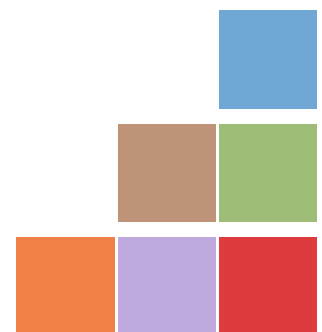# COBOL-IT® Compiler Suite Enterprise Edition

Introduction to DISAM & Migration Guidelines

Version 4.2

Contents

Acknowledgment

This documentation is derived from COBOL-IT Source code, parts of which are derived from OpenCOBOL.

Copyright (C) 2002-2007 Keisuke Nishida
Copyright (C) 2007 Roger While
Copyright (C) 2008-2020 COBOL-IT

In 2008, COBOL-IT forked its own compiler branch, with the intention of developing a fully featured product and offering professional support to the COBOL user industry.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Third-Party software components embedded in the SOFTWARE and Services and submitted to specific licenses:

VBISAM

* Copyright (C) 2003 Trevor van Bremen

* Copyright (C) 2008-2020 COBOL-IT

* License: LGPL

GMP (GNU Multiprecision Library)

* Copyright 1991, 1996, 1999, 2000, 2007 Free Software Foundation, Inc.

* License: LGPL

GNU LIBICONV

The libiconv libraries and their header files are under LGPL.

Microsoft and Windows are registered trademarks of the Microsoft Corporation. UNIX is a registered trademark of the Open Group in the United States and other countries. Other brand and

product names are trademarks or registered trademarks of the holders of those trademarks.

Contact Information:

The Lawn
22-30 Old Bath Road
Newbury, Berkshire, RG14 1QN
United Kingdom
Tel: +44-0-1635-565-200

Compiler, Runtime, and Debugger Topics

Introduction

This document describes how to install and how to use the **COBOL-IT Compiler Suite.**

This file contains part of the initial OpenCOBOL manual.
Copyright (C) 2002-2007 Keisuke Nishida
Copyright (C) 2007 Roger While
Copyright (C) 2008-2020 COBOL-IT

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

COBOL-IT License terms

COBOL-IT Compiler Suite

`cobc`, `cobcrun` and `cobcdb`, Copyright (C) 2008-2020 COBOL-IT

The executable components in the COBOL-IT Compiler Suite, `cobc` and `cobcrun` are based on

OpenCOBOL, originally developed by Keisuke Nishida and maintained since 2007 by Roger While.

Copyright (C) 2002-2007 Keisuke Nishida
Copyright (C) 2007 Roger While

COBOL-IT forked its own compiler branch (`cobc` and `cobcrun`) in 2008 to develop a fully-featured product and offer professional support to the COBOL user industry.

`cobcdb`, COBOL-IT® Debugger System (cobcdb®), Copyright © 2008-2020 COBOL-IT S.A.R.L. All rights reserved. You shall not duplicate or transfer this SOFTWARE, in whole or in part, in whatever media or manner, for any purpose, without COBOL-IT's prior written approval.

COBOL-IT Runtime System

`libcobit`, Copyright (C) 2008-2020 COBOL-IT

The executable component in the COBOL-IT runtime system, `libcobit` is based on the `libcob` library originally developed by Keisuke Nishida and maintained since 2007 by Roger While.

Copyright (C) 2002-2007 Keisuke Nishida
Copyright (C) 2007 Roger While

For more information, please contact us at: contact@cobol-it.com

COBOL-IT Corporate Headquarters are located at:
The Lawn
22-30 Old Bath Road
Newbury, Berkshire, RG14 1QN
United Kingdom
Tel: +44-0-1635-565-200

COBOL-IT, COBOL-IT Compiler Suite, CitSQL, CitSORT, and COBOL-IT Developer Studio are trademarks or registered trademarks of COBOL-IT.
Eclipse is a trademark of the Eclipse Foundation.
IBM, CICS, DB2, and AIX are registered trademarks of International Business Machines Corporation.
Linux is a registered trademark of Linus Torvalds.
Oracle, Pro*COBOL, Tuxedo and MySQL are registered trademarks of Oracle Corporation.
Postgres is a registered trademark of PostgreSQL Global Development Group
Syncsort is a registered trademark of Syncsort, Inc.
SQL Server, Windows, Visual Studio, and Visual Studio Express are registered trademarks of Microsoft Corporation.
Java and Solaris are registered trademarks of Sun Microsystems, Inc.
UNIX is a registered trademark of The Open Group
HP is a registered trademark of Hewlett Packard, Inc.
Red Hat is a registered trademark of Red Hat, Inc.
Micro Focus is a registered trademark of Micro Focus (IP) Limited in the United Kingdom, the United States, and other countries.

All other trademarks are the property of their respective owners.

Introduction to DISAM & Migration Guidelines

Overview

With COBOL-IT Version 4.0, The VBISAM engine has been deprecated and replaced by the D-ISAM indexed file engine. The D-ISAM engine is more stable and more widely used than the VBISAM engine and is fully compatible with IBM C-ISAM 7.2. Unfortunately, VBISAM files are not readable by D-ISAM and will require conversion. Conversion can be done using CitSORT, or by writing a program that uses the COB_EXTFH environment variable to direct the runtime to use the VBISAM Extfh driver (on READs) or the D-ISAM Extfh driver (on WRITEs) in your conversion.

For Customers who have been using VBISAM files in their application, an upgrade to COBOL-IT Version 4.0 will require that all of the source code in their application be re-compiled, and that they consider the implications of operating in a runtime environment with a different default file system.

Introduction to DISAM, including Migration Guidelines is intended to provide users with guidelines on how to manage their transition from COBOL-IT Version 3.X or earlier to COBOL-IT Version 4.x and greater.

DISAM provides higher performance than VBISAM, and has more robust handling of record locking in multi-user situations. DISAM also provides a richer set of file utilities.

COBOL-IT produces the same file status codes for VBISAM, CISAM, DISAM, and BerkeleyDB ndexed files.

For details, see the COBOL-IT Compiler & Runtime Reference Manual chapter on File Status Codes.

For information on how to map COBOL-IT file status codes to custom file status codes, see the fstatus-map compiler flag. The -fstatus-map compiler flag may be repeated as many times as necessary for cases where more than one file status code needs to be re-mapped.

Conversion Guidelines

Overview
There are a number of different cases that we will review. These are:
- Customer has a large store of VBISAM files and wishes to convert these files to DISAM as part of their pre-production work. Moving forward, these Customers will have all of their indexed file stored in DISAM files. Converting files from VBISAM to DISAM can be managed either using CITSORT or programmatically. When converting programmatically, you must recognize the restriction that the COB_EXTFH environment variable can be set dynamically at the beginning of a program, but cannot be re-set during the program. As a

result, best practice is to structure the process in two parts; first read your VBISAM files and write them out to a sequential file. Then, in a separate program, read your sequential files and write them into DISAM files.

- Customer has a large store of VBISAM files, but does not wish to convert them to DISAM. Customer wishes to store new data in DISAM files (the default), while maintaining existing data in VBISAM files. Care needs to be taken to modify the COB_EXTFH environment variable sot that it is set to vbisamextfh, when directing the runtime to use VBISAM files within the command shell.

Our examples
In this examples, our indexed file is called holidaysIX. The record length is fixed at 73 bytes. The primary key starts on the fourth character position, and is 25 bytes long. As a consequence, you will see the ISAM files in our scripts described as follows:
org ix
key (4,25,P)
record F 73

With regards to positioning, when describing the key(s) in your indexed file, it is important to note that the first byte of the file is described with position 1. This is an important note because the DISAM utility dcheck sets the first byte of the file with position 0 when describing the key(s) in the indexed file.

Using COB_EXTFH to manage your conversion

When CITSORT opens an indexed file, it follows the following rules, when determining the file format:
- If COB_EXTFH is not set, then DISAM is assumed as the default.
- For other file formats, COB_EXTFH should be set as follows:

File System	COB_EXTFH environment var
BerkeleyDB	>export COB_EXTFH=bdbextfh (UNIX/Linux)
D-ISAM	>export COB_EXTFH=disamextfh (UNIX/Linux)
C-Tree	>export COB_EXTFH=ctextfh (UNIX/Linux)
VBISAM	>export COB_EXTFH=vbisamextfh (UNIX/Linux)

As a consequence, a general guideline to follow for converting files using CITSORT is to set COB_EXTFH to the file format of the indexed file, dump the file into a line sequential file, set COB_EXTFH to the file format of the indexed file system to which you are converting, and then load the line sequential file into the indexed file.

We have provided examples below, in which we provide sample data in line sequential format, and parameter files for loading data into DISAM and VBISAM files. These samples are intended to inform the important examples, which include scripts that can be used to use CITSORT to convert data from VBISAM

to DISAM, and from DISAM to VBISAM. Variations of these scripts could be used to convert data between any of the file systems supported by COBOL-IT>

Data used in our examples

holidaysIX.txt

```
013ALL Saints Day        WednesdayNovember 012017201812021348l737+0000
017Christmas             Monday   December 252017201812021348l737+0000
011Columbus Day          Monday   October  092017201812021348l737+0000
018Day-After Christmas   Tuesday  December 262017201812021348l737+0000
016Day-After Thanksgiving Friday  November 242017201812021348l737+0000
008Fathers Day           Sunday   June     182017201812021348l737+0000
012Halloween             Tuesday  October  312017201812021348l737+0000
009Independence Day      Tuesday  July     042017201812021348l737+0000
010Labor Day             Monday   September042017201812021348l737+0000
002Martin Luther King Day Monday  January  162017201812021348l737+0000
007Memorial Day          Monday   May      292017201812021348l737+0000
006Mothers Day           Sunday   May      142017201812021348l737+0000
019New Year's Eve        Sunday   December 312017201812021348l737+0000
001New Years Day         Sunday   January  012017201812021348l737+0000
004Presidents Day        Monday   February 202017201812021348l737+0000
005St Patrick's Day      Friday   March    172017201812021348l737+0000
015Thanksgiving          Thursday November 232017201812021348l737+0000
003Valentines Day        Tuesday  February 142017201812021348l737+0000
014Veterans Day          Saturday November 112017201812021348l737+0000
```

Converting files using CITSORT

Overview
        The process for converting files from VBISAM to DISAM using CITSORT follows this
basic process:

1- Dump your VBISAM file into a line sequential file
2- Load your line sequential file into a DISAM file

These two important functions can be handled most efficiently using CITSORT's ability to take a
parameter file:
For example:

```
>citsort take params1.txt
```

In addition to your parameter files, you must be careful when constructing your conversion shell
scripts to make sure that environment variables are set correctly for DEFAULT_CITDIR,
COBOLITDIR, and COB_EXTFH.

The scripts provided presume that you have installed COBOL-IT Version 4.0, and that
COB_EXTFH is set to DISAM by default, and must be set, for the initial data dump, to
vbisamextfh.

You will recognize this general process:

```
[ Set environment ]
[ run citsort take params1.txt to dump the VBISAM file to a sequential file ]
[ Set environment ]
[ run citsort take params2.txt to load the sequential file into a DISAM file.
```

params1.txt – Dumps indexed file to line sequential file

```
copy
use holidaysIX
org ix
key=(4,25,P)
record F 73
give holidaysIX.txt
org ls
record F 73
```

params2.txt - Loads line sequential file to indexed file

```
use holidaysIX.txt
org ls
record F 73
give holidaysIX
org ix
key (4,25,P)
```

```
record F 73
```

Loading a sequential file into a DISAM file

```
#loaddisam.sh
unset DEFAULT_CITDIR
unset COBOLITDIR
unset COB_EXTFH
source /opt/cobol-it4-64/bin/cobol-it-setup.sh
# params2.txt loads a line sequential file into an ISAM file.
# COB_EXTFH is unset so a DISAM chmofile will be created
citsort take params2.txt
dcheck holidaysIX
```

Using CITSORT to convert VBISAM to DISAM

```
# cat vbisam2disam.sh
export DEFAULT_CITDIR=/opt/cobol-it4-64
export COBOLITDIR=$DEFAULT_CITDIR
unset COB_EXTFH
source /opt/cobol-it4-64/bin/cobol-it-setup.sh
export COB_EXTFH=vbisamextfh
citsort take params1.txt
unset COB_EXTFH
citsort take params2.txt
```

Using CITSORT to convert DISAM to VBISAM

```
# cat disam2vbisam.sh
unset DEFAULT_CITDIR
unset COBOLITDIR
unset COB_EXTFH
source /opt/cobol-it4-64/bin/cobol-it-setup.sh
#params1.txt dumps a DISAM file into a line sequential file
citsort take params1.txt
# load your line sequential file into a VBISAM file
# first, set COB_EXTFH so VBISAM is used by default
export COB_EXTFH=vbisamextfh
# params2.txt loads a line seque3ntial file into an ISAM file.
# COB_EXTFH is set to vbisamextfh so a vbisam file will be created
citsort take params2.txt
```

Continuing to use VBISAM files

To continue running applications with VBISAM files, set the COB_EXTFH environment variable
to vbisamextfh prior to launching your application, as follows:
>export COB_EXTFH=vbisamextfh (Linux)
>SET COB_EXTFH=vbisamextfh (Windows)

Note that you may not re-set COB_EXTFH after your application has been launched.

Converting files programmatically

The COBOL-IT runtime checks for the setting of COB_EXTFH when the runtime is launched. As a result, when you launch cobcrun, you must consider that the setting of COB_EXTFH (DISAM by default) will be used throughout the runtime session. Converting files programmatically can be accomplished by creating two consecutive runtime sessions, the first which sets COB_EXTFH to **vbisamextfh**, READs the VBISAM file and WRITEs the data out as sequential file, and the second which sets COB_EXTFH to **disamextfh**, READs the sequential file, and WRITEs out the DISAM file. Alternatively, in your second runtime session, you can just unset COB_EXTFH. In the absence of any setting of COB_EXTFH, COBOL-IT Compiler Suite Version 4.0 automatically loads the DISAM extfh library.

The DISAM File Utilities

dcheck

dcheck – describe, check and maintain isam files
usage – dcheck [-bfhi] <isamfile> […]
The option string, preceded by a dash, can be placed anywhere on the command line. all options must be specified in one string, and all options apply to all files specified.
dcheck is not interactive, and will ask no questions, so it can be safely used in batch and script files without operator intervention.
dcheck has been designed to run co-operatively (unless -f or -b specified) on files in active use, but note that other processes will be blocked for the duration of the check cycle.
options –
b – rebuild all indexes regardless of errors
>dcheck -b customerfile
f – fix (rebuild) corrupt indexes
>dcheck -f customerfile

h – display isam header information only
>dcheck -h customerfile

i – just check indexes, ignore data file
>dcheck -i customerfile

dpack

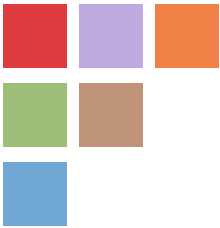**Rebuilding and reclaiming space using Disam's dpack**
dpack takes a list of one or more isam files and will rebuild each in turn. it works by building a

temporary copy of the original file and copying all active data records to it, then deleting the original and renaming the working copy. The temporary file is created in the same directory as the original and is named dpnnnnn, where nnnnn is the current process id.
You must ensure that you have write permission on the data directory and that there is enough room for dpack to build the working copy. dpack will fail if it cannot obtain an exclusive lock on the file.
Dpack can be used to refine the rebuilding process, and to reclaim space by removing artifacts associated with deleted records, and to pack both the .dat and the .idx files.

In DISAM, a deleted record is marked for deletion in the .dat and the corresponding index value is removed from the index node. An indexed node from which all the key values have been deleted is an empty indexed mode and is removed from the .idx file.

dpack - isam file pack/rebuild utility
usage - dpack [options] <isamfile> [...]
 -B<recnum> - begin at (one based) recnum
-C<count> - continue for count records
-u - undelete mode - restore deleted records
>dpack -B5 holidaysIX
>dpack -B5 -C10 holidaysIX
>dpack -u holidaysIX

www.cobol-it.com

June, 2020