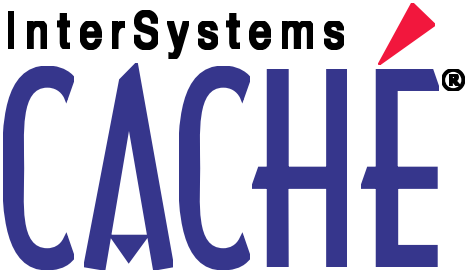

Using cvendian to Convert Between Big-endian and Little-endian Systems

Version 2009.1
30 June 2009

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Using cvendian to Convert Between Big-endian and Little-endian Systems
Caché Version 2009.1 30 June 2009

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at www.w3c.org. The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.

InterSystems CACHÉ® and InterSystems ENSEMBLE®

Caché WEBLINK, Distributed Cache Protocol, M/SQL, M/NET, and M/PACT are registered trademarks of InterSystems Corporation.

INTERSYSTEMS, InterSystems HEALTHSHARE™, InterSystems DEEPSEE™ and InterSystems TRAKCARE™

InterSystems Jalapeño Technology, Enterprise Cache Protocol, ECP, and InterSystems Zen are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Customer Support
Tel: +1 617 621-0700
Fax: +1 617 374-9391
Email: support@InterSystems.com

Table of Contents

Using cvendian to Convert Between Big-endian and Little-endian Systems

Caché provides a utility to convert the byte order of a Caché database from Big-endian (that is, most-significant byte first) to Little-endian (that is, least-significant byte first), and vice versa. It is called **cvendian**, for *con*vert *endian*. This is useful when moving a database among platforms of the two types. It also provides an option to report on the byte order of a given database.

1 Platforms and Location of Utility

On Windows and UNIX systems, the **cvendian** utility is the file <cache-install-dir>\Bin\cvendian.exe. On OpenVMS systems, this utility is not provided. For information about the Endianness of supported platforms, see "Platform Endianness" in *Supported Platforms*.

2 Conversion Process

You can run **cvendian** on either the system that has the files to be converted or the system that will be using the converted files (unless one of those systems is running OpenVMS; see the preceding section).

For example, to convert a database from a Little-endian to a Big-endian system, you can perform the conversion on the Little-endian system and then transfer the database to the Big-endian system, or you can transfer the file first, and then convert it.

Note: This utility does not work for backup and journal files. You must restore databases on a platform of the same endian, move the restored databases to the different endian platform, and then use the **cvendian** utility to convert the databases.

To convert a database, the process is:

1. Make a copy of your database files, because the utility replaces the source files with the converted files.
2. Run **cvendian** using the syntax described below, in Utility Syntax.

If you are using legacy multivolume databases, after converting a multivolume database file, use the **^LABEL** utility to rename the directory in each volume. Using the Caché Terminal, from the %SYS namespace, call **^LABEL** with the **Do** command; it prompts you for the proper input:

```
USER>zn "%SYS"

%SYS>Do ^LABEL


Enter the name of the directory in which the database is
stored. For a multi-volume database, enter the name of the
primary volume's directory, even if you want to relabel a
secondary volume. For a multi-volume legacy 2K database, you
should enter the name of the secondary volume directory if
you need to relabel it.

Directory:
```

3 Utility Syntax

With the **cvendian** endian utility, you can specify the desired byte order, or you can report the current byte order without conversion. Use the following syntax:

```
cvendian [-option] file1 [file2 ... file8]
```

The *option* argument is one of the following:

- `-big` — convert the database to Big-endian
- `-little` — convert the database to Little-endian
- `-report` — report the byte order of the database

You can shorten the options to their initial letter. If this is a conversion request (*-big* or *-little*), and the database already has the specified byte order, the utility displays a warning message and stops processing.

If you do not provide the *option* argument, the utility converts the database from the existing byte order to the other byte order. It is recommended, however, that you use the *option* argument.

The *file1* through *file8* arguments are the files to convert; each file can include a complete pathname. The *file2* through *file8* arguments are for a multivolume database; if you are converting a multivolume database, you must specify all the volumes on the command line, in order.

The utility performs the following actions:

- Auto-detects the byte order of the database
- Displays endian information and other information
- Performs the conversion
- Displays a message indicating success or failure

For multivolume databases, if the files are out of order or the list is not complete, the utility does not perform any conversions and leaves the files as they are.

3.1 Example

For example, suppose you are converting a database for use on Solaris SPARC from Windows XP. Because SPARC and Intel have incompatible data representations, you must convert from Little-endian (for Intel) to Big-endian (for SPARC). The output from running **cvendian** on the Windows system before moving the file to the Solaris system looks similar to this:

```
C:\CacheSys\Bin>cvendian -big c:\temp\solarisdb\cache.dat

This database is little-endian.
This database has a block size of 8192 bytes.

This database has 1 volume and 1 map.
The last block in the primary volume is 18176.

Original manager directory is c:\temp\solarisdb\

No extension volumes.

Done converting c:\temp\solarisdb\cache.dat to big-endian

C:\CacheSys\Bin>
```

You can now move the converted database file to the Solaris system.