mixed case - other* rather than *To mixed case - addresses* over the street address.

In countries where single letter prepositions do not exist, the process is called *To Mixed Case*. The processes *To Mixed Case* and *To Mixed Case - Addresses* are the same. Therefore, when choosing the option *Apply parameters to all countries*, choosing to run only *To Mixed Case - Other* on a field will leave fields unprocessed in countries with the default *To Mixed Case* process. You should therefore choose to run both *To Mixed Case - Addresses* and *To Mixed Case - Other* in this case to ensure that all fields are processed.

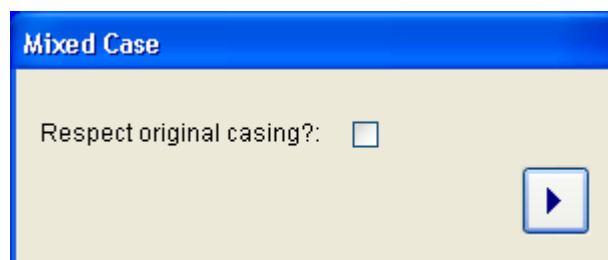
#### **Information required:**

Abbreviations which are written with neither spaces nor commas in the field(s) defined cannot be distinguished by the process from other words, and will also be made into mixed case, i.e.

*BBC Transmission Of Programmes Plc* will become  
*Bbc Transmission of Programmes PLC*

This does not happen if abbreviations are consistently written with full stops or spaces, i.e. *B.B.C.* or *B B C*.

If the data in the field to be processed is already in mixed case, and you wish only to correct the casing, you can choose to check the *Respect original casing?* in this dialogue box:



Checking this option prevents the process from lower-casing the original data during processing, so

*BBC Transmission Of Programmes Plc* will become  
*BBC Transmission of Programmes PLC*

This option can only be used usefully if the data being processed is already in mixed case.

There is a clash in the casing rules for Belgium between Dutch/German and French; in Canada between French and English; and in Switzerland between German, French and Italian. For this reason, this process handles each language area differently. It is therefore necessary to have already written the language areas to the database file using the procedure *Assign language region* (or that the language areas already coded). When running a mixed-case process for Belgium, Canada or Switzerland, you are required to specify which field contains the language region and which codes are used for each language.

Belgium: Mixed Case

Field containing language region information : LANGUAGE

Code indicating (majority) DUTCH speaking : nld

Code indicating (majority) FRENCH speaking : fra

Code indicating (majority) GERMAN speaking : deu

▶

**10.21. Articles - move or parse****Articles - move or parse**

**Purpose:** to move articles (“the”, “a”, “an”) from one end of a string to the other, whilst standardising string format.

**Example:** Baker’s Dozen, The or Baker’s Dozen (The) become The Baker’s Dozen or The Baker’s Dozen becomes Baker’s Dozen, The or Baker’s Dozen (The)



- For moving of articles from the *FRONT* to the *BACK* of a string, this process is unable to distinguish between single-lettered articles (such as “a” in English) and abbreviations/prepositions/conjunctions. These occurrences are therefore not changed by this process. For example, **A Baker’s Dozen** remains unchanged. However, these articles can be moved from the *BACK* to the *FRONT* of the address provided they are preceded by a comma or are in brackets. Thus **Baker’s Dozen (A)** will become **A Baker’s Dozen**.
- This process will only move the article from *BACK* to *FRONT* if it is the last string found on the line (to prevent articles which are part of sub-clauses within a field, such as **Ground Floor, The Smithsonian Building** from being affected).
- When an article is found by the process and a move has been made, the process looks no further - it is assumed that only one article per field will be found.
- Articles moved are put into a standard format, as also are those already in place. I.e., if you are moving the article to the back, with a comma, those articles at the back between brackets will be altered to be at the back with a comma.

**Information required:** For each field chosen, you must specify in which case the field data is (upper or mixed (upper and lower)); and whether the article is to be moved from the front to the back of the string or vice versa.

Is the data in the field TESTCOMP in Upper or Mixed case? :  Upper  Mixed

Move the article in TESTCOMP to which end of the string? :  Front  Back

Next

If you choose to move the article to the *BACK* of the string, you will be required to define whether you require the article to be placed after a comma at the end of the string or within brackets.

Specify format of article in TESTCOMP after move :  With Comma - , The  With Brackets - (The)

Next

**10.22. "And" strings - standardise****"And" strings - standardise**

**Purpose:** To standardize the different forms of the word AND (and symbol equivalents) in the local language.

**Example:** C. and A., C. + A. etc. become C. & A.



Care must be taken when using this process for languages where the word “and” consists of a single character, such as Italian (“e”) and Spanish (“y”). Whereas in most countries the lookup tables have been set to standardize the word “and” into the ampersand symbol (“&”), this is not the case for these countries as it is not possible to distinguish between the different uses of this single letter (as the word “and”, as an abbreviation, etc.). For these countries therefore, the ampersand is standardised into the letter rather than the other way around.

**Information required:** For each field chosen, **GRC Tools™** will ask whether this field is in upper case or mixed (upper and lower) case so that it can replace each occurrence of the word AND with its equivalent in the correct case.

Is the data in the field TESTCOMP in Upper or Mixed case? :

Upper  Mixed

▶

For Switzerland there is a clash between Italian- and French/German-speaking areas. For Switzerland, therefore it is necessary to have written the language areas to the database file using the procedure **Assign Language Region** (or have the language areas already coded). You are required to specify which field contains the language region and which code is used for Italian in Switzerland.

Switzerland: Standardize AND strings

Field containing language region information : LANGUAGE

Code indicating ITALIAN speaking : ita

▶

**10.23. Abbreviations / Abbreviations in personal names - standardise**

## Abbreviations - standardise

### Abbreviations in personal names - standardise

**Purpose:** to standardize the format of abbreviations and acronyms so that they either all have full stops or all have spaces.

**Example:** A B C D becomes A.B.C.D.; A.B. C. becomes A.B.C.  
or A.B.C.D. becomes A B C D; A B. C becomes A B C



This process cannot recognize abbreviations which have been entered into the database without either spaces or full stops, such as **BBC**. As certain countries have prepositions or conjunctions ('and') which themselves consist of a single character, standalone characters are not given a full stop with the process *Standardize abbreviations*. Thus **SILVIO Y DONNA SERVICIOS DE PUBLICIDAD SA** remains unchanged. However, the process *Standardize abbreviations- names* does give a full stop to standalone characters, and should be run over fields containing personal names, so that initials can be correctly formatted. With this process, **Mr T Smith** would become **Mr T. Smith**.

**Information required:** For each country chosen, you are required to specify whether you want to give all abbreviations full stops, or to replace the existing full stops with spaces. You are assumed to require the same format for all fields. If this is not the case, run this process twice over the data: once for those fields requiring full stops and once for those fields requiring spaces.

United Kingdom: Standardize Abbreviations

Standardize with Full Stops or Spaces?:

To Full Stops (A B C -> A.B.C.)

To Spaces (A.B.C. -> A B C)

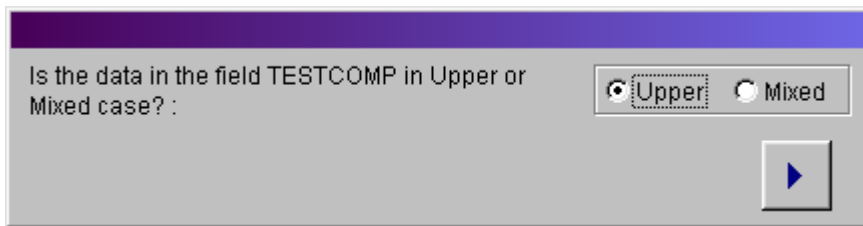
## 10.24. Company types - standardise

# Company types - standardise

**Purpose:** to standardize company type indications (Ltd, PLC, SA etc.) and other commonly found words and abbreviations in company names.

**Example:** *Société de la Cité S.A.* becomes **Ste. de la Cité SA**; **Burnleys Bank Limited** becomes **Burnleys Bank Ltd**

**Information required:** For each field chosen for this process, you must define whether the data contained within it is in upper or mixed (upper and lower) case.



Is the data in the field TESTCOMP in Upper or Mixed case? :

Upper  Mixed

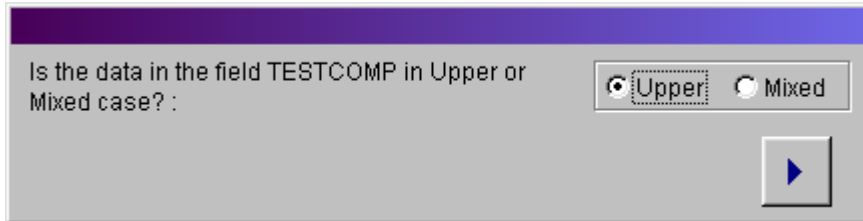
**10.25. Department strings - standardise**

## Department strings - standardise

**Purpose:** to standardize department strings.

**Example:** **Libr. Acquisitions Dept.** becomes **Library Acquisitions Department**; **Afd. Inkoop** becomes **Afdeling Inkoop**

**Information required:** For each field chosen for this process, you must define whether the data contained within it is in upper or mixed (upper and lower) case.



Is the data in the field TESTCOMP in Upper or Mixed case? :

Upper  Mixed

## 10.26. Sub-building information - parse & standardise

# Sub-building information - parse & standardise

**Purpose:** to identify sub-building data associated with numbers within street address fields and to move this information to a new field.

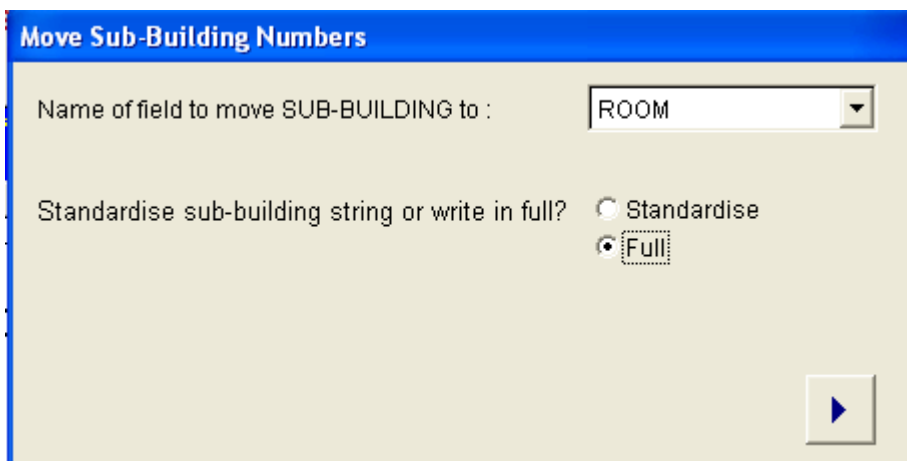
This process works by identifying strings in the user-defined address field which indicate a sub-building. The number(s) following (or preceding) this string are written to a user-defined field (the process stops taking data from the field once anything other than a number or space is found). See example below. Once a string is found, the process stops checking - each address is assumed to have only one sub-building string.

**Example:** **Rm 12 A, Station House** becomes: **Room 12 A** and **Station House**; **11th Floor Station House** becomes **11th Floor** and **Station House**

This process currently does not parse sub-building data without an associated number (e.g. *The Penthouse*).

**Information required:** You need to specify once for each **GRC Tools™** session these parameters:

- the field to which to move the sub-building information
- whether to write the sub-building string in the new field in full or in its standardised form.

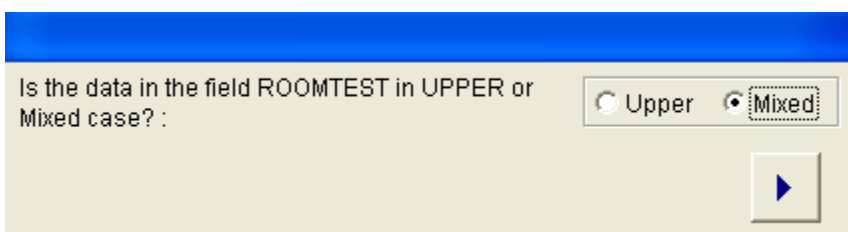


**Move Sub-Building Numbers**

Name of field to move SUB-BUILDING to :

Standardise sub-building string or write in full?  Standardise  Full

For each field specified, you must specify whether the data in the field is in upper or mixed (upper and lower) case. The same case is retained in the field to which the sub-building information is written.



Is the data in the field ROOMTEST in UPPER or Mixed case? :

Upper  Mixed

**10.27. Post office box numbers - parse****Post office box numbers - parse**

**Purpose:** to identify post office box numbers within street address fields and to move these numbers to a new field, with or without the correct, standardised local-language word for “post office box”.

This process works by identifying strings in the user-defined address field which indicate a post office box number. The number(s) following this string are written to a user-defined field (the process stops taking data from the field once anything other than a number or space is found). If required, the correct local-language form of the word “post office box” is also written to this field. Postal code and postal town information may also be copied to new fields as post office boxes in some countries may have a separate postal code and postal town to the street address. The remaining street address data is stripped of any remaining preceding punctuation (.,-,: or /). See example below. Once a string is found, the process stops checking - each address is assumed to have only one post office box number.

**Example:** **Postbox 17, Ooststraat 21** in the Netherlands will be split into two fields: **Postbus 17** and **Ooststraat 21**.

Postboxes which commence with a single letter (e.g. *Locked Bag E142*) or those which end in one or two letters (e.g. *G.P.O. Box 128A* or *P.O. Box 1DT*) are also recognized and parsed. Apparent postbox strings which are not followed by a number are not parsed.

**Information required:** You need to specify once for each **GRC Tools™** session these parameters:

- the field to which to move the post office box number
- optionally, the field to take the postal code from and the field to which to copy it. Leave these fields empty if you do not want to copy the postal code.
- check the *Overwrite ONLY if EMPTY* box if you only want to copy the postal code if the field to move the postal code to does not already contain data
- optionally, the field to take the settlement name from and the field to which to copy it. Leave these fields empty if you do not want to copy the settlement name.
- check the second *Overwrite ONLY if EMPTY* box if you only want to copy the settlement name if the field to move the settlement name to does not already contain data
- Check the *Retain word POSTBOX (in correct translation)* box if you want to store the data preceded by the local-language form of the word “post office box”. If this remains unchecked, only the number is written to the post office box field, though both the post office box string and the number are removed from the address string. Thus, if this choice is checked, **Postbox 17, Ooststraat 27** in the Netherlands would become **Postbus 17** and **Ooststraat 27**; whereas unchecked it would become **17** and **Ooststraat 27**. Make sure that the post office box field is long enough to take the post office box string as well as the post office box number if you check this box.

**Move Postbox Numbers**

Name of field to move POSTBOX TO : TESTPB

Name of field to move POSTCODE FROM : PC

Name of field to move POSTCODE TO : MOVEDPC

Overwrite ONLY if EMPTY :

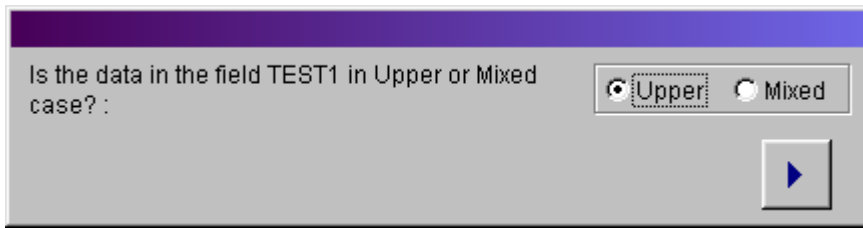
Name of field to move TOWN NAME FROM : POSTTOWN

Name of field to move TOWN NAME TO : MOVEDTOWN

Overwrite ONLY if EMPTY :

Retain word POSTBOX (in correct translation)?


For each field specified, you must specify whether the data in the field is in upper or mixed (upper and lower) case. The same case is retained in the field to which the post-box information is written.



A screenshot of a dialog box with a grey background and a blue header bar. The text inside the dialog box reads: "Is the data in the field TEST1 in Upper or Mixed case? :". To the right of the text is a group box containing two radio buttons: "Upper" (which is selected) and "Mixed". Below the radio buttons is a button with a blue right-pointing triangle.

**10.28. Thoroughfare strings - standardise****Thoroughfare strings - standardise**

**Purpose:** to standardize strings indicating thoroughfare types and other address type strings (“Street”, “strasse”, “rue”, “house”, “apartment”, “zone” etc.) within the user-defined field.

 The lookup tables used by **GRC Tools™** for this process are very large. **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

**Example:** **R. de Paris** becomes **rue de Paris**; **Kölnstr.** becomes **Kölnstraße** etc.

**Information required:** For each field chosen, **GRC Tools™** needs to know whether the data should be written in a *standardized* form or in *full*. By default, **GRC Tools™** writes the data in its long form. However, in some cases a single abbreviation can have more than one long form. For example, *ST* in English might be an abbreviation of *SAINT* or *STREET*; *PTE* in Dutch might be a normal word ending or the abbreviation of *PORTE*, and so on. In these cases, all occurrences are *standardized* to the short form. Thus:

*SAINT HELENS ST*  
*GEDEMPTE PORTE*

will become

*ST HELENS ST*  
*GEDEMPTE PTE*

Choosing the option *full* will retain the long form when it already exists, and leave the short forms untouched when found. Thus:

*SAINT HELENS ST*  
*GEDEMPTE PORTE*

will become

*SAINT HELENS ST*  
*GEDEMPTE PORTE*

*Standardize* will produce more standardized but less accurate results; *full* will produce less standardized but more accurate results.

You also need to specify whether the data each field contains is in upper-case or in mixed (upper and lower) case to ensure that the data is correctly replaced.

The option is also provided to change *all* thoroughfare types within the user-specified field, or *only one*. For example, if the field contains the Italian address **V V Emmanuel 12**, then changing *all* occurrences will change this to **via via Emmanuel 12**. Changing *only one* occurrence will change it correctly to **via V Emmanuel 12**. When *One Occurrence Only* is chosen, **GRC Tools™** will stop looking for thoroughfare types once the first is found, even when this thoroughfare type is already correctly written. Thus **via V Emmanuel 12** will remain unchanged - the **V** will not be altered. *All Occurrences* is useful when there is more than one thoroughfare type in a field, e.g. **31 Railroad Terr., 17 High St.** When *One Occurrence Only* is chosen, then you must specify which end of the address string to start searching for the thoroughfare type (or choose *Accept lookup table defaults* to use the correct searching end for that country). In Italy, for example, thoroughfare types are usually at the front of the address (**via V Emmanuel**), whilst they are at the back for countries like the United Kingdom (**High Street**). If *All Occurrences* is chosen, the address end is setting is ignored.

Note that choosing *One occurrence only* produces more accurate but less standardized data than choosing *All occurrences*.

**Standardise Thoroughfare Types**

**Standardize Thoroughfare Types**

Field: **NEWADD1**


Standardise thoroughfares or write in full?  Standardise  
 Full

Write the standardised thoroughfare types in:  UPPER CASE  
 Mixed Case

Change for this field:  All occurrences  
 One occurrence only

Change occurrence closest to which end of the field string?  Front  
 Back

Accept lookup table defaults



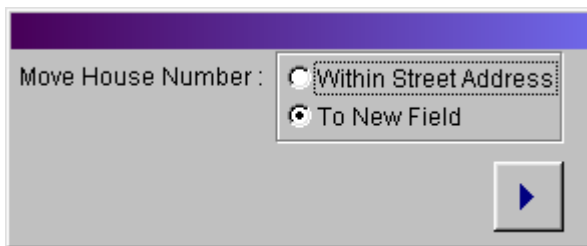
**10.29. Building numbers - move or parse****Building numbers - move or parse**

**Purpose:** to move house/building numbers from the front to the back of the street address string or vice versa, allowing commas to be added as desired after or before the house number; or to move the house number to a new field.

**Example:** **27 Wilhelmstrasse** becomes **Wilhelmstrasse 27** in Germany; **rue d’Amiens 27** becomes **27, rue d’Amiens** in France.

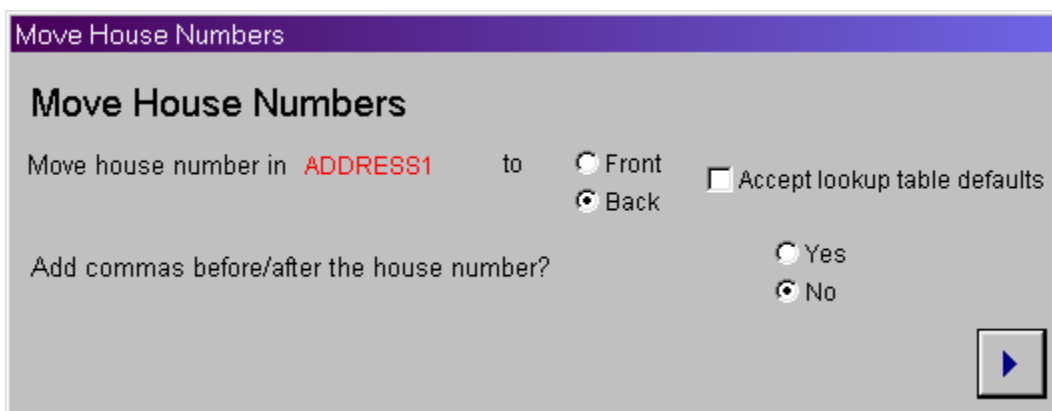
The process checks the first 10 digits for numeric characters if the house number is being moved from the front; and, because much information can follow house numbers situated at the end of street address strings, all but the first character if the house number is checked when a move is requested from the back. For this reason, the process of moving house numbers from the back can take slightly longer than the process of moving them from the front.

**Information required:** **GRC Tools™** needs to know whether to move the data within the street address strings (from front to back or vice versa) or to a new field.



If *Within Street Address* is chosen, **GRC Tools™** asks whether the house number should be moved from the front to the back or vice versa. Check “Accept lookup table formats” if you want to accept the parameters built into **GRC Tools™**, which contain the position of the house number normal for the country you have chosen. You must also specify whether commas should be added after (if the number is being moved to the front of the address string) or before (if the number is being moved to the back of the address string) the house number.

- ! If the house number structure for a country is complicated, such as for Spanish- or Portuguese-speaking countries, it is better to move the house number to a new field than to move it within the field. This prevents the “shuffling” of data within a field.




- ! Though the option to add commas is allowed for every country for the sake of flexibility, it is incorrect to add commas to the addresses of some countries. Note also that commas are added/removed for numbers moved during this process, but not for house numbers already correctly positioned. Commas can be added to/removed from these numbers using

the process **Add/Remove Commas**.

If *To new field* is chosen, **GRC Tools™** asks the name of the field to which to move the number, and whether to start the search from the front of the address string or the back. Check “Accept lookup table formats” if you want to accept the built in **GRC Tools™** parameters, which contain the position of the house number normal for the country you have chosen. Numbers often contain a suffix, for example *bis* in *23 bis rue de Paris*. You may specify a field to which to move this suffix. If you do not specify a field, the suffix will be moved to the same field as the house number. Furthermore, if moving numbers from the back of a street address, a string may occur after the house number and its suffix, as in this case: *Bahnhofstrasse 25 C Zimmer 3*. To enable you to print this data out in the correct order after running this process, you may also specify a field to which to move the string after the house number. In this case, the fields would be filled with:

*Bahnhofstrasse*  
*25 C*  
*Zimmer 3*

If no field is specified, the data after the house number and suffix is retained in the original field.

 A street address can contain a whole set of numbers, such as dates. **GRC Tools™** attempts only to identify the house number. Checking from the end of the address which is most likely to contain the house number greatly increases its accuracy.

Addresses in Switzerland have a different format depending on the language region in which the address is situated. It is therefore necessary to specify these preferences per language region, along with the name of the field where the language code (using the process **Assign language region**) is written and the codes which have been used. (When parameters are assigned on a country-by-country basis, the language region data is asked on the same screen as the other parameters as above).

Field containing language region information : LANGUAGE

Code indicating FRENCH speaking : fra

Check house numbers from

Front  
 Back

Code indicating GERMAN speaking : deu


Check house numbers from

Front  
 Back

Code indicating ITALIAN speaking : ita

Check house numbers from

Front  
 Back



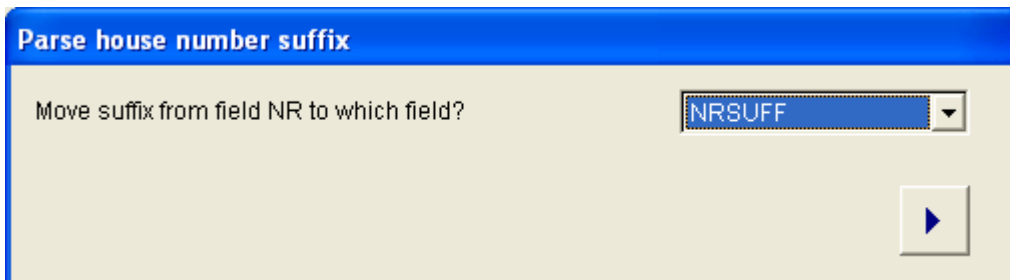
**10.30. Building number suffix - parse****Building number suffix - parse**

**Purpose:** to parse the part of the house number from the first non-numeric character to a new field.

**Example:** 27-29 becomes 27 and -29; 1d, 7th floor becomes 1 and d, 7th floor.

Though not all countries use addressing systems that include house numbers, this process can be run on data from any country for the sake of maximum flexibility.

**Information required:** For each field chosen, **GRC Tools™** needs to know to which field to move the house number suffix.



Parse house number suffix

Move suffix from field NR to which field?

NRSUFF

▶

**10.31. Commas - add or remove**

# Commas - add or remove

**Purpose:** to standardize house number/street address formats by either adding commas after (if the number is at the front of the address string) or before (if it is at the end) the house number; or by removing the commas from these places.

This process checks the first 8 characters of the street address string if the house number is at the front of the string, or the last 8 if it at the end of the string. This process takes into account not just the position of the house number, but also its position relative to other strings, such as stand-alone characters (assumed to be part of the house number) or other strings which should not be split from the house number with a comma. Spacing is also corrected. This process is best illustrated using examples:

**Examples:**

House number at front, add commas:

1,A	becomes	1, A
1 ABC	becomes	1, ABC
1 A ABC	becomes	1 A, ABC
1A ABC	becomes	1A, ABC
1 A,ABC	becomes	1A, ABC
1 BIS ABC	becomes	1 BIS, ABC

House number at back, add commas:

,1	becomes	,1
ABC 1	becomes	ABC, 1

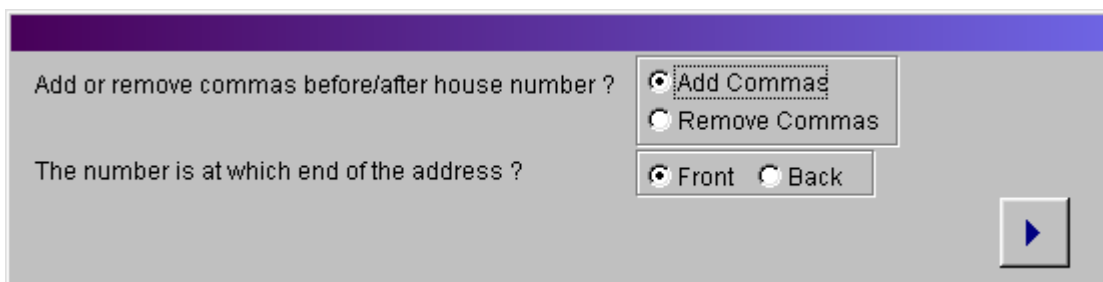
House number at front, remove commas:


1,A	becomes	1 A
1, ABC	becomes	1 ABC
1 A,ABC	becomes	1 A ABC
1A, ABC	becomes	1A ABC
1 A, ABC	becomes	1A ABC
1 BIS, ABC	becomes	1 BIS ABC


House number at back, remove commas:

ABC,1	becomes	ABC 1
ABC, 1	becomes	ABC 1

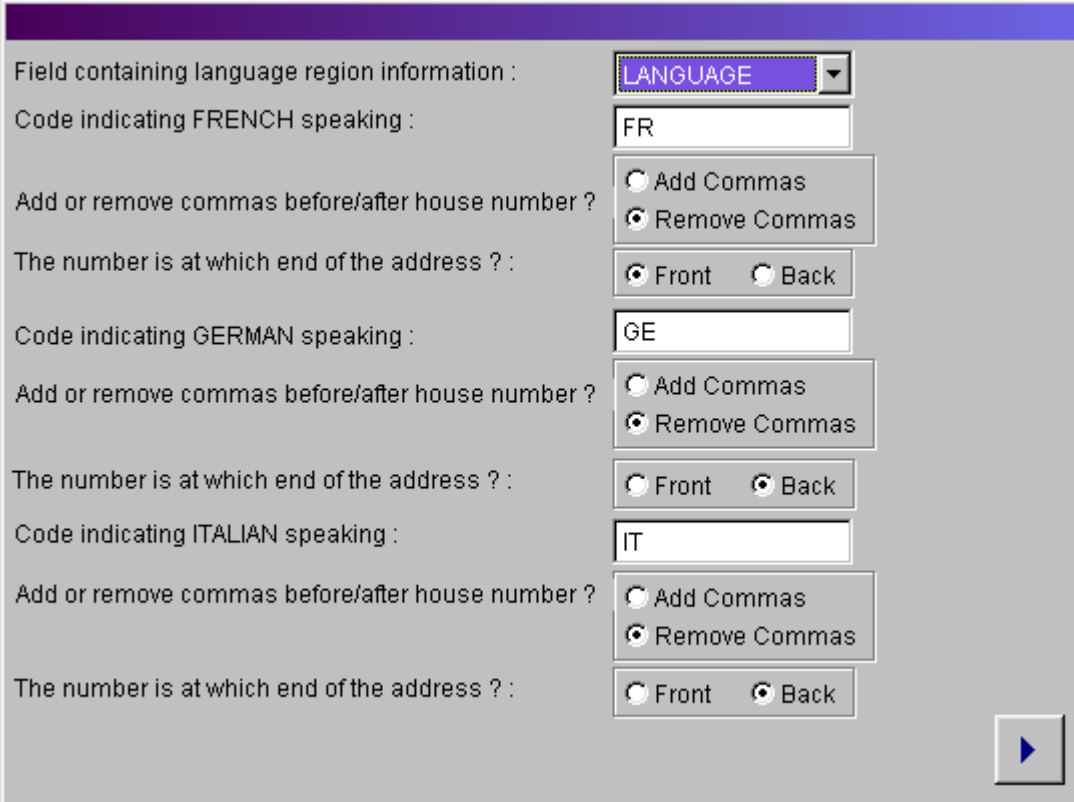
**Information required:** For each country specified, **GRC Tools™** needs to know whether commas are to be added or removed, and at which end of the address to look for the house numbers (defaulted to the recognized norm for the country concerned).



 If *apply parameters to all countries* is chosen, this process will look for house numbers at the FRONT of the field. For optimal results, choose *set parameters for each country*.

 Though the option to add commas is allowed for most countries for the sake of flexibility, it is incorrect to add commas to the addresses of some countries.

Addresses in Switzerland have a different format depending on the language region in which the address is situated. It is therefore necessary to specify the end of the field in which the number is currently written and whether commas should be added or removed per language region, along with the name of the field where the language code is written and the codes which have been used (assigned, if not already existing, using *Assign language region*).



The screenshot shows a configuration window with a purple header bar. It contains three sections for different language regions: FRENCH, GERMAN, and ITALIAN. Each section has a label, a text input field for a language code, and two radio button options for comma placement. A 'Next' button is located at the bottom right.

Language Region	Field Name	Code	Add Commas	Remove Commas	Number Position
Field containing language region information :	LANGUAGE				
Code indicating FRENCH speaking :		FR	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/> Front <input type="radio"/> Back
Add or remove commas before/after house number ?					
The number is at which end of the address ? :					
Code indicating GERMAN speaking :		GE	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/> Front <input checked="" type="radio"/> Back
Add or remove commas before/after house number ?					
The number is at which end of the address ? :					
Code indicating ITALIAN speaking :		IT	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/> Front <input checked="" type="radio"/> Back
Add or remove commas before/after house number ?					
The number is at which end of the address ? :					

**10.32. Building number/letter - format**

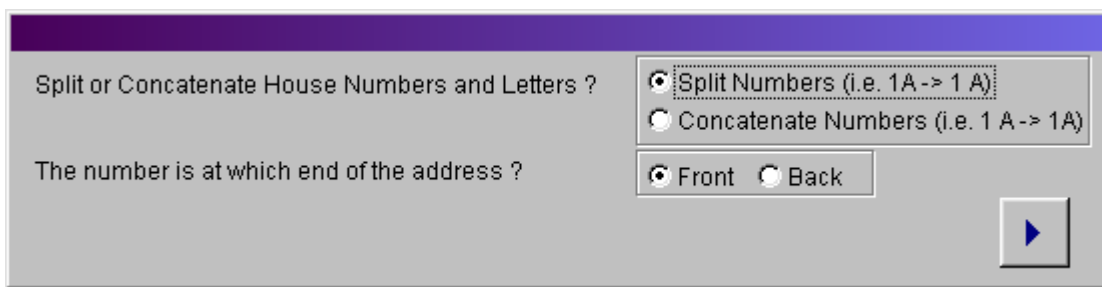
## Building number/letter - format

**Purpose:** to split or concatenate house numbers and their associated single characters.

**Example:** **1 A London Road** becomes **1A London Road**; or **1A London Road** becomes **1 A London Road**

This process checks the first 8 characters of the street address string if the house number is at the front of the string, or the last 8 if it is at the end of the string. It searches for all numbers followed by a single character followed in its turn by a space or a comma.

**Information required:** For each country specified, **GRC Tools™** needs to know at which end of the street address string the house number is situated, and whether the house numbers and their associated letters need to be concatenated (**1 A** to **1A**) or split (**1A** to **1 A**).



The screenshot shows a dialog box with a purple title bar. It contains two main sections. The first section is titled "Split or Concatenate House Numbers and Letters ?" and has two radio button options: "Split Numbers (i.e. 1A -> 1 A)" which is selected, and "Concatenate Numbers (i.e. 1 A -> 1A)". The second section is titled "The number is at which end of the address ?" and has two radio button options: "Front" which is selected, and "Back". A blue arrow button is located in the bottom right corner of the dialog box.

**!** If *apply parameters to all countries* is chosen, this process will look for house numbers at the **FRONT** of the field. For optimal results, choose *set parameters for each country*.

Addresses in Switzerland have a different format depending on the language region in which the address is situated. It is therefore necessary to specify these preferences per language region, along with the name of the field where the language code is written (by the process *Assign language region* if not already existing) and the codes which have been used.

Field containing language region information :	<input type="text" value="LANGUAGE"/>
Code indicating FRENCH speaking :	<input type="text" value="fra"/>
Split or Concatenate House Numbers and Letters ?	<input checked="" type="radio"/> Split Numbers (i.e. 1A -> 1 A) <input type="radio"/> Concatenate Numbers (i.e. 1 A -> 1A)
The number is at which end of the address ? :	<input checked="" type="radio"/> Front <input type="radio"/> Back
Code indicating GERMAN speaking :	<input type="text" value="deu"/>
Split or Concatenate House Numbers and Letters ?	<input type="radio"/> Split Numbers (i.e. 1A -> 1 A) <input checked="" type="radio"/> Concatenate Numbers (i.e. 1 A -> 1A)
The number is at which end of the address ? :	<input type="radio"/> Front <input checked="" type="radio"/> Back
Code indicating ITALIAN speaking :	<input type="text" value="ita"/>
Split or Concatenate House Numbers and Letters ?	<input type="radio"/> Split Numbers (i.e. 1A -> 1 A) <input checked="" type="radio"/> Concatenate Numbers (i.e. 1 A -> 1A)
The number is at which end of the address ? :	<input type="radio"/> Front <input checked="" type="radio"/> Back

10.33. Sorting code - parse & standardise

## Sorting code - parse & standardise

**Purpose:** to identify numbers and strings indicating sorting codes and to move these codes into a new field, standardizing the format at the same time. For certain countries, sorting code may be assigned if not found within the user-specified field.

In many countries, a sorting code is added, usually after the town name, to allow the post office of the relevant country to send the mail to the correct office for sorting. This code can consist of a string (e.g. **Paris Cédex**), a number (e.g. **Oslo 1**), a combination of the two (e.g. **Lille Cédex 2**), or the name of a province (e.g. **Milano (MI)**).

**Example:** **Lille Cedex 2** becomes **Lille** and **Cédex 2**.

Sorting codes are defined in this program as any valid string written directly after, and on the same line as, the city name in an address, and used by postal services as a sorting code, though this may be a province or region name, for example.

The process searches first for strings indicating a sorting code and then searches for any numeric characters at the end of the field specified and adds these to the sorting code written. The string indicating sorting code must be preceded by and followed by at least one space in the user-defined field to be recognized.

**Information required:** For each field chosen, **GRC Tools™** needs to know if the data in the field is to be written in upper or mixed (upper and lower) case; and the field to which to move the sorting code.

The screenshot shows a dialog box with a purple title bar. It contains two rows of controls. The first row is labeled "Is the SORTING CODE data to be written in Upper or Mixed ca" and has two radio buttons: "Upper" (unselected) and "Mixed" (selected). The second row is labeled "Move Sorting Code from CITYTEMP to which field? :" and has a dropdown menu with "SC" selected. A blue arrow button is located at the bottom right of the dialog.

In certain countries the sorting code, if not found elsewhere in the address, can be assigned on the basis of the postal code. For this reason, the field containing the postal code must be specified.

The screenshot shows a dialog box with a purple title bar labeled "Move/Correct Sorting Code". It contains a single row with the label "Name of field containing POSTCODE :" and a dropdown menu with "ZIPCODE" selected. A blue arrow button is located at the bottom right of the dialog.

In countries where the sorting code is an administrative district, such as Spain and Italy, many of the names of the administrative districts are the same as those of a city within it. To prevent the city name itself from being mistaken for a province name and being moved to a different field, the lookup tables used will not contain the city name as it would be written as a city. Thus for Italy, the lookup table may contain **MI**, the correct sorting code abbreviation for the province of Milano, or **(Milano)** as the city name will never be found between brackets, but not **Milano**.



**10.34. Place names - parse & standardise****Place names - parse & standardise**

**Purpose:** to locate, parse and standardize place names, replacing foreign or minority language forms of a town name, names incorrectly spelt or including typos, transcribed letters, incorrect diacritical marks and so on with the standardized local language equivalents.

**Example:**


<b>LONDNO</b>	
<b>16 High Street, London</b>	
<b>Londres</b>	
<b>Station House, London</b>	

becomes

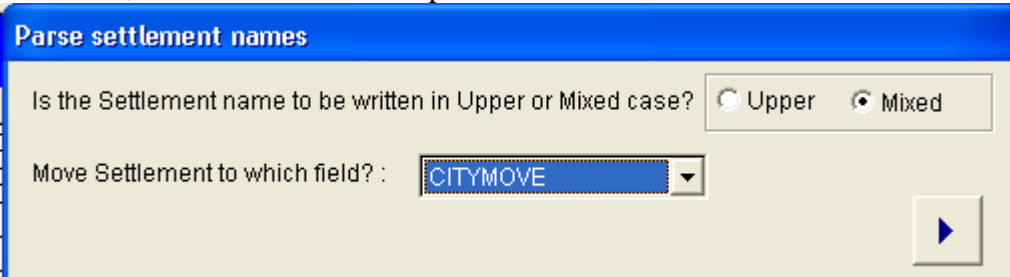
	<b>London</b>
<b>16 High Street</b>	<b>London</b>
	<b>London</b>
<b>Station House</b>	<b>London</b>

To be found, a place name must be on its own in a field or separated from the other field data with a comma. Any trailing commas after parsing are removed. A correctly formatted postal code is required.

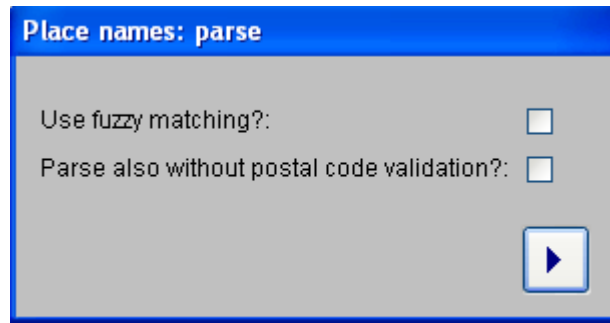
**Note:** as with all **GRC Tools™** processes, this process handles fields in alphabetical order. For parsing it is more accurate to start processing on fields which are most likely to contain the data being searched for. It may be more accurate to run the process twice or more than to parse from all fields in one pass.

 The lookup tables used by **GRC Tools™** for this process are very large. **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

**Information required:** For each field chosen, **GRC Tools™** needs to know in which case the corrected version should be written, and to which field the place name should be moved.




Place names can be parsed using two search methods – exact string searching and fuzzy matching. Fuzzy matching is less accurate than exact string matching, so it should be used with caution. Using fuzzy matching increases the number of settlements parsed. If you choose to use fuzzy matching, this is always done as well as, and after, exact string matching. It is never done instead of exact string matching.



**Place names: parse**

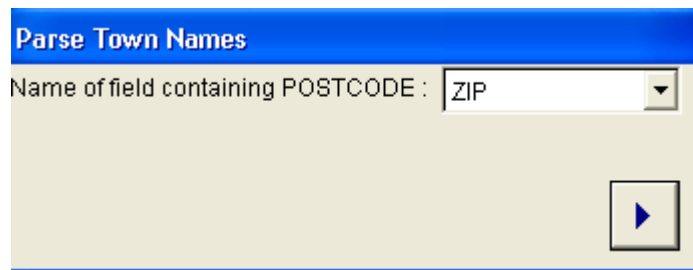
Use fuzzy matching?:

Parse also without postal code validation?:




You may also choose to attempt to parse place names without postal code validation. This is run after exact string searching and, if chosen, fuzzy matching. This is useful to locate place names where postal codes are incorrect or not in the file. However, it should be used with caution: it may find parts of addresses which are similar to place names within the country. Avoid choosing this option if your field contains a significant amount of non-place name data.

This process will only work for each town within the postal code area defined within the lookup table. This prevents address components with similar forms to settlement names being incorrectly parsed. For this reason the process requires knowing in which field the postal code is situated. The postal code should be in its correct format without punctuation or other codes such as country sorting codes (e.g. **GB-**).



**Parse Town Names**

Name of field containing POSTCODE : ZIP




**10.35. Place names - standardise**

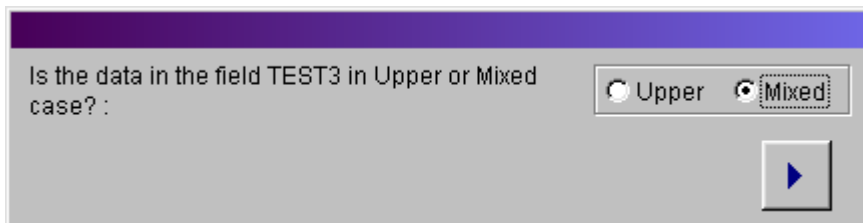
## Place names - standardise

**Purpose:** to standardize place names, replacing foreign or minority language forms of a town name, names incorrectly spelt or including typos, transcribed letters, incorrect diacritical marks and so on with the standardized local language equivalents.

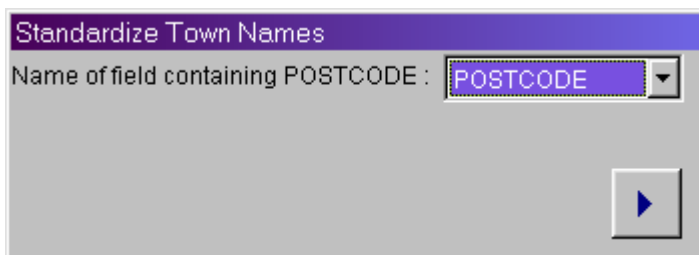
**Example:** **Londres** becomes **London** (United Kingdom); **Den Haag** and **The Hague** become **'s-Gravenhage** (The Netherlands).

 The lookup tables used by **GRC Tools™** for this process are very large. **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

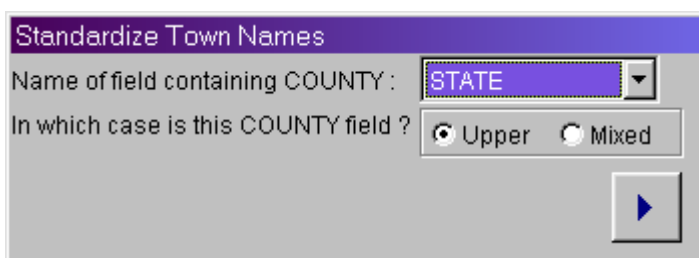
**Information required:** For each field chosen, **GRC Tools™** needs to know the case that it is in so that the replacement is written in the correct case.



This process will only work for each town within the postal code area defined within the lookup table. This prevents settlements with similar names being incorrectly re-named in different parts of the same country. For this reason the process requires knowing in which field the postal code is situated. The postal code should be in its correct format without punctuation or other codes such as country sorting codes (e.g. **GB-**).



*Ireland*, having no postal codes, is treated separately - instead of a check being made on postal code, it is made on county. This makes the process less efficient but retains the accuracy. If Ireland is chosen, you need to specify the field containing the county in Ireland, and the case in which it is stored.



Most lookup tables are shipped with **GRC Tools™** structured to change foreign words for town names into the local-language equivalent only. However, for countries where a number of languages exists, the rules used for the lookup tables are as follows.

- *Belgium*: language areas are fixed by law - the names are altered to the local-language name as defined by law. In mixed language areas, the majority language name is used. Brussels, which is purely bilingual, is excluded from this process.
- *Finland*: the town name assigned (Finnish or Swedish) is that of the majority of speakers in the town concerned.
- *France*: in all cases, the French name is assigned.
- *Greenland*: the Greenlandic rather than the Danish name is assigned.
- *Ireland*: in *An Ghaeltacht* (Irish-speaking areas) the Irish name is assigned. In all other cases, the English name is assigned.
- *Italy*: the Italian name is assigned in all cases.
- *Malta*: the Maltese name is assigned in all cases.
- *The Netherlands*: the Dutch names are assigned in all cases. *Den Haag* is changed to '*s-Gravenhage*'; *Den Bosch* is changed to '*s-Hertogenbosch*'.
- *Spain*: Catalan/Valencian names are assigned for towns within the provinces of Catalonia, the Balearic Islands and Valencia. Basque names are assigned for towns within the province of País-Vasco. Galician names are assigned for towns within the province of Galicia. In Navarra, town names are assigned (Basque or Spanish) on the basis of the majority language within each town. In all other cases, Spanish is assigned.
- *Switzerland*: Town names are assigned on the basis of majority language within each town. The bilingual town of Biel/Bienne becomes in all cases *Biel/Bienne*.

**10.36. Provinces / Regions - parse / assign**

## Provinces - parse/assign Provinces/regions - assign Regions - assign

**Purpose:** to identify strings indicating administrative districts (provinces, counties, states, cantons etc.) and to move these strings into a new field, standardizing the format at the same time. For certain countries, province and region name may be assigned if not found within the user-specified field.

**Provinces - parse/assign** is used for countries where provinces are used in addresses and where provinces which are not found can be assigned on the basis of postal code.

**Provinces/regions - assign** is used where province and/or region name can be assigned on the basis of postal code.

**Regions - assign** is used where region names can be assigned on the basis of postal code.

**Example:** **Milano (MILANO)** becomes **Milano** and **MI** and the region name **Lombardia** is assigned.

The process searches first for strings indicating an administrative district. The string indicating administrative district must be preceded and followed by at least one space in the field specified to be recognized. Should no administrative district be found, and postal codes are available within the lookup table, province names (and in certain cases region names) are assigned.

**Information required:** For each field chosen, **GRC Tools™** needs to know if the province name is to be written in upper or mixed (upper and lower) case; and the field to which to move/assign the province name.

When **Assign provinces/regions** is chosen, **GRC Tools™** needs to know the case in which the province/region data is to be written and the field to which this data is to be moved (if this field already has data in it, it will not be overwritten). Also required is the name of the field containing the postal code. The postal code should be contained in this field in the correct format, without being preceded by a country postal sorting code (**GB-** etc.).

For certain countries, it is possible to assign the province/region on the basis of postal code. **GRC Tools™** therefore requires the name of the field containing the postal code; the name of the field to which the region name is to be written, and the case in which it is to be written.

! In countries such as Spain and Italy many of the names of the administrative districts are the same as those of a city within it. To prevent the city name itself from being mistaken for a province name and being moved to a different field, the lookup table does not contain the city name as it would be written as a city. Thus for Italy, the lookup table may contain **(Milano)** as the city name will never be found between brackets, but not **Milano**; **Oxon** and **Oxfordshire** but not **Oxford** for the United Kingdom and so on. Equally, if the name of one region is the same as part of the name of another (for example *Lothian* and *East Lothian*, *Yorkshire* and *West Yorkshire* and so on), the shorter form has not been added to the lookup table to prevent the longer form from being incorrectly split, leaving, for example, *West* in one field and moving only the *Yorkshire*.

! Avoid running this process over fields containing street addresses. Region names which are part of street addresses, as in *27 Avon Road* will be moved.

- !
- **France:** *Département* names are never found in addresses. These can therefore be assigned on the basis of postal code, they are not otherwise listed in the lookup tables.
  - **Italy:** Provinces and regions strings are searched for, and, if not found, provinces and regions are assigned on the basis of postal code. By default the region names are written in full. If the correct abbreviations (used as a sorting code) are required, use the process **Move/Correct Sorting Code**
  - **Spain** Provinces and regions strings are searched for, and, if not found, provinces and regions are assigned on the basis of postal code. By default the region names are written in full. If they are required between brackets (used as a sorting code), use the process **Move/Correct Sorting Code**
  - **Switzerland:** Canton names are often found in addresses but should not be used. They are searched for but cannot be assigned if none is found.

**10.37. University strings - parse**

## University strings - parse

**Purpose:** to locate, parse and standardize university strings.

**Example:**


**Univ. of Hull**

is parsed and becomes

**University of Hull**

To be parsed, a university string must be on its own in a field. This prevents strings such as *University Road* being incorrectly parsed. Fields containing any other apparent address information will not be parsed.

**Note:** as with all **GRC Tools™** processes, this process handles fields in alphabetical order. For parsing it is more accurate to start processing on fields which are most likely to contain the data being searched for. It may be more accurate to run the process twice or more than to parse from all fields in one pass.

 The lookup tables used by **GRC Tools™** for this process are very large. **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

**Information required:** For each field chosen, **GRC Tools™** needs to know whether the data should be written in a *standardized* form or in *full*. By default, **GRC Tools™** writes the data in its long form. *Standardize* will produce more standardized but less accurate results; *full* will produce less standardized but more accurate results. E.g.:

*Full form:* **University of Bristol**

*Standardised form:* **Univ. of Bristol**

You need to specify whether the data each field contains is in upper-case or in mixed (upper and lower) case to ensure that the data is correctly replaced.

Specify the field to which the string will be parsed.

**Standardise Thoroughfare Types****Parse university strings**Field: **COMP1**Standardise university strings or write in full?  Standardise  
 FullWrite the standardised university strings in:  UPPER CASE  
 Mixed Case

Parse university string to which field:



---

**10.38. Building strings - move field contents**

## Building strings - move field contents

**Purpose:** to locate, parse and standardize fields containing building strings.

**Example:**


**Grand View Hse**

is parsed and becomes

**Grand View House**

By default, to be parsed a building string must be on its own in a field. This prevents strings such as *Station Road* being incorrectly parsed. Fields containing any other apparent address information will not be parsed. However, this also prevents building strings such as *University House* from being moved. You can choose to allowing parsing of fields containing non-building address elements by ignoring opposing types (see below).

**Note:** as with all **GRC Tools™** processes, this process handles fields in alphabetical order. For parsing it is more accurate to start processing on fields which are most likely to contain the data being searched for. It may be more accurate to run the process twice or more than to parse from all fields in one pass.

 The lookup tables used by **GRC Tools™** for this process are very large. **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

**Information required:** For each field chosen, **GRC Tools™** needs to know whether the data should be written in a *standardized* form or in *full*. By default, **GRC Tools™** writes the data in its long form. *Standardize* will produce more standardized but less accurate results; *full* will produce less standardized but more accurate results.

You need to specify whether the data each field contains is in upper-case or in mixed (upper and lower) case to ensure that the data is correctly replaced.

Specify the field to which the string will be parsed.

**Building strings - move field contents**


Field: **GRCBUILD2**

Standardise building strings or write in full?  Standardise  
 Full

Write the building strings in:  UPPER CASE  
 Mixed Case

Parse building string to which field:

Ignore opposing types:



To move data in fields which contain non-building address elements, such as "University" in *University House*, check the "Ignore opposing types" box.

### 10.39. Attention strings - move field contents

## Attention strings - move field contents

**Purpose:** to move the contents of fields containing attention strings, such as ATTN:, C/O etc.

**Example:**

**Attn: Miss Amy Grundwick**

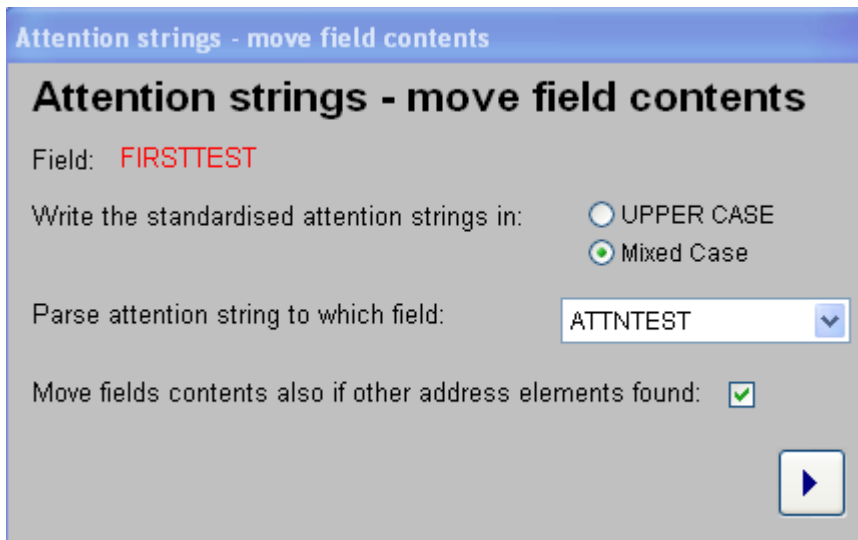
is parsed and becomes

**Attention: Miss Amy Grundwick**

**Note:** as with all **GRC Tools™** processes, this process handles fields in alphabetical order. For parsing it is more accurate to start processing on fields which are most likely to contain the data being searched for. It may be more accurate to run the process twice or more than to parse from all fields in one pass.

**Information required:** For each field chosen, **GRC Tools™** needs to know whether the data each field contains is in upper-case or in mixed (upper and lower) case to ensure that the data is correctly replaced.

Specify the field to which the string will be parsed.



By default, to be parsed an attention string must be on its own in a field. Fields containing other address elements, such as

**Attn: Miss Amy Grundwick, Physics Department**

will not be parsed. This can be overridden by checking the "Move field contents also if other address elements found" box.

**10.40. Department strings - move field contents**

## Department strings - move field contents

**Purpose:** to move the contents of fields containing only department strings.

**Example:**


**Dept. of Linguistics**

is parsed and becomes

**Department of Linguistics**

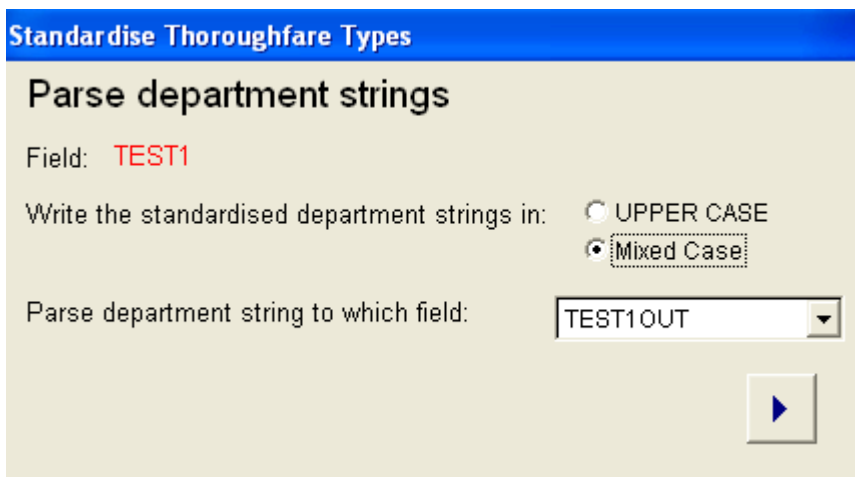
To be parsed, a department string must be on its own in a field. Fields containing any other apparent address information will not be parsed.

**Note:** as with all **GRC Tools™** processes, this process handles fields in alphabetical order. For parsing it is more accurate to start processing on fields which are most likely to contain the data being searched for. It may be more accurate to run the process twice or more than to parse from all fields in one pass.

 The lookup tables used by **GRC Tools™** for this process are very large. **GRC Tools™** may appear to hang for some time at 0% and 100% during processing. This is normal - avoid interrupting the program at these points.

**Information required:** For each field chosen, **GRC Tools™** needs to know whether the data each field contains is in upper-case or in mixed (upper and lower) case to ensure that the data is correctly replaced.

Specify the field to which the string will be parsed.



The screenshot shows a dialog box titled "Standardise Thoroughfare Types" with a sub-section "Parse department strings". The "Field:" is set to "TEST1". Under "Write the standardised department strings in:", the "Mixed Case" radio button is selected. The "Parse department string to which field:" dropdown menu is set to "TEST1OUT". A blue play button is located at the bottom right of the dialog.

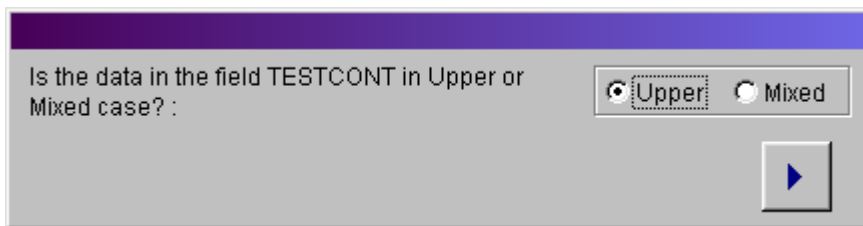
### 10.41. Personal name prefixes - parse & standardise

## Personal name prefixes - parse & standardise

**Purpose:** to identify strings indication forms of address and other constants at the front of the personal name and to move these to a new field, make the strings consistent at the same time. This process will continue searching until no more prefixes are found, so multiple prefixes such as "Prof. Dr." can be parsed.

**Example:** **Mister Smith** becomes **Mr** and **Smith**; **Mons. Dupont** becomes **M.** and **Dupont**.

**Information required:** For each field chosen, **GRC Tools™** needs to know if the prefix is to be written in upper or mixed (upper and lower) case.

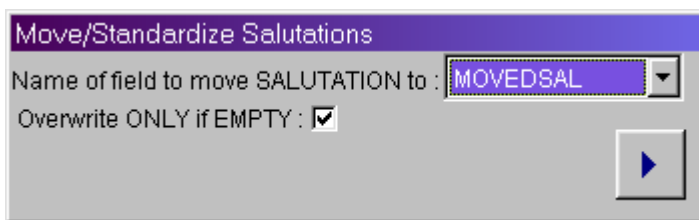


Is the data in the field TESTCONT in Upper or Mixed case? :

Upper  Mixed

▶

You must then specify to which field to move the prefix, if found, and whether this field is to be overwritten in all cases or only if it is empty.



Move/Standardize Salutations

Name of field to move SALUTATION to : MOVEDSAL

Overwrite ONLY if EMPTY :

▶

! When a form of address can consist of a single letter, for example **M** to indicate **Monsieur** in French-speaking countries, then this has not been added to the lookup table as **GRC Tools™** cannot distinguish between **M Dupont** where the **M** indicates, for example, **Michel** and **M Dupont** where the **M** indicates **Monsieur**.

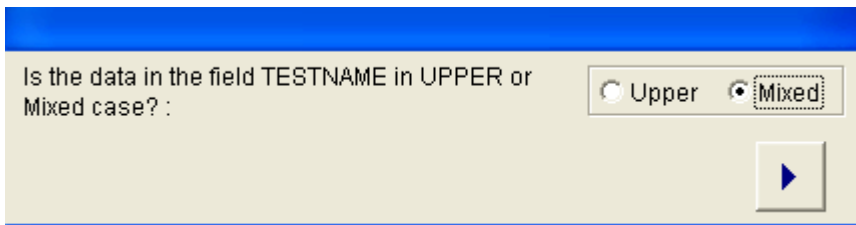
**10.42. Personal name suffixes - parse & standardise**

## Personal name suffixes - parse & standardise

**Purpose:** to identify strings indicating academic titles, seniority indicators and other constants at the end of the personal name and to move these to a new field, make the strings consistent at the same time. This process will continue searching until no more suffixes are found, so multiple suffixes such as "OBE MBE" can be parsed.

**Example:** **Graham Rhind PHD** becomes **Graham Rhind** and **Ph.D.**

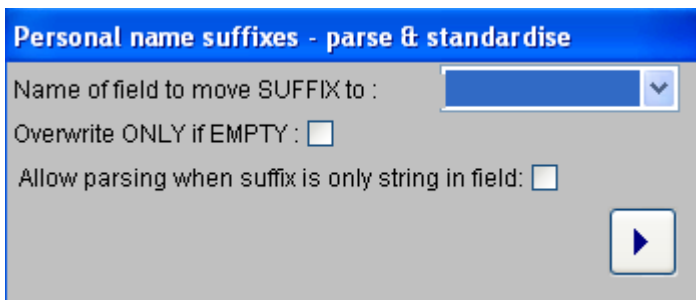
**Information required:** For each field chosen, **GRC Tools™** needs to know whether the suffix is to be written in upper or mixed (upper and lower) case.



Is the data in the field TESTNAME in UPPER or Mixed case? :

Upper  Mixed

You must then specify to which field to move the suffix, if found, and whether this field is to be overwritten in all cases or only if it is empty.



**Personal name suffixes - parse & standardise**

Name of field to move SUFFIX to :

Overwrite ONLY if EMPTY :

Allow parsing when suffix is only string in field:

By default this process will only parse a string which is suffixed to another string. You may have data where a field contains only "suffix" data. If this is the case and you would like also to parse that data, check the "Allow parsing when suffix in only string in field" box.

**10.43. Personal name middle name(s)/initial(s) - parse****Personal name middle name(s)/initial(s) - parse**

**Purpose:** it is not possible to distinguish definitively between first given names and middle given names, so this process makes no attempt to validate the name strings found within a field. This process simply splits a personal name string after the first full stop or space found.

**Example:**

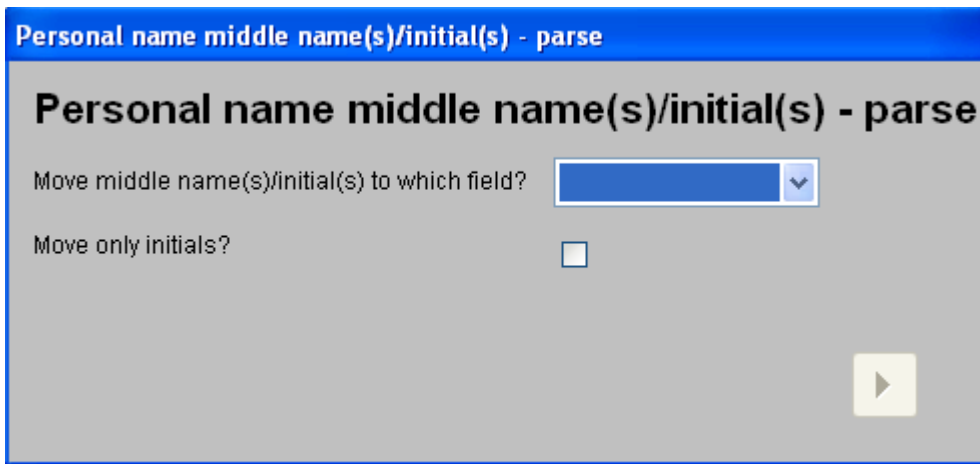
**Graham Robert** becomes **Graham** and **Robert**

**Graham R.** becomes **Graham** and **R.**

**Jean-Paul Robert** becomes **Jean-Paul** and **Robert**

**J.G.P.L.** becomes **J.** and **G.P.L.**

**Information required:** GRC Tools™ needs to know the field to which to move the middle name(s)/initial(s).



If you would like to move only initials to the middle name field, choose the *Move only initials?* option. Choosing this option means that:

**Graham Robert** remains unchanged

**Graham R.** becomes **Graham** and **R.**

**Jean-Paul Robert** remains unchanged

**J.G.P.L.** becomes **J.** and **G.P.L.**

**10.44. Job titles - standardise**

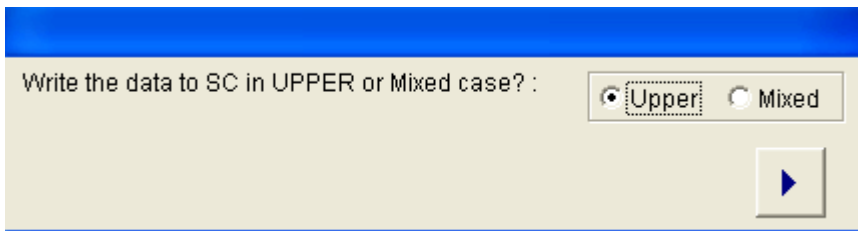
## Job titles - standardise

**Purpose:** to standardize titles names into their fullest correct equivalent.

*This process costs four credits per altered field.*

**Example:** C.E.O. becomes **Chief Executive Officer**; Mging Dir. becomes **Managing director**

**Information required:** For each field chosen, **GRC Tools™** needs to know in which case to write the correct job title data.



The image shows a dialog box with a blue title bar. The main area is light beige. It contains the text "Write the data to SC in UPPER or Mixed case? :". To the right of this text are two radio buttons: "Upper" (which is selected) and "Mixed". Below the radio buttons is a button with a right-pointing arrow.

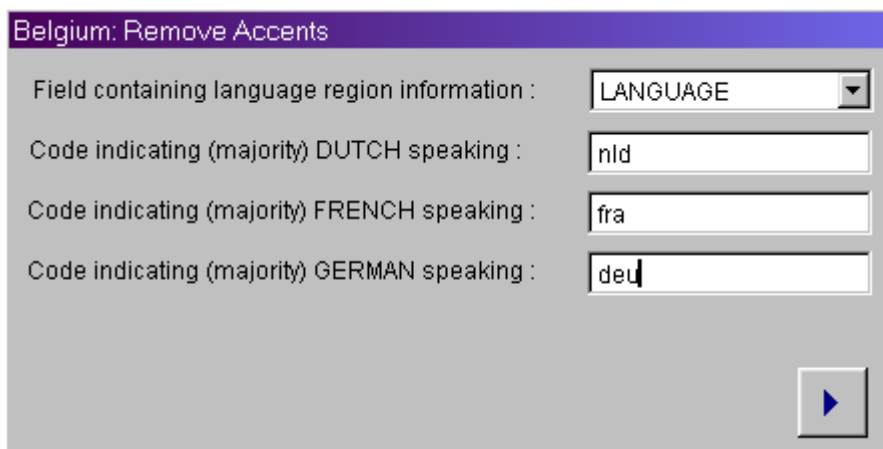
**Note:** this process writes the job title in its most expanded correct form. Thus, **C.E.O.** (English) becomes **Chief Executive Officer**; **Dir. Technique** (French) remains **Dir. Technique** because it is not know if the job title owner is male (**Directeur**) or female (**Directrice**). This being so, the job titles in the database can be very long – the longest currently in the table is 120 characters long. **Ensure that the field which you are processing is at least this long, to avoid data being truncated.**

**10.45. Accents - remove****Accents - remove**

**Purpose:** to replace accented characters in the user-defined field with the correct non-accented equivalents.

**Information required:** there is a clash in equivalent characters in Belgium between German and Dutch/French; in Switzerland between German, French and Italian; and in Canada between French and English. For this reason, this process handles each language area differently. It is therefore necessary to have written the language areas to the database file using the procedure

**Assign Language Region.** When running this process for Belgium, Switzerland or Canada, you are required to specify which field contains the language region and which codes are used for each language.



Belgium: Remove Accents

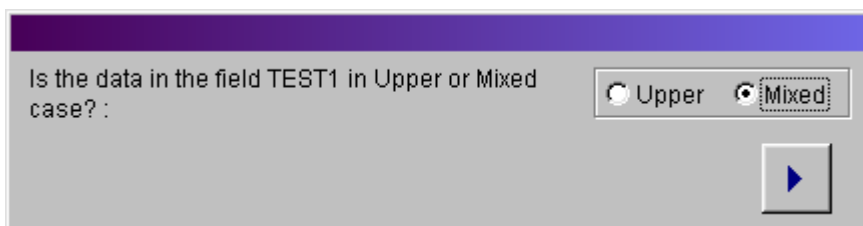
Field containing language region information : LANGUAGE

Code indicating (majority) DUTCH speaking : nld

Code indicating (majority) FRENCH speaking : fra

Code indicating (majority) GERMAN speaking : deu

For each field chosen for processing, **GRC Tools™** needs to know which case the data is in. Though it may seem self-evident that an upper-case accented-letter is replaced by an equivalent upper-case letter without accent, for some languages an accented character may be replaced by more than one letter and the casing of these letters needs to be assigned. For example, the German **Ü** is replaced by **Ue** at the start of a mixed-case word and by **UE** at the start of an upper case string.



Is the data in the field TEST1 in Upper or Mixed case? :

Upper  Mixed

## **11. Frequently asked questions**

### 11.1. The different parts of my addresses in my database are all mixed up. How can GRC Tools help me to parse this data into different fields?

“The different parts of my addresses in my database are all mixed up. How can GRC Tools help me to parse this data into different fields?”

Your data might look something like this:

Country	Address1	Address2	Address3
NL	Postbus 15213, Nieuwe Prinsengracht 80	1001 ME	Amsterdam
NL	Nwe Prinsengracht 80	1001 ME AMSTERDAM	
GB	Station House	17 Hull Road	Cottingham HU7 7SL
I	via Milano 200	Casa Postale 21	13245 BIELLA
F	BP 9	17, rue de la Gare	75689 PARIS
F	rue de la Gare 17	PARIS	

Data such as the postal code and the postbox address has been entered simply in three fields: address1, address2 and address3. You want to be able to parse the data into different fields so that it is easier to process. The country has already been assigned (this is a requirement for GRC Tools).

For this file you would choose to run these processes:

- Locate postal codes - this finds the postal codes which are currently floating in the address fields.
- Write located postal code - this moves the postal code which has been found to a new field.
- Parse post office box numbers - this will find the post office box addresses which are mixed up with the street addresses, and move them to a new field.
- Parse building numbers - this will move the house numbers to a new field.

To run these processes, you need to make sure that your table has fields to which to move the data, so you need to add fields to hold the postal code, the post office box number and the house number:

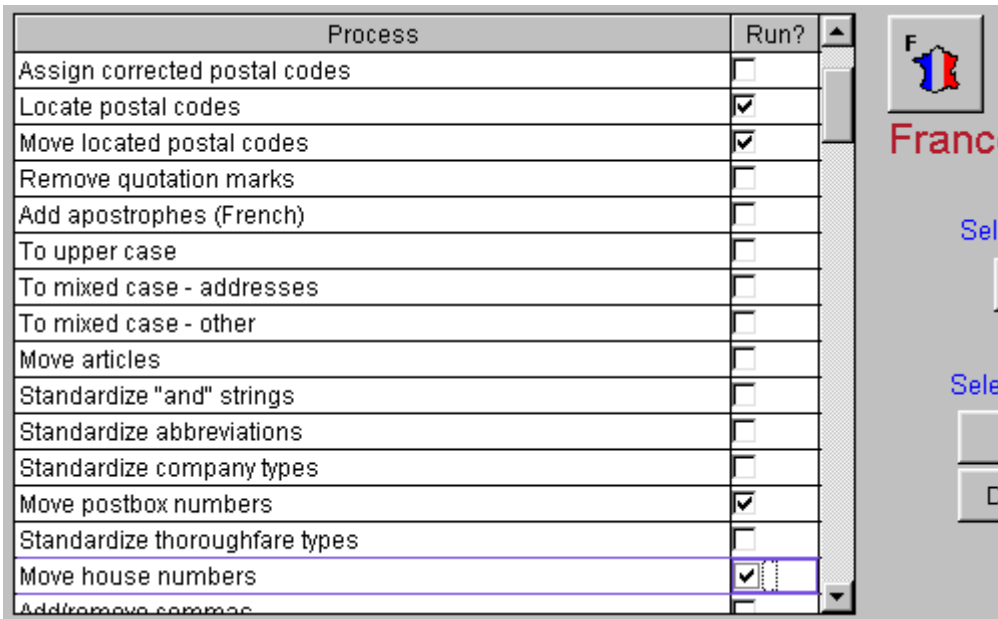
Country	Address1	Housenr	Address2	Address3	Postbox	Postalcode
NL	Postbus 15213, Nieuwe Prinsengracht 80		1001 ME	Amsterdam		
NL	Nwe Prinsengracht 80		1001 ME AMSTERDAM			
GB	Station House		17 Hull Road	Cottingham HU7 7SL		
I	via Milano 200		Casa Postale 21	13245 BIELLA		
F	BP 9		17, rue de la Gare	75689 PARIS		
F	rue de la Gare 17		PARIS			

Run GRC Tools. Specify that the country code is stored in the field *country*, link the codes to the country concerned, click *apply same fields to each process to all countries* (because you are running the same processes over the same fields for all countries), and continue.

The screenshot shows a window with a list of countries on the left: F --> France, GB --> United Kingdom, I --> Italy, and NL --> Netherlands, The. To the right of the list are two buttons: "Remove Link" and "Remove all links". Below the list, there is a checkbox labeled "Apply same fields for each process to all countries?" which is checked. To the right of this checkbox are two radio buttons: "Apply parameters to all countries" (which is selected) and "Set parameters for each country".

For each country, choose the four processes *locate postal codes*, *write located postal codes*, *parse post office box*

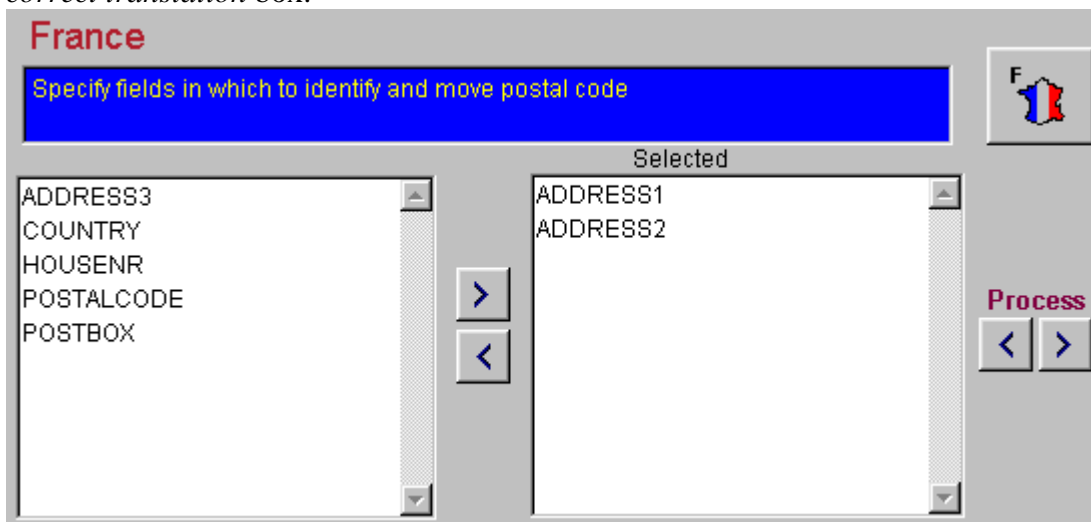
*numbers and parse building numbers.*



Repeat for each country.

You know that in your data the postal code can be in the field address2 or the field address3, so choose these fields for the process *locate postal codes*. Note the name of the file that GRC Tools will automatically create to hold the postal codes found. Add the name of this file to the *write located postal codes* dialogue, along with the name of the field to contain the moved postal code.

Post office boxes are found in your data either in field address1 or field address2, so give these fields in the *choose fields to process* screen for the *parse post office box numbers* process. In the dialogue provide the name of the field to move the post office box to. The other fields are not required for this case, so you can leave them blank. If you want to retain the word “Post office box” before the number which is moved, click the *retain word POSTBOX (in correct translation box)*.



**Move Located Postcodes**

Choose field containing UNIQUE ID CODE:

Check to use RECORD NUMBERS instead of ID:

Choose file from which to move postcodes :

Field to which to move postcode :

**Move Postbox Numbers**

Name of field to move POSTBOX TO :

Name of field to move POSTCODE FROM :

Name of field to move POSTCODE TO :

Overwrite ONLY if EMPTY :

Name of field to move TOWN NAME FROM :

Name of field to move TOWN NAME TO :

Overwrite ONLY if EMPTY :

Retain word POSTBOX (in correct translation)?

In your data the house numbers are found in either field address1 or field address2. Specify these fields in the *parse building numbers* field choice screen. You want to move the numbers to a new field, so specify this at the next dialogue.

Move House Number :  Within Street Address  
 To New Field

**Move House Numbers**

Move House Number from field ADDRESS1 to which new field?:

Check for house number from:  Front or  Back  Accept lookup table defaults

Move house number suffix to which field:

**Moving FROM BACK only**

Specify the name of the field to which the house number is to be moved, and check the *accept lookup table defaults* box, which means that GRC Tools will look in the normal place for this country for the number. Repeat for each country.

GRC Tools will now process your data. This is how your data will look at the end of the processing:

Country	Address1	House	Address2	Address3	Postbox	Postalcode
NL	Nieuwe Prinsengracht	80		Amsterdam	Postbus 15213	1001 ME
NL	Nwe Prinsengracht	80	AMSTERDAM			1001 ME
GB	Station House	17	Hull Road	Cottingham		HU7 7SL
I	via Milano	200		BIELLA	Case Postale 21	13245
F		17	rue de la Gare	PARIS	Boite Postale 9	75689
F	rue de la Gare 17		PARIS			

You will note that the postal code has in all cases been found and move. The postbox numbers have been found and moved, and the postbox strings have been corrected and standardised. The house numbers have all been moved, except for those which are found at the wrong end of the street address (e.g. rue de la Gare 17 in France). To move these, re-run the *parse building numbers* process, asking for the program to search from the back of the street address string for France.

## 11.2. How can GRC Tools help to standardise and format my data to make it more suitable for mailing and to improve the performance of de-duplication and postal validation programs?

“How can GRC Tools help to standardise and format my data to make it more suitable for mailing and to improve the performance of de-duplication and postal validation programs?”

Your data might look something like this:

Country	Address	City	Pc
NL	Stationstraat 25	s Gravenhage	2584 AP
NL	25 Stationstr	NL - THE HAGUE	2509LL
NL	Molenln 19	S GRAVENHAGE	2562 EH
NL	MOLENLAAN 19	S' GRAVENHAGE	2512EW
NL	23 ZEEZICHT	SCHEVENINGEN	2585
NL	MOLENLAAN 19	DEN HAAG	2532 BB
NL	Nw PRINSENGR. 1	DEN HAAG	2517 EA
NL	1 NWE PRINSENGRACHT	HAGUE SOUTH NETHERLANDS	2586

There are a number of problems with this file:

- The data is not all in the same case
- The street names have been written inconsistently
- The house number is written in different places in the street address (at the front or at the back)
- The city name is not written consistently
- The postal code is not written consistently.

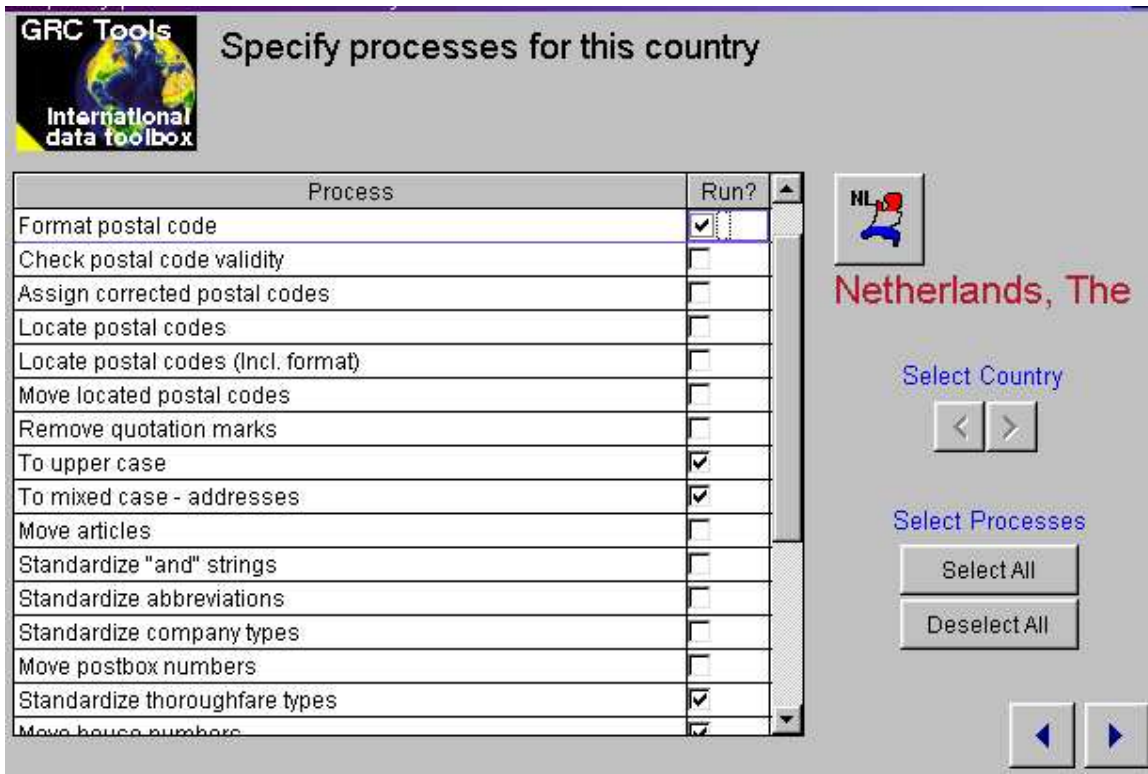
For this file you would choose to run these processes:

- To mixed case (- addresses) for the street address and To upper case for the city name.
- Standardise thoroughfare strings.
- Parse building numbers (from the front to the back)
- Standardise place names.
- Format postal code to correct the postal code inconsistencies.

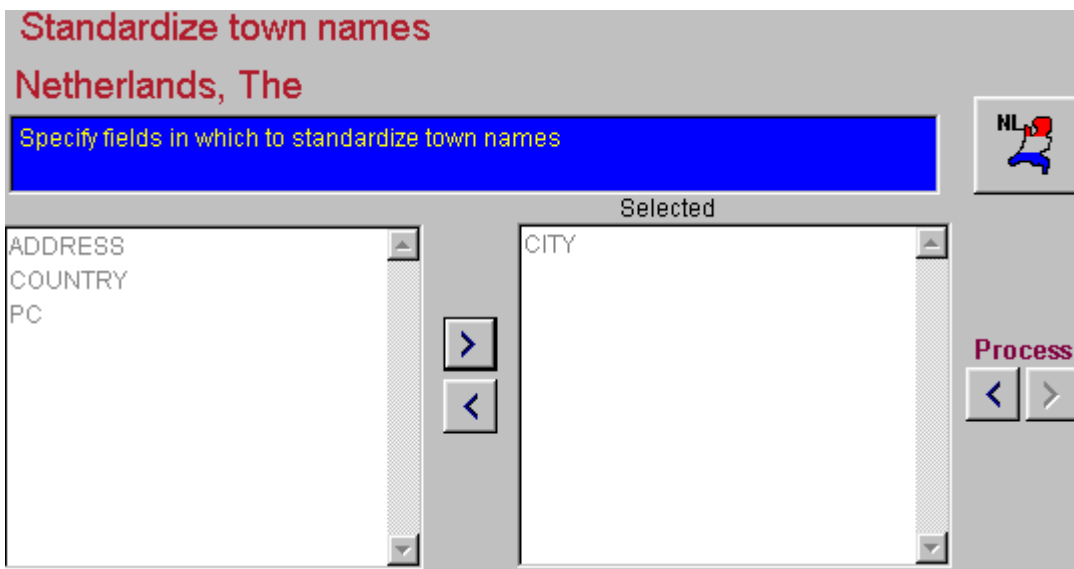
You will not need to change your table structure to run these processes.

Run GRC Tools. Specify that the country code is stored in the field *country*, link the codes to the country concerned and continue.

For this country, choose the processes *Format postal codes*, *To upper case*, *To mixed case - addresses*, *Standardize thoroughfare strings*, *Build building numbers* and *Standardise place names*.



For each process, specify the field in which the data which is to be processed is in - *PC* for *Format postal code*, *city* for *To upper case*, *address* for *To mixed case - addresses*, *address* for *Standardise thoroughfare types*, *address* for *Move house numbers* and *city* for *Standardise town names*. For some of these processes you will be asked some additional questions according to your preferences for the data.



GRC Tools will now process your data. This is how your data will look at the end of the processing:

---

Country	Address	City	Pc
NL	Stationstraat 25	'S-GRAVENHAGE	2584 AP
NL	Stationstraat 25	'S-GRAVENHAGE	2509 LL
NL	Molenh 19	'S-GRAVENHAGE	2562 EH
NL	Molenh 19	'S-GRAVENHAGE	2512 EW
NL	Zeezicht 23	'S-GRAVENHAGE	2585
NL	Molenh 19	'S-GRAVENHAGE	2532 BB
NL	Nw Prinsengr. 1	'S-GRAVENHAGE	2517 EA
NL	Nwe Prinsengr. 1	'S-GRAVENHAGE	2586

You will note that GRC Tools has correctly cased all strings, that the city names and street types have been standardised, house number position and postal code format have been corrected. This file is now suitable for output as addresses and these changes can massively improve the results of de-duplication and postal validation programs.

---

**12. Technical Support**

# Technical Support

**GRC Tools™** gives a great deal of flexibility to the user to choose how he/she wants his/her data to look after processing. In the same way, this flexibility provides plenty of opportunity for the user to make unwanted changes to data through misuse or incorrect assigning of parameters, and different data formats may produce different results than expected.

Always backup your data before using **GRC Tools™**. GRC Database Information accepts no responsibility for lost or damaged data through the use or misuse of **GRC Tools™**.

Please check the list of frequently encountered problems below.

Before assuming that **GRC Tools™** is not working, please try to close then restart **GRC Tools™** and re-run the same process(es), checking that all parameters are correctly filled in and that the fields are correctly assigned. Check the database over which **GRC Tools™** is being run - are the country codes present? Do they match those given in **GRC Tools™**? Are the parameters specified logical given the data to be processed? If you run the process using the **All Countries** option, try running again allowing options to be assigned on a country-by-country basis, which improves accuracy.

If all else fails, let us know.

GRC Database Information provides an e-mail only support service (address: [graham@grcdi.nl](mailto:graham@grcdi.nl)). You are guaranteed the full-attention of the author of the software. When looking for solutions to problems, we must be able to re-construct the problem at GRC Database Information. We must therefore have the following information:

- Your name, company name, address, telephone number and e-mail address
- The serial number of your version of **GRC Tools™**
- The operating system that you are using
- Which process(es) are failing
- A full description of what happens (what happens on the screen; what error message (if any) do you get; at what point (if any) does failure occur)
- What the data in the field (exactly!) looked like before the processing
- What the data in the field (exactly!) looked like after the processing
- Which parameters you chose to run this process
- Which process(es) you ran immediately prior to getting to the process where the problem(s) occurred.

We shall do our best to solve your problems, or, if **GRC Tools™** was deliberately designed to do something that you did not expect, why this is the case. If we have made an error, we shall try to get a new version of the software to you as quickly as possible, and provide you with free credits as compensation and thanks for finding errors we were not able to find.

If you have any suggestions for changes or improvements to **GRC Tools™** we would gratefully receive these at the same e-mail address. In this way we can ensure that **GRC Tools™** is the tool you need and not the tool we think you need.

## 12.1. Frequently encountered problems

### Frequently encountered problems

#### “Cannot access the selected table”

This error occurs occasionally after processing has completed. It does not affect processing so can be ignored.

#### **Error 3: File is in use**

This error is known to occur intermittently with Excel 97 files, when opening the file. The solution is to save the Excel file under another format, preferably dbf, and use the new file format as the input file.

#### **Error: A subdirectory or file c:\temp\ already exists**

This error is known to occur on some Windows NT systems where the c:\temp\ directory is a hidden system directory. GRCTools needs to write temporary files to this directory. If the directory is hidden GRCTools cannot locate the directory, but equally it cannot create the directory so this error occurs. To resolve this, the directory needs to be made into a “standard” directory, either by changing its properties, or by renaming it and creating a new c:\temp\ directory.

#### **Data type mismatch error**

Check that you are not trying to process a field which is not character/alphanumeric in type. Numeric and other fields, which should not be used to store address data, will give this error.

#### **Duplicate field names error**

The native format of **GRCTools™** is Visual FoxPro/dBase III+. Data imported in other formats is imported to Visual Foxpro format for processing, and then exported to the original format after processing. Data being imported must therefore be able successfully to be converted. **GRCTools™** supports field names of 10 characters or less. If your data contains long field names, they are truncated during import. If this truncation leads to duplicate field names, **GRC Tools™** will fail. For example, if you have the field names:

```
address_line_1  
address_line_2
```

in your incoming file, **GRCTools™** will make these into:

```
address_li  
address_li
```

so that you get a duplicate field names error. Renaming the fields to, for example:

```
address1  
address2
```

will resolve this problem.

#### **Error 112: Invalid key length**

**GRC Tools™** indexes data as required during processing. If field lengths are too long (for example, 256, to which fields may default when exported from other formats) **GRC Tools™** is unable to create indexes and gives this error.

Reduce the field lengths (e.g. to 100 or less) to prevent this error.

### Exception code=c0000005

This is a Windows general protection fault error, and is usually related to hardware issues, old or unstable printer drivers etc. Try deleting the files foxuser.dbf and foxuser.fpt from the GRCTools directory and rerunning the program. If this does not help, please report the problem to us but it is likely that it can only be resolved through cleaning up the PC concerned or installing **GRCTools™** on another PC.

### Error creating DEFAULT key container

This rare error is, we believe, due to insufficient read/write privileges on the computer you are using. A solution may be found at <http://support.microsoft.com/kb/q246459/>. Please report this error to us if you are unable to resolve it.



This error may occur (rarely) after processing has finished. You can safely ignore it by pressing "OK".

## GRCTools 5 Manual

1. Title.....	1
2. End-User Licence Agreement & Disclaimer.....	3
3. About GRC Tools.....	4
4. About GRC Country.....	6
5. Technical Requirements.....	7
6. Installation.....	8
7. Summary of the modules.....	9
8. Running GRC Country.....	15
8.1. GRC Country - The menu.....	16
8.2. GRC Country - Processing a file.....	17
9. Running GRC Tools.....	27
9.1. GRC Tools - The menu.....	29
9.2. GRC Tools - Processing a file.....	32
9.3. The results file.....	42
10. The Processes.....	43
10.1. Invalid strings - move field contents.....	49
10.2. Field - split string.....	50
10.3. Field - Trim leading spaces.....	52
10.4. Field - trim final string.....	53
10.5. Non-numeric characters - remove.....	54
10.6. Punctuation - remove.....	55
10.7. Double spaces - remove.....	56
10.8. Telephone numbers - burst.....	57
10.9. Postal codes / Postal codes (incl. format) - locate.....	58
10.10. Postal codes - write located.....	61
10.11. Postal code country code - remove.....	62
10.12. Thoroughfare types - parse.....	63
10.13. Postal code - format.....	65
10.14. Postal code - assess validity.....	66
10.15. Postal codes - write corrected.....	69
10.16. Language Regions - assign.....	70
10.17. Quotation marks - remove.....	71
10.18. Apostrophes (French) - add.....	72
10.19. Casing - to upper case.....	73
10.20. Casing - to mixed case / to mixed case - addresses / to mixed case - other.....	74
10.21. Articles - move or parse.....	76
10.22. "And" strings - standardise.....	77
10.23. Abbreviations / Abbreviations in personal names - standardise.....	78
10.24. Company types - standardise.....	79
10.25. Department strings - standardise.....	80
10.26. Sub-building information - parse & standardise.....	81
10.27. Post office box numbers - parse.....	82
10.28. Thoroughfare strings - standardise.....	84
10.29. Building numbers - move or parse.....	86
10.30. Building number suffix - parse.....	89
10.31. Commas - add or remove.....	90
10.32. Building number/letter - format.....	92
10.33. Sorting code - parse & standardise.....	94
10.34. Place names - parse & standardise.....	96
10.35. Place names - standardise.....	98
10.36. Provinces / Regions - parse / assign.....	100
10.37. University strings - parse.....	102
10.38. Building strings - move field contents.....	104
10.39. Attention strings - move field contents.....	106
10.40. Department strings - move field contents.....	107
10.41. Personal name prefixes - parse & standardise.....	108
10.42. Personal name suffixes - parse & standardise.....	109
10.43. Personal name middle name(s)/initial(s) - parse.....	110
10.44. Job titles - standardise.....	111
10.45. Accents - remove.....	112

11. Frequently asked questions.....	113
11.1. The different parts of my addresses in my database are all mixed up. How can GRC Tools help me to parse this data into different fields?.....	114
11.2. How can GRC Tools help to standardise and format my data to make it more suitable for mailing and to improve the performance of de-duplication and postal validation programs?.....	118
12. Technical Support.....	121
12.1. Frequently encountered problems.....	122