



---

unique: A Tool for Automatically Filtering for  
Unique Combinations of Conditions

CML00033-01

---

Code Magus Limited (England reg. no. 4024745)  
Number 6, 69 Woodstock Road  
Oxford, OX2 6EY, United Kingdom  
[www.codemagus.com](http://www.codemagus.com)  
Copyright © 2014 by Code Magus Limited  
All rights reserved



August 16, 2016

## **Contents**

<b>1 Introduction</b>	<b>2</b>
<b>2 Processing</b>	<b>2</b>
<b>3 Command line parameters</b>	<b>2</b>

## 1 Introduction

The `unique` program is a member of the `filetools` package. On OS/390, this program is known as `SRDFTUNQ`. The program can do simple domain frequency analyses and can perform a type of downsizing based on making certain fields, or combinations of fields unique in an output file by filtering the records from the given input file. The program works by supplying a copybook and a record in that copybook which is assumed to map the file. There can be any number of fields or groups of fields. Ideally a field or group of fields would be chosen because the cardinality of the column or column group is relatively small and the field or group of fields forms some sort of path or conditional indicator to the processing programs or system.

The program works on Unix and Unix-based platforms, Windows platforms as well as well as OS/390 Open MVS and Classic MVS environments. The program uses the `symbols [?]` library to process the copybooks. The `hashtab` hash-table library is used for internal hash-tables and the `rdwio` library is used for reading variable length image files on non-classic MVS environments. Parameter processing uses the `popt` library.

This tool is a prototype. The functionality of this tool will later be incorporated into a general data query tool and a generalised downsizing tool.

## 2 Processing

The input file is filtered to produce an output file by considering a number of fields or groups of fields. A record from the input file passes the filtering and is written to the output file if one of the fields or groups of fields within the file contains a value which has not been seen before. When considering a group fields, the group is assumed to make up a virtual field to which the same filtering rule applies.

## 3 Command line parameters

All processing is driven from the command line parameters. From the point of view of Windows-based and Unix-based systems this implies that the program is launched from a DOS or Unix shell. For classic MVS, the program can be run directly by naming the program on the `PGM JCL` parameter and supplying the command line arguments in the `PARM JCL` parameter. Alternatively, the program may be run under the TSO/E `TMP IKJECT01`, and in which case, the parameters are simply command parameters. In both these latter two cases the program name to use is `SRDFTUNQ`.

```
//C061379P JOB (#####), 'NAME', CLASS=A, MSGCLASS=X, NOTIFY=C061379,
```

```
// REGION=128M
//UNIQUE EXEC PGM=IKJEFT01
//STEPLIB DD DSN=QSCT00.SRDA1.LOADLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//LIB DD DSN=QSCT00.SRDA1.COPY(STATRUPY),DISP=SHR
//IN DD DSN=QSGG00.SWISS.DELETEME.D0.SQ231,DISP=SHR
//SYSTSIN DD *
SRDFTUNQ +
-v -d -m dd:LIB -i dd:IN -o dd:OUT -l 231 +
-r rb,type=record +
-w wb,type=record +
-n OUT-DATA-RECORD +
KEY-DATA
//OUT DD DSN=C061379.TMP.STATRUP,DISP=SHR,
// DCB=QSGG00.SWISS.DELETEME.D0.SQ231
//
//
```

The general format of running the program is:

```
unique [options] <column-group> ...
```

Where

```
<column-group> ::= <field-name>[:<field-name> ...]
```

That is, a column group is either a single field name or a list of field names separated with the colon character (without spaces). The singleton fields or the groups of fields themselves are separated from each other as separate command line parameters with at least a single space between the groups.

- `-m, --map-file=<map file>` Name of file which contains the layout map.
- `-i, --in-file=<in file>` Name of data file to process.
- `-r, --in-mode=r|<in mode>` Open mode string used when opening input file.
- `-o, --out-file=<out file>` Name of data file to process.
- `-w, --out-mode={w|<out mode>}` Open mode string used when opening output file.
- `-n, --map-name=<map name>` Name of the structure in the map file to use.
- `-c, --charset={ascii|ebcdic}` Default character set for character data.
- `-e, --endian={big|little}` Default binary data endian.

- `-h, --hash-mod={default | <integer>}` Hash table modulo value.
- `-l, --record-length={default | <integer>}` Input and out file record length.
- `-d, --list-domains` List the values of the column domains.
- `-r, --use-rdwio` Use the RDW I/O library for reading file.
- `-a, --var-records` Records are variable length records using OS/390 stdio.
- `-k, --key-count={1 | <integer>}` Maximum records per key group value.
- `-v, --verbose` Verbose printing during processing.
- `-?, --help` Show this help message.
- `--usage` Display brief usage message.