

UniBasic ctool utility

10 July 1998

Dynamic Concepts Incorporated (DCI) has prepared this document for use by DCI personnel, licensees, and authorized representatives. The material contained herein shall not be reproduced in whole or in part without the prior written consent of DCI.

DCI reserves the right to make changes to the specifications and material contained herein without notice. Every attempt has been made to present this document in an accurate and complete form. DCI shall not be responsible for any damages (including, but not limited to consequential) caused by the use of or reliance upon the product(s) described herein.

This document is intended for users of *UniBasic*.

© Copyright 1998 Dynamic Concepts, Inc. (DCI). All rights reserved.

Dynamic Concepts Inc.
One Columbia, Suite 100
Aliso Viejo, CA 92656

(949) 448-8600 or (800) 822-2742

www.dynamic.com

About this Guide

This Guide supposes that you have a working knowledge of UniBasic. You should, for example, know how to run the **ubrebuild** and **ubcompress** utilities and your backup programs. A knowledge of the files and how logical units are organized on disk will also be helpful.

This Guide also assumes that you understand the fundamental operations of both UniBasic and Unix. Consult the [UniBasic Reference Guide](#) and your Unix manuals as needed. If you experience difficulty, please contact your dealer or distributor.

Typographical Conventions

This guide uses the following typographic conventions:

Example of convention	Description
GOSUB	Capitalized words in bold indicate language-specified reserved words.
KILL <i>filename</i>	Variables are shown in italic type for clarity and to distinguish them from elements of the language itself.
LIST	Mono-spaced type is used to display screen output and example input commands and program examples.
<letter>	Information inside angle brackets <> must be from specified group, e.g., a single letter.
WHILE UNTIL	A vertical bar indicates that the user must choose one of the items.
[<i>expr</i>]	Items inside square brackets are mandatory.
{ <i>expr</i> }	Items inside braces are optional.
stmt {\ stmt} ...	A series of three periods (...) indicates that the item preceding them can be repeated one or more times.

Introduction

This utility converts UniBasic Non-Portable files (except Formatted files) to Universal Data files. After conversion, files are platform independent and may be accessed by UniBasic version 6 and greater, dL4 version 3 and greater, IQ version 4 and greater, and IQ for Windows version 5 and greater.

The destination directory must be specified and be different than the source directory. If the destination directory does not exist, the utility will create it and the absolute path/filename of the converted file will be maintained under this directory. This allows the user to archive the directory structure and move it to a different location or machine as desired. For example:

```
ls -R | cpio -ocv >archivefile
```

executed from the destination directory creates the absolute path/filename structure of the converted files on the device or file 'archivefile'. This structure may then be moved and unarchived with the command:

```
cpio -imcdv <archivefile
```

maintaining the original source files directory structure at the new location.

The converted file is a mirror image of the source file, leaving the source file unmodified. A file is always written to the destination directory, even if there is a conversion problem.

The utility sends all messages to standard out. If it encounters any problems with a field it is converting, it reports the record number and field number of the problem area. This allows the user to examine the file to determine the problem and course of action. Numeric fields that cannot be converted for any reason, such as numeric overflow, are set to zero (0). String fields that cannot be converted are set to null.

If a file was created with an **ISAMSECT** value greater than 8 it cannot be converted. To access converted files with UniBasic 6.1 or greater the **ISAMSECT** environment variable must be set to at least 8.

It is recommended that you have a current backup of the file(s) to be converted before processing. Also, verify that the file(s) to be converted are currently not in use by someone else. In other words, no one should have the file(s) open.

The user should use the UniBasic utilities, **ubrebuild** and **ubcompress**, to insure the integrity of the source file(s).

NOTE: This utility is in development and has not been extensively tested. The user assumes all responsibility for its use.

Installation

Type `./ctool` to run it from the current directory, or place it in `/usr/bin` directory and set permissions to 555.

ctool

Synopsis

Convert UniBasic Non-Portable Files to Universal Files.

Syntax

```
$ ctool { -h | -? } { -v } { -r } { -k } { -l } -o dir filename
```

Parameters

-h -?	Output help
-v	Output version number
-r	Replace destination file(s) unconditionally
-k	Use random key insertion algorithm
-l	Use dL4 LUMAP (use discouraged)
-o <i>dir</i>	Build destination file(s) in <i>dir</i> (required)
<i>filename</i>	Conversion layout filename (required)

Remarks

ctool is a program that converts UniBasic Non-Portable file(s) to Universal file(s). It must be executed on the native platform of the file(s) to be converted. It is executed from the Unix command prompt.

A destination directory for the output file(s) must be specified with the -o *dir* argument on the command line. The utility will create this directory and any sub-directory structure specified in the FILE section of the conversion layout file if it does not exist.

The '-r' option specifies that any file(s) of the same name in the destination directory will be overwritten. This option overrides the REPLACEFILE label in the conversion layout file.

The '-k' option modifies the index portion of the file to use the random insertion algorithm for key insertions. This may result in a larger index file size.

The '-l' option specifies that the dL4 environment variable **LUMAP** be used to map logical paths to actual paths.

filename is the name of the conversion layout file. The conversion layout is a text file, created by the user, that defines each file to be converted..

The Conversion Layout Defined

The conversion layout file consist of sections and labels. Sections are denoted by a keyword enclosed in square brackets, for example: [FILE]. Labels are denoted by a keyword equal to a value, such as: FILE=test. Lines beginning with a semicolon (;) are comments and blank lines are allowed.

Two sections must be defined for each file to be converted. The [FILE] section must be defined first and contains the required label 'FILE=filename'. It may also contain optional labels 'REPLACEFILE=Yes|No' and 'RECORDORIGIN=0|1'. If not used, REPLACEFILE defaults to NO and RECORDORIGIN defaults to 0. For example:

```
[File]
File=filename
ReplaceFile=Yes
RecordOrigin=1
```

The value of the label FILE is the name of the file to be converted.

The value of the label REPLACEFILE may be either Yes or No and determines the action if a file of the same name already exists in the destination directory. It is similar to the '-r' command line option but controls replacement on a file by file basis. The '-r' option if used will override this specification.

The value of the label RECORDORIGIN allows the specification of byte positions, in the record definition section that follows, to begin at 0 or 1. This allows the user to think of the first byte of the record as byte 0 or byte 1.

The [RECORD] section is the second section of the conversion layout file and contains FIELD labels, one for each data field in the record. The FIELD label has the form 'FIELD=*parm1*,*parm2*,*parm3*{,*parm4*}'.

parm1 is an optional identifier for the user and may be omitted. If omitted a comma must precede *parm2*. The utility does not use this identifier, but rather uses a count of the FIELD labels. For example, if the utility reports a problem with FIELD 3 then this refers to the third field defined in the [RECORD] section.

parm2 specifies the field starting byte position in the record.

parm3 specifies the length of the field. For string or binary fields, *parm3* is the byte count. For numeric fields, *parm3* must be the precision. The precision is entered as the mapped precision.

parm4 is used for binary and array fields. If the field is binary, enter a B for *parm4*. If the field is a numeric array, enter the DIMed value of the array for *parm4*. For example:

```
[Record]
FIELD=Alpha1,1,24
FIELD=Alpha2,25,24
FIELD=Numeric,50,4%
FIELD=Binary,58,10
FIELD=Array,68,2%,10
```

Putting it all together, the following is an example of a conversion layout file for multiple files:

```
[File]
File=ub/cust.master
ReplaceFile=Yes
RecordOrigin=1

[Record]
FIELD=Name,1,24
FIELD=Addr1,25,24
Field=Addr2,49,10
Field=Zip,59,2%
Field=Binary,63,10,B
Field=L4YS,73,4%,4
.
.
.

[File]
File=ub/detail.file

[Record]
;Detail file
Field=RecNumber,0,1%
Field=OrderDate,2,5%
Field=PartNumber,6,7
```

MRT conversion layout

If converting an MRT file, it will be necessary to use a RECORDNUMBER or RECORDID label in the [RECORD] section. A RECORDNUMBER or RECORDID label, if used, must precede any FIELD label. A RECORDNUMBER label is used if the field layout is dependent upon the location of the record in the file. A RECORDID label is used if the field layout is determined by a field in the record.

The RECORDNUMBER= label is followed by comma delimited parameters that specify an optional name, a byte offset which is just a placeholder, a byte length which is just a placeholder, a record number that identifies the starting record number for the following field definitions, and an optional ending record number that specifies an inclusive range for the field definitions to follow.

The RECORDID= label is followed by comma delimited parameters that specify an optional name, the starting byte offset of the field that identifies the record type, the byte length of the field that identifies the record type if the identifier is a string or the precision of the field that identifies the record type if the identifier is numeric, and the value of the record type identifier for the field definitions that follow.

General Remarks

The conversion maintains the record length and any bytes not defined in the conversion layout file will be set to nulls. Deleted records have all their fields set to null.

Example

```
$ ctool -h
$ ctool -o /tmp /ub/filedefs
$ ctool -r -l -o /newfiles filedefs
```

Errors

```
Unable to open file 'testfile': No such file or directory
Unable to open file 'testfile': Permission denied
```