

HP OpenVMS

HP C 国際化ユーティリティ・ リファレンス・マニュアル

AA-R1E9C-TE-2

2007年12月

本書では、HP C XPG4 ランタイム・ライブラリ・ユーティリティについて説明します。HP C ランタイム・ライブラリは、ローカライゼーションおよびタイム・ゾーン・データを管理する各国対応のソフトウェア・アプリケーションを開発する時に使用します。

改訂 / 更新情報:

日本語 OpenVMS V7.3 『Compaq C 国際化ユーティリティ・リファレンス・マニュアル』の改訂版です。

ソフトウェア・バージョン:

日本語 HP OpenVMS V8.3-1H1 for Integrity Servers

日本語 HP OpenVMS Alpha V8.3

日本語 HP OpenVMS VAX V7.3

© 2007 Hewlett-Packard Development Company, L.P.

本書の著作権は Hewlett-Packard Development Company, L.P. が保有しており、本書中の解説および図、表は Hewlett-Packard Development Company, L.P. の文書による許可なしに、その全体または一部を、いかなる場合にも再版あるいは複製することを禁じます。

また、本書に記載されている事項は、予告なく変更されることがありますので、あらかじめご承知おきください。万一、本書の記述に誤りがあった場合でも、日本ヒューレット・パッカーは一切その責任を負いかねます。

本書で解説するソフトウェア(対象ソフトウェア)は、所定のライセンス契約が締結された場合に限り、その使用あるいは複製が許可されます。

日本ヒューレット・パッカーは、弊社または弊社の指定する会社から納入された機器以外の機器で対象ソフトウェアを使用した場合、その性能あるいは信頼性について一切責任を負いかねます。

POSTSCRIPT は米国 Adobe Systems 社の登録商標です。

X/Open は米国 X/Open 社の商標です。

その他のすべての商標および登録商標は、それぞれの所有者が保有しています。

原典：『Compaq C Run-Time Library Utilities Reference Manual』

本書は、日本語 VAX DOCUMENT V 2.1を用いて作成しています。

目次

まえがき	v
1 概要	
1.1 XPG4 に準拠したローカライズ・アプリケーションの作成	1-1
1.1.1 メッセージ・カタログの作成と起動	1-1
1.1.1.1 メッセージ・ソース・ファイル	1-2
1.1.1.2 メッセージ・カタログ・ファイル	1-2
1.1.1.3 メッセージ・カタログからのメッセージの検索	1-2
1.1.2 コードセット変換の実行	1-2
1.1.2.1 変換テーブルの作成	1-3
1.1.2.2 1つのコードセットから別のコードセットへの変換	1-3
1.1.3 国際化環境論理名の設定	1-3
1.2 タイム・ゾーン変換情報の作成	1-5
1.2.1 ルール行	1-5
1.2.2 ゾーン行	1-7
1.2.3 リンク行	1-8
2 ロケール・ファイルの形式	
2.1 ロケール・カテゴリ	2-1
2.1.1 デフォルトの設定の変更	2-1
2.1.2 カテゴリ・ソース定義	2-2
2.2 LC_COLLATE カテゴリ	2-3
2.2.1 collating-element 文	2-4
2.2.2 collating-symbol 文	2-4
2.2.3 order_start 文	2-5
2.3 LC_CTYPE カテゴリ	2-7
2.4 LC_MESSAGES カテゴリ	2-11
2.5 LC_MONETARY カテゴリ	2-12
2.5.1 LC_MONETARY キーワード	2-12
2.5.2 各種の通貨編集形式	2-15
2.6 LC_NUMERIC カテゴリ	2-16
2.7 LC_TIME カテゴリ	2-17
2.7.1 キーワード	2-18
2.7.2 フィールド記述子	2-20
2.7.3 ロケール定義のサンプル	2-22

3	文字セット記述 (Charmap) ファイル	
3.1	Portable Character Set	3-1
3.2	charmap ファイルの構成要素	3-4
4	コマンド・リファレンス	
	GENCAT	4-2
	ICONV COMPILE	4-7
	ICONV CONVERT	4-13
	LOCALE COMPILE	4-15
	LOCALE LOAD	4-19
	LOCALE UNLOAD	4-21
	LOCALE SHOW CHARACTER_DEFINITIONS	4-23
	LOCALE SHOW CURRENT	4-25
	LOCALE SHOW PUBLIC	4-28
	LOCALE SHOW VALUE	4-30
	zic	4-34
A	付録: 日本語ロケールについて	
A.1	日本語ロケール・ファイル形式	A-1
A.2	ICONV CONVERT ユーティリティ補足説明 - 日本語文字コード変換テーブル・ ファイル	A-2

索引

表

1-1	ルールが有効になる日	1-6
1-2	ルールが有効になる時刻	1-6
2-1	LC_COLLATE カテゴリ・キーワード	2-4
2-2	LC_CTYPE カテゴリ・キーワード	2-8
2-3	LC_MESSAGES カテゴリ・キーワード	2-11
2-4	LC_MONETARY カテゴリ・キーワード	2-12
2-5	各種の通貨編集形式	2-15
2-6	LC_NUMERIC カテゴリ・キーワード	2-16
2-7	LC_TIME カテゴリ・キーワード	2-18
2-8	LC_TIME ロケール・フィールド記述子	2-20
3-1	Portable Character Set	3-1
4-1	特殊文字	4-4
4-2	コードセット宣言	4-9
4-3	ロケール・カテゴリ	4-25
4-4	ロケール・カテゴリとキーワード	4-30
A-1	日本語文字コードセット	A-1

まえがき

本書は、HP C ランタイム・ライブラリ・ユーティリティの使用法、およびリファレンス情報を記述しています。HP C ランタイム・ライブラリ・ユーティリティは、各国対応のソフトウェア・アプリケーションにおいてローカライゼーションやタイム・ゾーンのデータを管理します。

対象読者

本書は、HP C ランタイム・ライブラリを使用して、ローカライゼーションとタイム・ゾーン・データを管理するアプリケーションを開発するプログラマを対象にしています。

本書の構成

本書は、次の章から構成されています。

- 第 1 章では、GENGAT, ICONV, LOCALE, および ZIC プログラミング・ユーティリティについて説明します。
- 第 2 章では、ロケール定義ソース・ファイル、および HP がサポートする標準ロケール・カテゴリについて説明します。
- 第 3 章では、文字セット記述 (charmap) ファイルについて説明します。このファイルは、文字シンボルのエンコーディングを定義します。
- 第 4 章では、XPG4 ユーティリティおよび ZIC ユーティリティのコマンドについて詳しく説明します。
- 付録 A では、日本語ロケールについて説明します。

関連資料

HP C ランタイム・ライブラリ・ユーティリティについての詳しい説明は、次のマニュアルを参照してください。

- 『HP C ランタイム・ライブラリ・リファレンス・マニュアル』
- 『OpenVMS バージョン 7.0 新機能説明書』

HP OpenVMS 製品およびサービスについての詳細は、HP のウェブ・サイトをご覧ください。

<http://www.hp.com/jp/openvms/>

または

<http://www.hp.com/go/openvms/>

本書で使用する表記法

本書では次の表記法を使用しています。

表記法	意味
Ctrl/x	Ctrl/x という表記は、Ctrl キーを押しながら別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
PF1 x	PF1 x という表記は、PF1 に定義されたキーを押してから、別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
Return	例の中で、キー名が四角で囲まれている場合には、キーボード上でそのキーを押すことを示します。テキストの中では、キー名は四角で囲まれていません。 HTML 形式のドキュメントでは、キー名は四角ではなく、括弧で囲まれています。
...	例の中の水平方向の反復記号は、次のいずれかを示します。 <ul style="list-style-type: none">• 文中のオプションの引数が省略されている。• 前出の 1 つまたは複数の項目を繰り返すことができる。• パラメータや値などの情報をさらに入力できる。
.	垂直方向の反復記号は、コードの例やコマンド形式の中の項目が省略されていることを示します。このように項目が省略されるのは、その項目が説明している内容にとって重要ではないからです。
()	コマンドの形式の説明において、括弧は、複数のオプションを選択した場合に、選択したオプションを括弧で囲まなければならないことを示しています。
[]	コマンドの形式の説明において、大括弧で囲まれた要素は任意のオプションです。オプションをすべて選択しても、いずれか 1 つを選択しても、あるいは 1 つも選択しなくても構いません。ただし、OpenVMS ファイル指定のディレクトリ名の構文や、割り当て文の部分文字列指定の構文の中では、大括弧に囲まれた要素は省略できません。
[]	コマンド形式の説明では、括弧内の要素を分けている垂直棒線はオプションを 1 つまたは複数選択するか、または何も選択しないことを意味します。
{ }	コマンドの形式の説明において、中括弧で囲まれた要素は必須オプションです。いずれか 1 つのオプションを指定しなければなりません。
太字	太字のテキストは、新しい用語、引数、属性、条件を示しています。
<i>italic text</i>	イタリック体のテキストは、重要な情報を示します。また、システム・メッセージ (たとえば内部エラー <i>number</i>)、コマンド・ライン (たとえば <i>PRODUCER=name</i>)、コマンド・パラメータ (たとえば <i>device-name</i>) などの変数を示す場合にも使用されます。

表記法	意味
UPPERCASE TEXT	英大文字のテキストは、コマンド、ルーチン名、ファイル名、ファイル保護コード名、システム特権の短縮形を示します。
Monospace type	モノスペース・タイプの文字は、コード例および会話型の画面表示を示します。 C プログラミング言語では、テキスト中のモノスペース・タイプの文字は、キーワード、別々にコンパイルされた外部関数およびファイルの名前、構文の要約、または例に示される変数または識別子への参照などを示します。
—	コマンド形式の記述の最後、コマンド・ライン、コード・ラインにおいて、ハイフンは、要求に対する引数とその後の行に続くことを示します。
数字	特に明記しない限り、本文中の数字はすべて 10 進数です。10 進数以外 (2 進数, 8 進数, 16 進数) は、その旨を明記してあります。

HP C ランタイム・ライブラリ・ユーティリティは、各国対応ソフトウェア・アプリケーションのローカライゼーション・データ、および時刻データを管理するのに役立ちます。ローカライゼーション・データはアプリケーションとは別に定義され、実行時になってはじめて、アプリケーションに利用されます。

HP C ランタイム・ライブラリには、次のユーティリティが登録されています。

- XPG4 に準拠したユーティリティ (第 1.1 節)
- ZIC ユーティリティ (第 1.2 節)

1.1 XPG4 に準拠したローカライズ・アプリケーションの作成

国際的に使用できるアプリケーションを開発するのに役立つように、HP OpenVMS オペレーティング・システムでは、HP C ランタイム・ライブラリの一部として、XPG4 国際化モデル (X/Open Portability Guide Issue 4) をサポートするユーティリティを提供しています。XPG4 に準拠する次のユーティリティが提供されます。

- GENCAT ユーティリティ (第 1.1.1 項)
- ICONV ユーティリティ (第 1.1.2 項)
- LOCALE ユーティリティ (第 1.1.3 項)

これらのツールは、XPG4 モデルを対象に作成されたアプリケーションでのみ使用できます。

1.1.1 メッセージ・カタログの作成と起動

メッセージ・カタログとは、アプリケーションが表示または書き出すメッセージを含むバイナリ・ファイルです。このファイルには、アプリケーションが出力するすべてのメッセージ、たとえばエラー・メッセージ、情報メッセージ、画面メッセージ、プロンプトが格納されます。メッセージ・カタログを作成するには、GENCAT コマンドを使用します。

GENCAT は 1 つ以上の入力ソース・ファイルを読み込み、カタログ・ファイルが存在するときは、そのカタログ・ファイルも読み込みます。ソース・ファイルは、プログラムで出力するメッセージを格納するために作成するテキスト・ファイルです。ソース・ファイルにメッセージを入力するときは、任意のテキスト・エディタを使用で

きます。複数のソース・ファイルを指定した場合には、GENCAT は指定された順に 1 つずつソース・ファイル进行处理します。各ソース・ファイルがカタログを変更します。

開発者またはアプリケーションで GENCAT を実行する前に、必要な入力ソース・ファイルを作成し、必要に応じてカタログ・ファイルも作成しておかなければなりません。

GENCAT コマンドについての詳細は、第 4 章を参照してください。

1.1.1.1 メッセージ・ソース・ファイル

入力ソース・ファイルを作成する場合には、次のガイドラインに従ってください。

- メッセージはプログラムの機能に対応した集合に分類します。
- 各メッセージに数値の識別子を付けます。識別子は集合内で固有の値でなければなりません。
- メッセージの集合と個々のメッセージ进行处理するために、GENCAT が認識するコマンドを追加します。

1.1.1.2 メッセージ・カタログ・ファイル

catfile という名前のメッセージ・カタログが存在する場合には、GENCAT は旧バージョンの内容を格納したファイルの新しいバージョンを作成し、その新しいバージョンを変更します。カタログが存在しない場合には、GENCAT は catfile という名前のファイルを作成します。

1.1.1.3 メッセージ・カタログからのメッセージの検索

OpenVMS アプリケーションでは、次の HP C ランタイム・ライブラリ・ルーチンを使用して、メッセージ・カタログからメッセージを検索します。

- catopen
- catgets
- catclose

詳細は『HP C ランタイム・ライブラリ・リファレンス・マニュアル』を参照してください。

1.1.2 コードセット変換の実行

ICONV ユーティリティは変換テーブル・ファイルを作成し、また、このファイルを使用して文字を 1 つのコードセットから別のコードセットに変換するコマンドを提供します。

- ICONV COMPILE コマンドは変換テーブル・ファイルを作成します。
- ICONV CONVERT コマンドはこのファイルを使用して、指定されたファイルの文字を 1 つのコードセットから別のコードセットに変換します。

ICONV コマンドは、状態に依存しない 1 ~ 4 バイトのコードセットをサポートします。

注意

このインプリメンテーションでは、変換先のエンコーディングに制限があります。変換先内の文字の 4 バイト目で 0XFF を使用することはできません。

1.1.2.1 変換テーブルの作成

変換テーブル・ファイルを作成するには、DCL コマンドの ICONV COMPILE を使用します。

```
ICONV COMPILE sourcefile tablefile
```

変換ソース・ファイルのフォーマットについては、第 4 章の ICONV COMPILE コマンドの説明を参照してください。

tablefile 命名の規則については、第 4 章の ICONV CONVERT コマンドの説明を参照してください。

1.1.2.2 1 つのコードセットから別のコードセットへの変換

ファイル内の文字を 1 つのコードセットから別のコードセットに変換するには、ICONV CONVERT コマンドを使用します。

```
ICONV CONVERT infile outfile /FROMCODE=fromcodeset /TOCODE=tocodeset
```

変換された文字は、出力ファイル・パラメータ outfile に書き込まれます。

1.1.3 国際化環境論理名の設定

LOCALE ユーティリティは、次の操作を行う OpenVMS XPG4 ローカライゼーション・ユーティリティです。

- バイナリ・ロケール・ファイルを生成します。このファイルは、国際化環境論理名の設定に依存する各ユーティリティや C ルーチンが使用します。
- 共用の読み込み専用グローバル・データとして、システム・メモリにロケール名をロードします。
- 使用しているシステムで定義されている現在の国際化環境の概要、および使用しているシステムのロケールの詳細を表示します。
- システム・メモリからロケール名をアンロードします。

LOCALE ユーティリティは、次のコマンドをサポートしています。

- LOCALE COMPILE コマンドは、ロケール・ソース・ファイルをバイナリ・ロケール・ファイルに変換します。このファイルは、国際化環境論理名の設定に依存するユーティリティと C ルーチンで使用されます。このコマンドを使用すると、

(弊社が指定するローケルに加えて) 新しいローケルをシステムに追加することができます。

ローケルをコンパイルするために、LOCALE_COMPILE コマンドは 2 つのソース・ファイルを使用します。

- ローケルを記述するカテゴリを含む、ローケル定義ソース・ファイル表 4-3 に示すローケル・カテゴリには LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_MONETARY, LC_NUMERIC, LC_TIME があります。
- ローケルの文字セットを定義する charmap(character set description) ファイル。charmap はコード化文字セットのソース・ファイルであり、文字シンボルを文字コードとして定義します (詳細は第 3 章を参照してください)。
- LOCALE_LOAD コマンドは、共用の読み込み可能グローバル・セクションとして、システム・メモリにローケル名をロードします。ロードされたローケルにアクセスするすべてのプロセスは、この 1 つのローケルのコピーを使用します。したがって、全体として必要とされるシステム・メモリの容量を削減することができます。
- LOCALE_UNLOAD コマンドは、システム・メモリから指定したローケル名をアンロードします。
- LOCALE_SHOW_CHARACTER_DEFINITIONS コマンドは、論理名 SYS\$I18N_LOCALE で定義されるパブリック・ディレクトリ内の文字セット記述ファイル(charmap)の名前の一覧を表示します。charmap は、コーディングされた文字セットの文字のシンボル名と値を定義します。charmap ファイルのファイル・タイプは.CMAP です。
- LOCALE_SHOW_CURRENT コマンドは、各ローケル・カテゴリを示すいくつかの論理名により定義されている、現在の国際化環境の概要を表示します。このコマンドは各ローケル・カテゴリの設定と、環境変数 LC_ALL と LANG の値を表示します。カテゴリを定義する論理名には、カテゴリと同じ名前が付けられません。たとえば LC_MESSAGES という論理名は、LC_MESSAGES カテゴリの設定を定義します。
- LOCALE_SHOW_PUBLIC コマンドは、システム上のすべての公用ローケルをリストします。これには HP C ランタイム・ライブラリにより提供されるシステム・ローケル、および論理名 SYS\$I18N_LOCALE で定義されるディレクトリでリストされるローケルが含まれます。
- LOCALE_SHOW_VALUE コマンドは、現在の国際化環境での 1 つまたは複数のキーワードの値を表示します。各カテゴリのローケル・カテゴリおよびキーワードの一覧を表 4-4 に示します。

LOCALE コマンドについての詳細は第 4 章を参照してください。

1.2 タイム・ゾーン変換情報の作成

ZIC コンパイラは Zone Information Compiler(ZIC)ユーティリティを使用して、タイム・ゾーン変換情報を格納したバイナリ・ファイルを作成します。これらのファイルは、ユーザが指定したタイム・ゾーンのソース・ファイルから生成されます。

ソース・ファイルの行は複数のフィールドから構成されます。有効なタイム・ゾーンのソース・ファイルを作成するには、次の形式に従ってください。

- 各フィールドは任意の数のスペース文字で区切ります。
- 入力行の先頭と末尾のスペース文字は無視されます。
- 入力行で引用符で囲まれていないシャープ記号(#)はコメントを示し、この記号から行の末尾まではコメントになります。
- スペース文字とシャープ文字をフィールドの一部として使用するときには、二重引用符(" ")で囲みます。
- コメントを取り除いた後、何も無い行は無視されます。
- 空白行以外の行は次の3種類のいずれかであると解釈されます。
 - ルール行(第 1.2.1 項参照)
 - ゾーン行(第 1.2.2 項参照)
 - リンク行(第 1.2.3 項参照)

1.2.1 ルール行

ルール行の形式は次のとおりです。

```
Rule NAME FROM TO TYPE IN ON AT SAVE LETTER/S
```

次の例を参照してください。

```
Rule USA 1969 1973 - Apr lastSun 2:00 1:00 D
```

ルール行は次のフィールドで構成されます。

NAME

このルールが属しているルール集合の任意の名前を指定します。

FROM

ルールが適用される最初の年を指定します。minimum またはその短縮形は、表現可能な時刻値の年の最小値を表します。maximum またはその短縮形は、表現可能な時刻値の年の最大値を表します。

TO

ルールが適用される最後の年を指定します。FROM で定義した minimum と maximum の他に、FROM フィールドの値を繰り返し指定するために、minimum または only (またはその短縮形) が使用できます。

TYPE

ルールが適用される年のタイプを指定します。TYPE がハイフン- の場合は、FROM から TO までのすべての年にルールが適用されます。ZIC は年のタイプをチェックするために、次のコマンドを実行します。

```
yearistype year type
```

終了状態が 1 の場合は、年が指定されたタイプであることを示します。終了状態が 5 の場合は、年が指定されたタイプでないことを示します。

IN

ルールが有効になる月を指定します。月の名前は短縮できます。

ON

ルールが有効になる日を指定します。表 1-1 は認識される形式を示しています。

表 1-1 ルールが有効になる日

形式	意味
5	その月の 5 日
lastSun	その月の最後の日曜日
lastMon	その月の最後の月曜日
Sun>=8	8 日以降の最初の日曜日
Sun<=25	25 日以前の最後の日曜日

曜日名は短縮することも、つづりを完全に指定することもできます。ON フィールドの内部でスペース文字を使用することはできません。

AT

ルールが有効になる時刻を指定します。表 1-2 は認識される形式を示しています。

表 1-2 ルールが有効になる時刻

形式	意味
2	時
2:00	時と分
15:00	24 時間形式の時刻 (午後 12 時以降の時刻の場合)
1:28:14	時、分、秒

指定した時刻がローカル wall clock 時刻の場合は、これらの形式の後に w という文字を続けて指定することができ、時刻がローカル標準 (standard) 時の場合は s、時刻が UTC の場合は u (または g もしくは z)、という文字を続けることができます。文字が指定されていない時は、wall clock 時刻であると解釈されます。

SAVE

ルールが有効になるときに、ローカル標準時に加算する時間を指定します。このフィールドは AT フィールドと同じ形式ですが、もちろん、w や s という接尾語は使用しません。

LETTER/S

このルールが有効になるときに使用するタイム・ゾーンの短縮形の可変部を指定します。たとえばESTやEDTの場合はSまたはDです。このフィールドが- の場合には、可変部はヌルです。

1.2.2 ゾーン行

ゾーン行の形式は次のとおりです。

```
Zone NAME           GMTOFF  RULES/SAVE FORMAT UNTIL
```

次の例を参照してください。

```
Zone Australia/South-west 9:30  Aus           CST  1987 Mar 15 2:00
```

ゾーン行は次のフィールドで構成されます。

NAME

タイム・ゾーンの名前を指定します。この名前はタイム・ゾーンの時刻変換情報ファイルを作成するときに使用されます。

GMTOFF

このタイム・ゾーンの標準時刻を求めるために、GMT (Greenwich Mean Time) に加算する時間を指定します。このフィールドはルール行の AT フィールドや SAVE フィールドと同じ形式です。時間を GMT から減算しなければならない場合には、フィールドの先頭にマイナス記号を指定します。

RULES/SAVE

タイム・ゾーンに適用するルールの名前、またはローカル標準時に加算する時間を指定します。このフィールドが- の場合には、常にそのタイム・ゾーンにおける標準時が適用されます。

FORMAT

このタイム・ゾーンの短縮形の形式を指定します。タイム・ゾーンの短縮形の可変部を示すために%sを使用します。

UNTIL

GMT オフセットの時間、つまりその地域に対してルールが変化する時刻を指定します。これは次のように指定します。

```
年  
月  
日  
時刻
```

UNTIL を指定した場合には、タイム・ゾーン情報は、指定された時刻まで指定された GMT オフセットと変換ルールにより生成されます。

UNTIL を指定した場合は、次の行は継続行でなければなりません。継続行はゾーン行と同じ形式ですが、Zone という文字列と名前は省略されます。継続行は、前の行の UNTIL フィールドに指定された時刻から始まる情報を、そのファイルに格納します。継続行にはゾーン行と同様に UNTIL フィールドを指定することができ、その場合は、次の行がさらに継続行であることを示します。

1.2.3 リンク行

リンク行の形式は次のとおりです。

```
Link    LINK-FROM    LINK-TO
```

次の例を参照してください。

```
Link    US/Eastern    EST5EDT
```

OpenVMS インプリメンテーションでは、Link はコピーとして解釈されます。したがって、上記の行は情報を US/Eastern から EST5EDT にコピーします。

LINK-FROM フィールドは、ゾーン行に NAME フィールドとして表示されます。LINK-TO フィールドは、そのタイム・ゾーンの別の名前として使用されます。

継続行を除き、行はどの順序で入力してもかまいません。

注意

2 つ以上のローカル時刻を使用する地域では、コンパイルされたファイルに記録される最も早い遷移時刻 (transition time) が正しくなるように、遷移時刻 (transition time) が最も早いルールの AT フィールドで、ローカル標準時を指定する必要があります。

ロケール・ファイルの形式

ロケール定義ソース・ファイルは、ロケールを記述する 1 つ以上のカテゴリで構成されます。ロケール定義ソース・ファイルは、LOCALE_COMPILE コマンドを使用することによって、ロケールに変換できます。ロケールの変更は、ロケール定義ソース・ファイルを編集し、その新しいソース・ファイルに対してもう一度 LOCALE_COMPILE コマンドを実行することによってのみ行うことができます。ロケール・ソース・ファイルの各セクションが、ロケール・データのカテゴリを定義します。ソース・ファイルは、同じカテゴリに対して複数のセクションを持つことはできません。

2.1 ロケール・カテゴリ

次の標準ロケール・カテゴリがサポートされます。

- LC_COLLATE — 文字または文字列照合情報を定義する。
- LC_CTYPE — 文字分類、大文字/小文字変換、および他の文字属性を定義する。
- LC_MESSAGES — 肯定応答および否定応答の形式を定義する。
- LC_MONETARY — 通貨数値情報を書式化する上での規則および記号を定義する。
- LC_NUMERIC — 非通貨数値情報を書式化する上での規則および記号を定義する。
- LC_TIME — 日付/時刻情報を書式化する上での規則および記号を定義する。

2.1.1 デフォルトの設定の変更

ロケール・ソース・ファイルの先頭に、オプションの宣言を指定することによって、ロケール・カテゴリ定義で使用される、デフォルトのコメント文字およびエスケープ文字を変更できます。

- エスケープ文字

エスケープ文字は、ロケール・ファイルの中で 10 進または 16 進定数を指定するときに使用します。デフォルトのエスケープ文字は、バックスラッシュ (\) です。別のエスケープ文字を定義するには、次の形式で行を指定します。

```
escape_char <char_symbol>
```

- コメント文字

コメント文字は、ロケール・ファイル内の、コメント・エントリの先頭文字です。デフォルトのコメント文字は、番号記号(#)です。別のコメント文字を定義するには、次の形式を使用します。

```
comment_char <char_symbol>
```

上の例で、<char_symbol>は、ロケール・コードセットを作成するのに使用された charmap ファイルで定義されている、文字のシンボル名です。escape_char または comment_char と、<char_symbol>とのあいだは、1つ以上の空白文字(スペースまたはタブ)で区切らなければなりません。

2.1.2 カテゴリ・ソース定義

各カテゴリ・ソース定義は、次の内容で構成されます。

- カテゴリ・ヘッダ (*category_name*)
- カテゴリ本体を構成する関連キーワードまたは値のペア
- カテゴリ・トレーラ (END *category_name*)

例を次に示します。

```
LC_CTYPE
<source for LC_CTYPE category>
END LC_CTYPE
```

すべてのカテゴリについて、ソースは、キーワード、文字列、文字リテラル、および文字シンボルを使用して指定します。各キーワードは、定義と規則のいずれかを示しています。キーワードが含まれている文の残りの部分には、キーワードに対するオペランドが含まれます。オペランドとキーワードは、1つ以上の空白文字(スペースまたはタブ)で区切ります。なお、行を終了する改行文字の前に、最終文字としてバックslash (\)を入力すると、文は次の行に続きます。1桁目にコメント文字(#)がある行は、コメント行として扱われます。

シンボル名は、左山括弧文字(<)ではじまり、右山括弧文字(>)で終了します。<および>のあいだには、制御文字およびスペース文字を除く、Portable Character Set のすべての文字を指定できます。たとえば、<A-diaeresis>という指定は、ある文字のシンボル名として有効です。ロケール・ソース・ファイルの中で参照されるシンボル名は、Portable Character Set を使用して、またはそのロケール用の文字セット記述(charmap) ファイルの中で定義する必要があります。

文字リテラルは、文字そのもの、あるいは10進、16進、または8進定数です。10進定数は、2桁または3桁の10進数により、次の形式で構成されます。ここで、*n*は10進数を表します。

`\dnn` or `\dnnn`

16 進定数は、2桁の16進数により、次の形式で構成されます。ここで、*n*は16進数を表します。

`\xnn`

8 進定数は、2桁または3桁の8進数により、次の形式で構成されます。ここで、*n*は8進数を表します。

`\nn` or `\nnn`

ロケール定義ソース・ファイル内で、すべてのカテゴリを明示的に定義する必要はありません。ロケール定義ソース・ファイルの中でカテゴリが定義されていない場合、LOCALE_COMPILE コマンドの出力するロケール・ファイルには、このカテゴリのデータは何も格納されません。

2.2 LC_COLLATE カテゴリ

LC_COLLATE カテゴリは、照合項目間の相対順序を定義します。このカテゴリは、LC_COLLATEヘッダではじまり、END LC_COLLATEトレーラで終了します。

照合項目は、照合を行う上での比較の単位であり、文字または文字の並びです。ロケール内の照合項目はすべて、これがロケール内の別の照合項目より以前、等しい、または以降であるかどうかを決定する重みのセットを持っています。ロケール項目はそれぞれ、ロケール・ソース・ファイルをコンパイルするときに、LOCALE_COMPILE コマンドによって、照合重みが割り当てられます。これらの照合重みは、文字列を比較するアプリケーション・プログラムによって使用されます。

文字列の比較は、文字列内の各文字の照合重みの比較を、相違が検出されるまで、または文字列が等しいと判断されるまで実行することによって行われます。この比較は、ロケールが複数の照合順序を定義している場合は、数回に渡って行われることがあります。たとえば、フランスのロケールの場合、文字列は、照合重みの1次セットを使用して比較されます。この比較の段階で両者が等しいと判断されると、照合重みの2次セットを使用してもう一度比較が行われます。照合項目は、ロケールに対して定義されている照合ソート規則の数だけ、関連する照合重みのセットを持ちます。

charmap ファイル内で定義される各文字 (charmap ファイルが指定されていない場合は、Portable Character Set 内のすべての文字) は、これ自体が照合項目です。新たな照合項目は、collating-element文を使用して定義できます(次の説明参照)。

表 2-1 は、LC_COLLATEカテゴリで認識される文キーワードの一覧です。

表 2-1 LC_COLLATE カテゴリ・キーワード

キーワード	説明
copy	このカテゴリの定義として使用する、既存のロケール名を指定する。copy 文を指定すると、このカテゴリの中で他のキーワードを指定する必要はない。
collating-element	複数文字の照合項目を指定する。
collating-symbol	照合シーケンス文で使用する照合シンボルを指定する。
order_start	照合重みを照合項目に割り当てる照合シーケンス文を指定する。

collating-element 文, collating-symbol 文, および order_start 文については, 以降の項で詳しく説明します。

2.2.1 collating-element 文

collating-element 文は, 複数文字の照合項目を指定します。

構文

```
collating-element <character_symbol> from <string>
```

character_symbol 引数は, 1 文字以上の文字列の照合項目を, 1 つの照合項目として定義します。*character_symbol* は, 現在の charmap ファイル内のシンボル名, またはこの照合定義の中で定義されている他のシンボル名と重複することはできません。

string 引数は, *character_symbol* 引数を定義する 2 文字以上の文字列を指定します。collating-element 文の例を次に示します。

```
collating-element <ch> from "<c><h>"
collating-element <e-acute> from "<acute><e>"
collating-element <l1> from "<l><l>"
```

collating-element 文で定義される *character_symbol* 引数は, LC_COLLATE カテゴリの中に限って認識されます。

2.2.2 collating-symbol 文

collating-symbol 文は, 照合シーケンス文で使用する照合シンボルを指定します。

構文

```
collating-symbol <collating_symbol>
```

collating_symbol 引数は, 現在の charmap ファイル内のシンボル名, またはこの照合定義の中で定義されている他のシンボル名と重複することはできません。collating-symbol 文の例を次に示します。

```
collating-symbol <UPPER_CASE>
collating-symbol <HIGH>
```

collating_symbol文で定義される引数は、LC_COLLATEカテゴリの中に限って認識されます。

2.2.3 order_start 文

order_start文には、照合重みを照合項目に割り当てる1つ以上の照合順序文、およびorder_endキーワードが続きます。order_start文は、必ず指定する必要があります。

構文

```
order_start sort_rules;sort_rules;...;sort_rules
collation_order_statements
order_end
```

ソート規則

sort_rules指示文は、次の構文で指定します。

```
keyword, keyword, ..., keyword
```

ここで、*keyword*は FORWARD、BACKWARD、POSITION のいずれかです。

sort_rules指示文はオプションです。指定された場合には、文字列比較に適用される規則を定義します。指定されたsort_rules指示文の数は、各照合項目に割り当てられる重みの数を定義します（つまり、指示文は、ロケールの中での照合順序数を定義します）。sort_rules指示文が指定されない場合、1つのforward指示文が想定され、文字列ではなく、文字単位に比較が行われます。

sort_rules指示文がある場合、1次重みを使用して文字列を比較するときに先頭の指示文、2次重みを使用して文字列を比較するときに2番目の指示文がそれぞれ適用されます。以降についても同様です。各sort_rules指示文のセットは、セミコロン(;)で区切ります。sort_rules指示文は、コンマで区切った1つ以上のキーワードで構成されます。次のキーワードがサポートされています。

FORWARD — 照合重みの比較が、文字列の先頭から最後に向かって行われることを指定する。

BACKWARD — 照合重みの比較が、文字列の最後から先頭に向かって行われることを指定する。

POSITION — 照合重みの比較が、文字列内の対象要素の相対位置を考慮することを指定する（文字列が等しいことを比較する場合、比較の開始位置から最短の位置にある要素を、最初に照合する）。

forward キーワードおよび backward キーワードは、相互に排他的です。

`sort_rules`指示文の例を次に示します。

```
order_start      forward;backward
```

照合順序文

照合順序文には次の構文規則が適用されます。

- 各照合順序文は、`<character_symbol>`指定、空白、照合順序セットの順で構成される。
- 文字セット内の文字は、照合順序文で明示的に指定するか、または反復記号 (...) を使用して暗黙に指定できる。
- UNDEFINED 特殊シンボルではじまる照合順序文は、他の照合順序文で明示的または暗黙に指定されていない文字セット内の文字を指定する。

照合項目の 1 次重み、2 次重み、または以降の重みを定義するには、各照合項目に対してオプションのオペランドを使用します。また、文字列を比較する場合に無視される照合項目を指定するには、特殊シンボル IGNORE を使用します。

`collating_element_list` の位置に指定される反復記号キーワードは、指定範囲内の文字に対して、先行する文の左側にある文字シンボルの重みを基準に、昇順で、重みが割り当てられることを表します。

反復記号キーワードを使用すると、異なる文字セット記述 (`charmap`) ソース・ファイルでコンパイルした場合に、異なる照合を行うロケールが作成されます。

UNDEFINED 特殊シンボルは、明示的に、または反復記号シンボルで指定されていない、すべてのコード化文字セット値を含みます。これらの文字は、UNDEFINED 特殊シンボルで指示される位置に、文字照合順序に従って挿入され、同じ重みがすべて割り当てられます。UNDEFINED 特殊シンボルがなく、照合順序が、コード化文字セットからすべての照合項目を指定していない場合、警告メッセージが表示され、未定義文字がすべて、文字照合順序の最後に設定されます。

例

LC_COLLATEロケール定義ソース・ファイル・カテゴリ内の、照合順序文セクションの例を次に示します。

```
order_start      forward;backward
UNDEFINED        IGNORE;IGNORE
<LOW>
<space>          <LOW>;<space>
...              <LOW>;...
<a>              <a>;<a>
<a-acute>        <a>;<a-acute>
<a-grave>        <a>;<a-grave>
<A>              <a>;<A>
<A-acute>        <a>;<A-acute>
<A-grave>        <a>;<A-grave>
```

```

<ch>          <ch>;<ch>
<Ch>          <ch>;<Ch>
<s>           <s>;<s>
<ss>          <s><s>;<s><s>
<eszet>       <s><s>;<eszet><eszet>
...           <HIGH>;...
<HIGH>
order_end

```

この例は、次のように解釈されます。

- UNDEFINED 特殊シンボルは、定義内に明示的に、または反復記号で指定されていない文字はすべて、照合の際に無視されることを表す。
- <space>と<a>のあいだの照合項目はすべて、同じ 1 次等価クラス、およびコード化文字セット値に基づく個別の 2 次重みを持つ。
- あらゆる種類の文字 a (大文字と小文字、発音区別符の有無) は、同じ 1 次照合クラスに属する。
- <c><h>という複数文字照合項目は、<ch>照合シンボルで表わされ、<C><h>複数文字照合項目と同じ 1 次等価クラスに属する。
- <eszet>文字は、<s><s>文字列として照合される (つまり、1 つの<eszet>文字が、比較の前に 2 つの文字に展開される)。

2.3 LC_CTYPE カテゴリ

LC_CTYPE カテゴリは、文字分類、大文字/小文字変換、および他の文字属性を定義します。このカテゴリはLC_CTYPEヘッダではじまり、END LC_CTYPEトレーラで終了します。

LC_CTYPE カテゴリ文に対するオペランドはすべて、文字のリストとして定義されます。各リストは、セミコロンで区切られた 1 つ以上の文字またはシンボル文字名で構成されます。反復記号(...) は、一連の文字を表します。たとえば、<a>;...;<z> という指定は、a ~ z の範囲を表します。

表 2-2 は、LC_CTYPE カテゴリで認識される文キーワードの一覧です。キーワードの説明の中で、「自動的に含まれる」という表現は、参照文字が含まれる場合も含まれない場合も、エラーが発生しないことを表します。指定がない場合は文字が提供され、指定されている場合はその指定が適用されます。

表 2-2 LC_CTYPE カテゴリ・キーワード

キーワード	説明
copy	このカテゴリの定義として使用する、既存のロケール名を指定する。 copy文を指定すると、このカテゴリの中で他のキーワードを指定する必要はない。
upper	大文字を定義する。 cntrl, digit, punct, またはspaceキーワードで定義される文字は指定しない。大文字の A ~ Z は、このセットに自動的に含まれる。
lower	小文字を定義する。 cntrl, digit, punct, またはspaceキーワードで定義される文字は指定しない。小文字の a ~ z は、このセットに自動的に含まれる。
alpha	すべての英文字を定義する。 cntrl, digit, punct, またはspaceキーワードで定義される文字は指定しない。upperおよびlowerキーワードで定義される文字は、この文字クラスに自動的に含まれる。
digit	数字文字を定義する。 数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 だけを指定できる。数字 0 ~ 9 は、このセットに自動的に含まれる。
space	空白文字を定義する。 upper, lower, alpha, digit, graph, またはxdigitキーワードで定義される文字は指定しない。スペース、改ページ、用紙送り、改行、キャリッジ・リターン、タブ、および垂直タブ文字は、このセットに自動的に含まれる。
cntrl	制御文字を定義する。 upper, lower, alpha, digit, punct, graph, print, またはxdigitキーワードで定義される文字は指定しない。
punct	区切り文字を定義する。 upper, lower, alpha, digit, cntrl, またはxdigitで定義される文字は指定しない。
graph	スペース文字を除く、プリント可能文字を定義する。 cntrlキーワードで定義される文字は指定しない。upper, lower, alpha, digit, xdigit, およびpunctキーワードで定義される文字は、この文字クラスに自動的に含まれる。
print	スペース文字を含む、プリント可能文字を定義する。 cntrlキーワードで定義される文字は指定しない。スペース文字、およびupper, lower, alpha, digit, xdigit, およびpunctキーワードで定義される文字は、この文字クラスに自動的に含まれる。
xdigit	16進数字文字を定義する。 数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 だけを指定できる。ただし、16進値の 10 ~ 15 については、任意の文字を指定できる。これらの代替 16進数字は、数字文字列を 16進から数量に変換するときに、標準変換ルーチンが使用することはない。数字 0 ~ 9, および英字 A ~ F と英字 a ~ f は、このセットに自動的に含まれる。
blank	空白文字を定義する。 この文字クラスには、スペースおよび水平タブ文字が含まれる。この文で定義される任意の文字は、spaceクラスに自動的に含まれる。

(次ページに続く)

表 2-2 (続き) LC_CTYPE カテゴリ・キーワード

キーワード	説明
toupper	<p>小文字から大文字へのマッピングを定義する。</p> <p>このキーワードに対するオペランドは、コンマで区切られた文字のペアで構成される。各文字ペアは括弧()で囲まれ、セミコロン(;)で次のペアと区切られる。各ペアの最初の文字は小文字、2番目の文字は大文字と見なされる。lowerおよびupperキーワードで定義された文字に限って指定できる。toupperの指定を省略すると、デフォルトの設定で、a ~ zはA ~ Zにマップされる。</p>
tolower	<p>大文字から小文字へのマッピングを定義する。</p> <p>このキーワードに対するオペランドは、コンマで区切られた文字のペアで構成される。各文字ペアは括弧()で囲まれ、セミコロン(;)で次のペアと区切られる。各ペアの最初の文字は大文字、2番目の文字は小文字と見なされる。lowerおよびupperキーワードで定義された文字に限って指定できる。</p> <p>tolowerが指定されていない場合は、デフォルトの設定に従って、toupperキーワードが指定されている場合、その逆のマッピングが行われる。toupperとtolowerの両キーワードの指定が省略されると、各マッピングは、Cロケールのマッピングに従う。</p>

新たなキーワードを指定することで、新しい文字分類を定義できます。この例を次に示します。

```
charclass vowel
vowel      <a>;<e>;<i>;<o>;<u>;<y>
```

LC_CTYPE カテゴリは、複数文字要素をサポートしません(たとえば、ドイツ語の Eszet 文字は、通常、小文字として分類されます)。ドイツ語テキストの適切な大文字使用では、Eszet 文字は SS の 2 文字で置き換えられます。これは、対応する大文字がないためです。この種の変換は、toupperおよびtolowerキーワードの範囲を超えています。

ロケール定義ソース・ファイルに指定された LC_CTYPE カテゴリの例を、次に示します。

```

LC_CTYPE
#"alpha" is by default "upper" and "lower"
#"alnum" is by definition "alpha" and "digit"
#"print" is by default "alnum", "punct" and the space character
#"graph" is by default "alnum" and "punct"
#"tolower" is by default the reverse mapping of "toupper"
#
upper  <A>;<B>;<C>;<D>;<E>;<F>;<G>;<H>;<I>;<J>;<K>;<L>;<M>;\
       <N>;<O>;<P>;<Q>;<R>;<S>;<T>;<U>;<V>;<W>;<X>;<Y>;<Z>
#
lower  <a>;<b>;<c>;<d>;<e>;<f>;<g>;<h>;<i>;<j>;<k>;<l>;<m>;\
       <n>;<o>;<p>;<q>;<r>;<s>;<t>;<u>;<v>;<w>;<x>;<y>;<z>
#
digit  <zero>;<one>;<two>;<three>;<four>;<five>;<six>;\
       <seven>;<eight>;<nine>
#
space  <tab>;<newline>;<vertical-tab>;<form-feed>;\
       <carriage-return>;<space>
#
cntrl  <alert>;<backspace>;<tab>;<newline>;<vertical-tab>;\
       <form-feed>;<carriage-return>;<NUL>;<SOH>;<STX>;\
       <ETX>;<EOT>;<ENQ>;<ACK>;<SO>;<SI>;<DLE>;<DC1>;<DC2>;\
       <DC3>;<DC4>;<NAK>;<SYN>;<ETB>;<CAN>;<EM>;<SUB>;\
       <ESC>;<IS4>;<IS3>;<IS2>;<IS1>;<DEL>
#
punct  <exclamation-mark>;<quotation-mark>;<number-sign>;\
       <dollar-sign>;<percent-sign>;<ampersand>;<asterisk>;\
       <apostrophe>;<left-parenthesis>;<right-parenthesis>;\
       <plus-sign>;<comma>;<hyphen>;<period>;<slash>;\
       <colon>;<semicolon>;<less-than-sign>;<equals-sign>;\
       <greater-than-sign>;<question-mark>;<commercial-at>;\
       <left-square-bracket>;<backslash>;<circumflex>;\
       <right-square-bracket>;<underline>;<grave-accent>;\
       <left-curly-bracket>;<vertical-line>;<tilde>;\
       <right-curly-bracket>
#
xdigit <zero>;<one>;<two>;<three>;<four>;<five>;<six>;\
       <seven>;<eight>;<nine>;<A>;<B>;<C>;<D>;<E>;<F>;\
       <a>;<b>;<c>;<d>;<e>;<f>
#
blank  <space>;<tab>
#
toupper (<a>, <A>); (<b>, <B>); (<c>, <C>); (<d>, <D>); (<e>, <E>); \
        (<f>, <F>); (<g>, <G>); (<h>, <H>); (<i>, <I>); (<j>, <J>); \
        (<k>, <K>); (<l>, <L>); (<m>, <M>); (<n>, <N>); (<o>, <O>); \
        (<p>, <P>); (<q>, <Q>); (<r>, <R>); (<s>, <S>); (<t>, <T>); \
        (<u>, <U>); (<v>, <V>); (<w>, <W>); (<x>, <X>); (<y>, <Y>); \
        (<z>, <Z>)
#
END LC_CTYPE

```

2.4 LC_MESSAGES カテゴリ

LC_MESSAGES カテゴリは、肯定および否定のシステム応答の形式を定義します。このカテゴリはLC_MESSAGESヘッダではじまり、END LC_MESSAGESトレーラで終了します。

LC_MESSAGES カテゴリに対するオペランドはすべて、二重引用符(")で囲まれた文字列、または拡張正規表現として定義されます。これらのオペランドは、1つ以上の空白文字(スペースまたはタブ)によって、定義しているキーワードと区切られます。二重引用符が2つ並んでいる場合("")は、未定義値を指定します。

表 2-3 は、LC_MESSAGES カテゴリで認識される文キーワードの一覧です。

表 2-3 LC_MESSAGES カテゴリ・キーワード

キーワード	説明
copy	このカテゴリの定義として使用する、既存のロケール名を指定する。 copy文を指定すると、他のキーワードは指定できない。
yesexpr	肯定または否定応答を求める質問に対する、適切な肯定応答を記述する拡張正規表現を指定する。
noexpr	肯定または否定応答を求める質問に対する、適切な否定応答を記述する拡張正規表現を指定する。
yesstr	適切な肯定応答の、ロケールにおける同値を指定する。 この文字列は、nl_langinfoサブルーチンを使用すると、nl_langinfo(YESSTR)として、アプリケーションからアクセスできる。なお、yesstrは、XPG4 標準で使用されなくなる可能性があるため、代わりにyesexprを使用する方が望ましい。
nostr	適切な否定応答の、ロケールにおける同値を指定する。 この文字列は、nl_langinfoサブルーチンを使用すると、nl_langinfo(NOSTR)として、アプリケーションからアクセスできる。なお、nostrは、XPG4 標準で使用されなくなる可能性があるため、代わりにnoexprを使用する方が望ましい。

ロケール定義ソース・ファイルに指定された LC_MESSAGES カテゴリの例を、次に示します。

```
LC_MESSAGES
#
yesexpr "<circumflex><left-square-bracket><y><Y>\
<right-square-bracket>"
noexpr "<circumflex><left-square-bracket><n><N>\
<right-square-bracket>"
yesstr "<y><e><s>"
nostr "<n><o>"
#
END LC_MESSAGES
```

2.5 LC_MONETARY カテゴリ

LC_MONETARY カテゴリは、通貨数値情報を書式化する上での規則および記号を定義します。このカテゴリはLC_MONETARYヘッダではじまり、END LC_MONETARYトレーラで終了します。

2.5.1 LC_MONETARY キーワード

LC_MONETARY カテゴリ・キーワードに対するオペランドはすべて、文字列または整数値として定義されます。文字列値は、二重引用符(")で囲まれます。値はすべて、定義しているキーワードと、1つ以上の空白文字(スペースまたはタブ)で区切られます。二重引用符が2つ並んでいる場合("")は、未定義の文字列値を示します。また、マイナス1(-1)は、未定義の整数値を示します。

表 2-4 は、LC_MONETARY カテゴリで認識される文キーワードの一覧です。

表 2-4 LC_MONETARY カテゴリ・キーワード

キーワード	説明
copy	このカテゴリの定義として使用する、既存のロケール名を指定する。 copy文を指定すると、他のキーワードは指定できない。
int_curr_symbol	国際通貨記号として使用する文字列を指定する。 このキーワードに対するオペランドは、4文字の文字列である [†] 。先頭の3文字は、英字の国際通貨記号であり、4番目の文字は、国際通貨記号と金額値のあいだに挿入される区切り文字を定義する。
currency_symbol	ローカル通貨記号として使用する文字列を指定する。
mon_decimal_point	金額値を書式化する上で使用される、小数点文字列を指定する。
mon_thousands_sep	書式化された金額値の中で、小数点の左側で数字を位取りするのに使用される、文字セパレータを指定する。

[†]現在の HP C ランタイム・ライブラリのインプリメンテーションでは、実際には5文字以上の文字を指定できる。ただし、ユーザが5文字以上を指定するのは適当ではなく、ここに指定されたとおりに使用する。今後のバージョンの HP C ランタイム・ライブラリでは、4文字制限がインプリメントされる予定である。

(次ページに続く)

表 2-4 (続き) LC_MONETARY カテゴリ・キーワード

キーワード	説明										
mon_grouping	<p>書式化された金額値の中で、数値の位取りの桁数を定義する文字列を指定する。</p> <p>このキーワードに対するオペランドは、セミコロンで区切られた整数の並びで構成される。各整数は、位取りの桁数を指定する。先頭の整数は、小数点のすぐ左側の桁数を定義し、以降の整数は、その左側の桁数を順次定義する。最後の整数が-1でない場合、その整数が残りのすべての数値に対して適用される。これに対して最後の整数が-1の場合、以降の位取りは行われない。</p> <p>mon_grouping文の例を次に示す。値は 123456789 で、mon_thousands_sepのオペランドが' (単一引用符) と仮定すると、次の結果が得られる。</p> <table border="1"> <thead> <tr> <th>mon_grouping</th> <th>Formatted Value</th> </tr> </thead> <tbody> <tr> <td>3;-1</td> <td>123456'789</td> </tr> <tr> <td>3</td> <td>123'456'789</td> </tr> <tr> <td>3;2;-1</td> <td>1234'56'789</td> </tr> <tr> <td>3;2</td> <td>12'34'56'789</td> </tr> </tbody> </table>	mon_grouping	Formatted Value	3;-1	123456'789	3	123'456'789	3;2;-1	1234'56'789	3;2	12'34'56'789
mon_grouping	Formatted Value										
3;-1	123456'789										
3	123'456'789										
3;2;-1	1234'56'789										
3;2	12'34'56'789										
positive_sign	書式化された負でない金額値を表すのに使用される文字列を指定する。										
negative_sign	書式化された負の金額値を表すのに使用される文字列を指定する。										
int_frac_digits	int_curr_symbol値を使用して書式化された金額値で表示される、小数部の桁数 (小数点以降の桁数) を表す整数値を指定する。										
frac_digits	currency_symbol値を使用して書式化された金額値で表示される、小数部の桁数 (小数点以降の桁数) を表す整数値を指定する。										
p_cs_precedes	<p>int_curr_symbolまたはcurrency_symbol文字列が、負でない書式化された金額値の前に表示されるか、後に表示されるかを示す整数を指定する。</p> <p>次のいずれかの整数値を指定する。</p> <ul style="list-style-type: none"> 0 通貨記号が金額値の後に表示される。 1 通貨記号が金額値の前に表示される。 										
p_sep_by_space	<p>int_curr_symbolまたはcurrency_symbol文字列が、負でない書式化された金額値とスペースで区切られるかどうかを示す整数値を指定する。</p> <p>次のいずれかの整数値を指定する。</p> <ul style="list-style-type: none"> 0 通貨記号と金額値をスペースで区切らない。 1 通貨記号と金額値をスペースで区切る。 2 通貨記号とpositive_sign文字列が隣接する場合、両者をスペースで区切る。 										

(次ページに続く)

表 2-4 (続き) LC_MONETARY カテゴリ・キーワード

キーワード	説明
n_cs_precedes	<p>int_curr_symbolまたはcurrency_symbol文字列が、負の書式化された金額値の前に表示されるか、後に表示されるかを示す整数を指定する。</p> <p>次のいずれかの整数を指定する。</p> <p>0 通貨記号が金額値の後に表示される。</p> <p>1 通貨記号が金額値の前に表示される。</p>
n_sep_by_space	<p>int_curr_symbolまたはcurrency_symbol文字列が、負の書式化された金額値とスペースで区切られるかどうかを示す整数値を指定する。</p> <p>次のいずれかの整数値を指定する。</p> <p>0 通貨記号と金額値をスペースで区切らない。</p> <p>1 通貨記号と金額値をスペースで区切る。</p> <p>2 通貨記号とnegative_sign文字列が隣接する場合、両者をスペースで区切る。</p>
p_sign_posn	<p>負でない書式化された金額値における、positive_sign文字列の位置を示す整数値を指定する。</p> <p>次のいずれかの整数値を指定する。</p> <p>0 左括弧と右括弧で、金額値、およびint_curr_symbolまたはcurrency_symbol文字列の両方を囲む。</p> <p>1 positive_sign文字列が、金額値、およびint_curr_symbolまたはcurrency_symbol文字列の前に表示される。</p> <p>2 positive_sign文字列が、金額値、およびint_curr_symbolまたはcurrency_symbol文字列の後に表示される。</p> <p>3 positive_sign文字列が、int_curr_symbolまたはcurrency_symbol文字列の直前に表示される。</p> <p>4 positive_sign文字列が、int_curr_symbolまたはcurrency_symbol文字列の直後に表示される。</p>

(次ページに続く)

表 2-4 (続き) LC_MONETARY カテゴリ・キーワード

キーワード	説明
n_sign_posn	<p>負の書式化された金額値における、negative_sign文字列の位置を示す整数値を指定する。</p> <p>次のいずれかの整数値を指定する。</p> <p>0 左括弧と右括弧で、金額値、およびint_curr_symbolまたはcurrency_symbol文字列の両方を囲む。</p> <p>1 negative_sign文字列が、金額値、およびint_curr_symbolまたはcurrency_symbol文字列の前に表示される。</p> <p>2 negative_sign文字列が、金額値、およびint_curr_symbolまたはcurrency_symbol文字列の後に表示される。</p> <p>3 negative_sign文字列が、int_curr_symbolまたはcurrency_symbol文字列の直前に表示される。</p> <p>4 negative_sign文字列が、int_curr_symbolまたはcurrency_symbol文字列の直後に表示される。</p>

2.5.2 各種の通貨編集形式

1つの文の値を変更することによって、独自のカスタマイズされた通貨編集形式を作成できます。表 2-5 は、p_cs_precedes、p_sep_by_space、およびp_sign_posn文に対して定義されている値のすべての組み合わせを使用した結果です。

表 2-5 各種の通貨編集形式

		p_sep_by_space =	2	1	0
p_cs_precedes = 1	p_sign_posn = 0		(\$1.25)	(\$ 1.25)	(\$1.25)
	p_sign_posn = 1		+\$1.25	+\$ 1.25	+\$1.25
	p_sign_posn = 2		\$1.25 +	\$ 1.25+	\$1.25+
	p_sign_posn = 3		+\$1.25	+\$ 1.25	+\$1.25
	p_sign_posn = 4		\$ +1.25	\$+ 1.25	\$+1.25
p_cs_precedes = 0	p_sign_posn = 0		(1.25 \$)	(1.25 \$)	(1.25\$)
	p_sign_posn = 1		+1.25 \$	+1.25 \$	+1.25\$
	p_sign_posn = 2		1.25\$ +	1.25 \$+	1.25\$+
	p_sign_posn = 3		1.25+ \$	1.25 +\$	1.25+\$
	p_sign_posn = 4		1.25\$ +	1.25 \$+	1.25\$+

ロケール定義ソース・ファイルに指定される LC_MONETARY カテゴリの例を次に示します。

```

LC_MONETARY
#
int_curr_symbol      "<U><S><D><space>"
currency_symbol      "<dollar-sign>"
mon_decimal_point    "<period>"
mon_thousands_sep   "<comma>"
mon_grouping         3
positive_sign        "<plus-sign>"
negative_sign        "<hyphen>"
int_frac_digits      2
frac_digits          2
p_cs_precedes        1
p_sep_by_space       2
n_cs_precedes        1
n_sep_by_space       2
p_sign_posn          3
n_sign_posn          3
#
END LC_MONETARY

```

2.6 LC_NUMERIC カテゴリ

LC_NUMERIC カテゴリは、通貨以外の数値情報を書式化する上での規則およびシンボルを定義します。このカテゴリはLC_NUMERICヘッダではじまり、END LC_NUMERICトレーラで終了します。

LC_NUMERIC カテゴリ・キーワードに対するオペランドはすべて、文字列または整数値として定義されます。文字列値は、二重引用符(")で囲まれます。値はすべて、1つ以上の空白文字(スペースまたはタブ)により、定義しているキーワードと区切られます。二重引用符が2つ並んでいる場合("")は、未定義の文字列値を示します。また、マイナス1(-1)は、未定義の整数値を示します。

表 2-6 は、LC_NUMERIC カテゴリで認識される文キーワードの一覧です。

表 2-6 LC_NUMERIC カテゴリ・キーワード

キーワード	説明
copy	このカテゴリの定義として使用する、既存のロケール名を指定する。copy文を指定すると、他のキーワードは指定できない。
decimal_point	金額以外の数値を書式化する上で使用される、小数点文字列を指定する。 このキーワードの指定を省略することはできない。また、未定義文字列値に設定することはできない。
thousands_sep	書式化された金額以外の数値で、小数点の左側で数字を位取りするのに使用される、文字セパレータを指定する。

(次ページに続く)

表 2-6 (続き) LC_NUMERIC カテゴリ・キーワード

キーワード	説明										
grouping	<p>書式化された金額以外の数値で、数値の位取りの桁数を定義する。</p> <p>このキーワードに対するオペランドは、セミコロンで区切られた整数の並びで構成される。各整数は、位取りの桁数を指定する。先頭の整数は、小数点のすぐ左側の桁数を定義し、以降の整数は、その左側の桁数を順次定義する。位取りは、groupingキーワードで指定されている整数ごとに行われる。最後の整数が-1でない場合、その整数が残りのすべての数値に対して繰り返し適用される。これに対して最後の整数が-1の場合、以降の位取りは行われない。</p> <p>grouping文の例を次に示す。値は 123456789 で、thousands_sepオペランドが' (単一引用符) と仮定すると、次の結果が得られる。</p>										
	<table border="1"> <thead> <tr> <th>grouping</th> <th>Formatted Value</th> </tr> </thead> <tbody> <tr> <td>3;-1</td> <td>123456'789</td> </tr> <tr> <td>3</td> <td>123'456'789</td> </tr> <tr> <td>3;2;-1</td> <td>1234'56'789</td> </tr> <tr> <td>3;2</td> <td>12'34'56'789</td> </tr> </tbody> </table>	grouping	Formatted Value	3;-1	123456'789	3	123'456'789	3;2;-1	1234'56'789	3;2	12'34'56'789
grouping	Formatted Value										
3;-1	123456'789										
3	123'456'789										
3;2;-1	1234'56'789										
3;2	12'34'56'789										

ロケール定義ソース・ファイルに指定される LC_NUMERIC カテゴリの例を次に示します。

```
LC_NUMERIC
#
decimal_point    "<period>"
thousands_sep   "<comma>"
grouping         <3>
#
END LC_NUMERIC
```

2.7 LC_TIME カテゴリ

LC_TIME カテゴリは、日付/時刻情報を書式化する上での規則および記号を定義します。このカテゴリはLC_TIMEヘッダではじまり、END LC_TIMEトレーラで終了します。

LC_TIME カテゴリ・キーワードに対するオペランドはすべて、文字列または整数値として定義されます。文字列値は、二重引用符(")で囲まれます。値はすべて、1つ以上の空白文字(スペースまたはタブ)により、定義しているキーワードと区切られます。二重引用符が2つ並んでいる場合("")は、未定義の文字列値を示します。なお、LC_TIMEカテゴリを照会して、日付/時刻形式の要素を表すコマンドおよびサブルーチンは、フィールド記述子を使用します。このフィールド記述子については、後で説明します。

2.7.1 キーワード

表 2-7 は、LC_TIME カテゴリで認識される文キーワードの一覧です。

表 2-7 LC_TIME カテゴリ・キーワード

キーワード	説明
copy	このカテゴリの定義として使用する，既存のロケール名を指定する。 copy文を指定すると，他のキーワードは指定できない。
abday	%a フィールド記述子に対応する，曜日の短縮名を定義する。 認識される値は，セミコロンで区切られた 7 つの文字列で構成される。最初の文字列が週の最初の曜日の短縮名 (Sun)，2 番目の文字列が週の 2 番目の曜日の短縮名，以降についても同様に対応する。
day	%A フィールド記述子に対応する，曜日の完全名を定義する。 認識される値は，セミコロンで区切られた 7 つの文字列で構成される。最初の文字列が週の最初の曜日の完全名 (Sunday)，2 番目の文字列が週の 2 番目の曜日の完全名，以降についても同様に対応する。
abmon	%b フィールド記述子に対応する，月の短縮名を定義する。 認識される値は，セミコロンで区切られた 12 個の文字列で構成される。最初の文字列が最初の月の短縮名 (Jan)，2 番目の文字列が 2 番目の月の短縮名，以降についても同様に対応する。
mon	%B フィールド記述子に対応する，月の完全名を定義する。 認識される値は，セミコロンで区切られた 12 個の文字列で構成される。最初の文字列が最初の月の完全名 (January)，2 番目の文字列が 2 番目の月の完全名，以降についても同様に対応する。
d_t_fmt	%c フィールド記述子に対応する，標準の日付/時刻形式に使用される文字列を定義する。文字列には，文字とフィールド記述子の任意の組み合わせを指定できる。
d_fmt	%x フィールド記述子に対応する，標準の日付形式に使用される文字列を定義する。文字列には，文字とフィールド記述子の任意の組み合わせを指定できる。
t_fmt	%X フィールド記述子に対応する，標準の時刻形式に使用される文字列を定義する。文字列には，文字とフィールド記述子の任意の組み合わせを指定できる。
am_pm	%p フィールド記述子に対応する， a.m. (午前) および p.m. (午後) を表すのに使用される文字列を定義する。 認識される値は，セミコロンで区切られた 2 つの文字列で構成される。最初の文字列が a.m. 指定，2 番目の文字列が p.m. 指定に対応する。
t_fmt_ampm	am_pm値 (%p フィールド記述子) を含む，標準の 12 時間表示の時刻形式に使用される文字列を定義する。 この文は，%r フィールド記述子に対応する。文字列には，文字列とフィールド記述子の任意の組み合わせが指定できる。文字列が空の場合，12 時間形式がロケールでサポートされることはない。
era	ロケールにおける，年の数えかた，および各年号での表示のしかたを定義する。これは，%E フィールド記述子に対応する。 年号ごとに，次の形式で 1 つの文字列を指定する必要がある。

direction:offset:start_date:end_date:name:format

(次ページに続く)

表 2-7 (続き) LC_TIME カテゴリ・キーワード

キーワード	説明
	<p>年号文字列編集形式の変数は、次のように定義される。</p> <ul style="list-style-type: none"> • <i>direction</i> — マイナス(-)またはプラス(+)文字を指定する。 マイナス文字(-)は、開始日付から終了日付に向かって、負の方向で年を数えることを指定する。これに対してプラス文字(+)は、開始日付から終了日付に向かって、正の方向で年を数えることを指定する。 • <i>offset</i> — %Ey フィールド記述子に対応する、年号の最初の年を表す数値を指定する。 • <i>start_date</i> — 年号の開始日付をyyyy/mm/ddの形式で指定する。ここで yyyy, mm, dd は、グレゴリオ暦の年、月、日にそれぞれ相当する。 A.D. 1 より以前の年は、負の値で表される。たとえば、100 B.C. の 3 月 5 日からはじまる年号は、-100/03/05 で表される。 • <i>end_date</i> — 年号の最終日付を、start_date変数の場合と同じ形式、あるいは-*または+*という特殊値のいずれかで指定する。 -*値は、年号の最終日付が時の起源に向かって逆方向に進むことを表す。これに対して+*値は、年号の最終日付が順方向に進むことを表す。従って、最終日付は、年代順で、年号の開始日付の前でも後でも構わない。たとえば、西暦 A.D. および B.C. に対する文字列は、それぞれ次のように入力する。 <pre>+ : 0 : 0000 / 01 / 01 : + * : AD : %Ey %EC + : 1 : -0001 / 12 / 31 : - * : BC : %Ey %EC</pre> • <i>name</i> — %EC フィールド記述子に置き換えられる、年号の名前を表す文字列を指定する。 • <i>format</i> — %EY フィールド記述子を書式化するとき使用する、strftime, strptime, およびwcsftime編集形式文字列を指定する。 この文字列には、任意のstrftime, strptime, およびwcsftime書式制御文字(%EYを除く)、およびロケールに依存するマルチバイト文字を指定できる。 <p>年号値は、年号ごとに引用符で囲まれた 1 つの文字列で構成される。複数の年号を指定する場合、各年号文字列はセミコロン(;)で区切る。</p>
era_d_fmt	%Ex フィールド記述子に対応する、代替年号形式の日付を表すのに使用される文字列を定義する。文字列には、文字とフィールド記述子の任意の組み合わせを指定できる。
era_t_fmt	strftime, strptime, およびwcsftimeに対して、%EX フィールド記述子で表される、ロケールの代替時刻形式を定義する。
era_d_t_fmt	strftime, strptime, およびwcsftimeに対して、%Ec フィールド記述子で表される、ロケールの代替日付/時刻形式を定義する。

(次ページに続く)

表 2-7 (続き) LC_TIME カテゴリ・キーワード

キーワード	説明
alt_digits	%O フィールド記述子に対応する、数値用の代替文字列を定義する。 認識される値は、セミコロンで区切られた文字列のグループで構成される。先頭の文字列が 0 (ゼロ) の代替文字列、2 番目の文字列が 1 の代替文字列、以降についても同様に対応する。最大で 100 個の代替文字列を指定できる。

2.7.2 フィールド記述子

LC_TIME ロケール定義ソース・ファイルは、フィールド記述子を使用して、日付/時刻形式の要素を表します。これらのフィールド記述子を組み合わせると、別のフィールド記述子を作成したり、日付/時刻形式文字列を作成することができます。フィールド記述子と他の文字を含む書式化文字列で使用される場合、フィールド記述子は、その現在の値で置き換えられます。他の文字はすべて、変更されることなくコピーされます。表 2-8 は、時刻を書式化するときに LC_TIME カテゴリを照会するコマンドおよびサブルーチンによって使用される、フィールド記述子の一覧です。

表 2-8 LC_TIME ロケール・フィールド記述子

フィールド記述子	内容
%a	abday文で定義される曜日の短縮名 (Sun など) を表す。
%A	day文で定義される曜日の完全名 (Sunday など) を表す。
%b	abmon文で定義される月の短縮名 (Jan など) を表す。
%B	month文で定義される月の完全名 (January など) を表す。
%c	d_t_fmt文で定義される日付/時刻形式を表す。
%C	世紀を 10 進数 (00 ~ 99) で表す。
%d	月のうちの日を 10 進数 (01 ~ 31) で表す。
%D	日付を %m/%d/%y の形式 (01/31/91 など) で表す。
%e	月のうちの日を 10 進数 (1 ~ 31) で表す。 月のうちの日が 2 桁の数字でない場合、先頭の桁にはスペース文字が詰められる。
%Ec	ロケールに対して、代替の日時表示を指定する。
%EC	ロケールの代替表示において、基準となる年 (年号) の名前を指定する。
%Ex	ロケールに対して、代替の日付表示を指定する。
%Ey	ロケールの代替表示において、%EC (年のみ) からのオフセットを指定する。
%EY	完全な代替年表示を指定する。
%h	abmon文で定義される月の短縮名 (Jan など) を表す。このフィールド記述子は、%b フィールド記述子の機能に等しい。
%H	24 時間形式の時刻を 10 進数 (00 ~ 23) で表す。
%I	12 時間形式の時刻を 10 進数 (01 ~ 12) で表す。

(次ページに続く)

表 2-8 (続き) LC_TIME ロケール・フィールド記述子

フィールド記述子	内容
%j	年間通算日を 10 進数 (001 ~ 366) で表す。
%m	月を 10 進数 (01 ~ 12) で表す。
%M	分を 10 進数 (00 ~ 59) で表す。
%n	改行文字を指定する。
%Od	月のうちの日を、ロケールの代替数値シンボルで指定する。
%Oe	月のうちの日を、ロケールの代替数値シンボルで指定する。
%OH	時間 (24 時間形式) を、ロケールの代替数値シンボルで指定する。
%OI	時間 (12 時間形式) を、ロケールの代替数値シンボルで指定する。
%Om	月をロケールの代替数値シンボルで指定する。
%OM	分をロケールの代替数値シンボルで指定する。
%OS	秒をロケールの代替数値シンボルで指定する。
%OU	年頭からの週数 (日曜日を週の最初の曜日とする) を、ロケールの代替数値シンボルで指定する。
%Ow	曜日をロケールの代替表示の番号で指定する (Sunday = 0)。
%OW	年頭からの週数 (月曜日を週の最初の曜日とする) を、ロケールの代替数値シンボルで指定する。
%Oy	年 (%C からのオフセット) を、ロケールの代替数値シンボルで指定する。
%p	am_pm文で定義される a.m. または p.m. 文字列を表す。
%r	t_fmt_ampm文で定義される a.m./p.m. 表記を伴う 12 時間形式の時刻を表す。
%S	秒を 10 進数 (00 ~ 59) で表す。
%t	タブ文字を指定する。
%T	24 時間形式の時刻を%H:%M:%S (16:55:15 など) の形式で表す。
%U	年頭からの週数を 10 進数 (00 ~ 53) で表す。 日曜日、またはday文で定義される曜日を週の最初の曜日として、このフィールド記述子の値が計算される。
%w	曜日を 10 進数 (0 ~ 6) で表す。 日曜日、またはday文で定義される曜日を 0 (ゼロ) として、このフィールド記述子の値が計算される。
%W	年頭からの週数を 10 進数 (00 ~ 53) で表す。 月曜日、またはday文で定義される曜日を週の最初の曜日として、このフィールド記述子の値が計算される。
%x	d_fmt文で定義される日付形式を表す。
%X	t_fmt文で定義される時刻形式を表す。
%y	世紀の年 (00 ~ 99) を表す。
%Y	年を 10 進数 (1989 など) で表す。
%%	% (パーセント記号) 文字を指定する。

2.7.3 ロケール定義のサンプル

ロケール定義ソース・ファイルに指定される LC_TIME カテゴリのサンプルを次に示します。

```
LC_TIME
#
#Abbreviated weekday names (%a)
abday  "<S><u><n>"; "<M><o><n>"; "<T><u><e>"; "<W><e><d>"; \
        "<T><h><u>"; "<F><r><i>"; "<S><a><t>"

#Full weekday names (%A)
day    "<S><u><n><d><a><y>"; "<M><o><n><d><a><y>"; \
        "<T><u><e><s><d><a><y>"; "<W><e><d><n><e><s><d><a><y>"; \
        "<T><h><u><r><s><d><a><y>"; "<F><r><i><d><a><y>"; \
        "<S><a><t><u><r><d><a><y>"

#Abbreviated month names (%b)
abmon  "<J><a><n>"; "<F><e><b>"; "<M><a><r>"; "<A><P><r>"; \
        "<M><a><y>"; "<J><u><n>"; "<J><u><l>"; "<A><u><g>"; \
        "<S><e><P>"; "<O><c><t>"; "<N><o><v>"; "<D><e><c>"

#Full month names (%B)
mon    "<J><a><n><u><a><r><y>"; "<F><e><b><r><u><a><r><y>"; \
        "<M><a><r><c><h>"; "<A><P><r><i><l>"; "<M><a><y>"; \
        "<J><u><n><e>"; "<J><u><l><y>"; "<A><u><g><u><s><t>"; \
        "<S><e><P><t><e><m><b><e><r>"; "<O><c><t><o><b><e><r>"; \
        "<N><o><v><e><m><b><e><r>"; "<D><e><c><e><m><b><e><r>"

#Date-and-time format (%c)
#Note that for improved readability, this section uses actual
#characters, rather than symbolic names, and is inconsistent with
#the other sections in this example. This is bad form.
#In practice, symbolic names should be used.
d_t_fmt  "%a %b %d %H:%M:%S %Y"
#
#Date format (%x)
d_fmt    "%m/%d/%y"
#
#Time format (%X)
t_fmt    "%H:%M:%S"
#
#Equivalent of AM/PM (%p)
am_pm    "<A><M>"; "<P><M>"
#
#12-hour time format (%r)
#Note that for improved readability, this section uses actual
#characters, rather than symbolic names, and is inconsistent with
#the other sections in this example. This is bad form.
#In practice, symbolic names should be used.
t_fmt_ampm "%I:%M:%S %p"
#
era      "+:0:0000/01/01:+*:AD:%Ey %EC"; \
        "+:1:-0001/12/31:-*:BC:%Ey %EC"
```

```
era_d_fmt      ""
alt_digits     "<0><t><h>";"<1><s><t>";"<2><n><d>";"<3><r><d>";\
               "<4><t><h>";"<5><t><h>";"<6><t><h>";"<7><t><h>";\
               "<8><t><h>";"<9><t><h>";"<1><0><t><h>"
#
END LC_TIME
```

文字セット記述 (Charmap) ファイル

本章では、charmap ファイルと呼ばれる文字セット記述ファイルについて説明します。この charmap ファイルは、文字エンコードとして文字シンボルを定義するもので、コード化文字セット、つまりコードセットに対するソース・ファイルです。

3.1 Portable Character Set

サポートされているすべてのコードセットは、固有のサブセットとして Portable Character Set (PCS) を持ちます。PCS は、(標準シンボル名で一覧されている) 文字シンボル、およびその 16 進エンコードで構成されます。表 3-1 を参照してください。

表 3-1 Portable Character Set

シンボル名	16 進エンコード
<NUL>	\x00
<alert>	\x07
<backspace>	\x08
<tab>	\x09
<newline>	\x0A
<vertical-tab>	\x0B
<form-feed>	\x0C
<carriage-return>	\x0D
<space>	\x20
<exclamation-mark>	\x21
<quotation-mark>	\x22
<number-sign>	\x23
<dollar-sign>	\x24
<percent>	\x25
<ampersand>	\x26
<apostrophe>	\x27
<left-parenthesis>	\x28
<right-parenthesis>	\x29
<asterisk>	\x2A

(次ページに続く)

表 3-1 (続き) Portable Character Set

シンボル名	16 進エンコード
<plus-sign>	\x2B
<comma>	\x2C
<hyphen>	\x2D
<period>	\x2E
<slash>	\x2F
<zero>	\x30
<one>	\x31
<two>	\x32
<three>	\x33
<four>	\x34
<five>	\x35
<six>	\x36
<seven>	\x37
<eight>	\x38
<nine>	\x39
<colon>	\x3A
<semi-colon>	\x3B
<less-than>	\x3C
<equal-sign>	\x3D
<greater-than>	\x3E
<question-mark>	\x3F
<commercial-at>	\x40
<A>	\x41
	\x42
<C>	\x43
<D>	\x44
<E>	\x45
<F>	\x46
<G>	\x47
<H>	\x48
<I>	\x49
<J>	\x4A
<K>	\x4B
<L>	\x4C
<M>	\x4D
<N>	\x4E
<O>	\x4F

(次ページに続く)

表 3-1 (続き) Portable Character Set

シンボル名	16 進エンコード
<P>	\x50
<Q>	\x51
<R>	\x52
<S>	\x53
<T>	\x54
<U>	\x55
<V>	\x56
<W>	\x57
<X>	\x58
<Y>	\x59
<Z>	\x5A
<left-bracket>	\x5B
<backslash>	\x5C
<right-bracket>	\x5D
<circumflex>	\x5E
<underscore>	\x5F
<grave-accent>	\x60
<a>	\x61
	\x62
<c>	\x63
<d>	\x64
<e>	\x65
<f>	\x66
<g>	\x67
<h>	\x68
<i>	\x69
<j>	\x6A
<k>	\x6B
<l>	\x6C
<m>	\x6D
<n>	\x6E
<o>	\x6F
<p>	\x70
<q>	\x71
<r>	\x72
<s>	\x73
<t>	\x74

(次ページに続く)

表 3-1 (続き) Portable Character Set

シンボル名	16 進エンコード
<u>	\ x75
<v>	\ x76
<w>	\ x77
<x>	\ x78
<y>	\ x79
<z>	\ x7A
<left-brace>	\ x7B
<vertical-line>	\ x7C
<right-brace>	\ x7D
<tilde>	\ x7E

3.2 charmap ファイルの構成要素

charmap ファイルは、次の要素で構成されています。

- オプションの特殊シンボル名宣言セクション

このセクション内の各宣言は、特殊シンボル名、1 つ以上のスペースまたはタブ文字、および値の順で構成されます。次のリストは、宣言セクションに指定できる特殊シンボル名の説明です。

<code_set_name>

charmap ファイルの定義の対象となるコードセット名を指定します。この値は、nl_langinfo (CODESET) サブルーチンで返される値を決定します。<code_set_name>が宣言されない場合、Portable Character Set の名前が使用されます。

<mb_cur_max>

コードセット内の文字の最大バイト数を指定します。値は 1 ~ 4 の範囲で有効です。省略時の値は 1 です。

<mb_cur_min>

コードセット内の文字の最小バイト数を指定します。サポートされるコードセットはすべて、固有のサブセットとして Portable Character Set を持つため、この値は必ず 1 です。

<escape_char>

16 進または 8 進表記のエンコードを示すエスケープ文字を指定します。省略時の値はバックスラッシュ(\)です。

<comment_char>

charmap ファイル内でのコメントを示すために使用される文字を指定します。省略時の値は番号記号(#)です。

- CHARMAP セクション・ヘッダ

このヘッダは、文字シンボルとエンコードを対応させるセクションの先頭を指定します。

- コードセット内の文字のマッピングを指定する文

各文は、文字のシンボル名、およびその文字に対応するエンコードを指定します。マッピング文は、次の形式で指定します。

<char_symbol> encoding

シンボル名は左山括弧(<)文字ではじまり、右山括弧(>)文字で終了します。*char_symbol* (<および>で囲まれた名前)は、制御文字とスペース文字を除く、Portable Character Set のすべての文字を使用して指定できます。*char_symbol*に>を使用することもできます。この場合、最後の>を除くすべての>文字の前に、<escape_char> 特殊シンボル名で指定されるエスケープ文字を指定します。

エンコードは1文字以上の文字定数で指定しますが、その最大文字数は、<mb_cur_max>特殊シンボル名で指定されます。エンコードは、次の形式の10進数、8進数、または16進数の定数で指定します。

- 10進定数: \dnnまたは\dnnn。ここでnは任意の10進値。
- 8進定数: \nnまたは\nnn。ここでnは任意の8進値。
- 16進定数: \xnn。ここでnは任意の16進値。

文字シンボル定義のサンプルを次に示します。

```
<A>      \d65      #decimal constant
<B>      \x42      #hexadecimal constant
<j10101> \x81\xA1  #multiple hexadecimal constants
```

シンボル名の範囲とこれに対応するエンコード値を定義することもできます。なお、各シンボル名は、数値でない共通接頭辞を持ち、2番目のシンボル名の数値部分が、最初のシンボル名の数値部分以上であることが必要です。この形式では、シンボル名値が、ゼロ文字以上の非数値文字、および1けた以上の10進数の整数の順で構成されます。この指定形式は、一連のシンボル名を定義します。たとえば、文字列<j0101>...<j0104>は、シンボル名<j0101>、<j0102>、<j0103>、および<j0104>として、順に解釈されます。

シンボル名の範囲を定義する文の中で、指定されたエンコード値は、範囲内の先頭のシンボル名の値に相当します。以降のシンボル名は、昇順のエンコード値を持ちます。次にサンプル文を示します。

```
<j0101>...<j0104>      \d129\d254
```

このサンプル文は、次のように解釈されます。

```
<j0101> \d129\d254
<j0102> \d129\d255
<j0103> \d130\d0
<j0104> \d130\d1
```

1つのシンボル名に対して複数のエンコードを割り当てることはできませんが、文字によっては複数の共通名を持つため、1つのエンコード値に対して複数の名前を作成することはできます。たとえば、.文字は、ピリオドと呼ぶこともあれば、終止符 (full stop) と呼ぶこともあります。両方の名前を、charmap ファイルに指定できます。この例を次に示します。

```
<period> \x2e
<full-stop> \x2e
```

コメントは、<comment_char>特殊シンボル名で指定される文字で開始しなければなりません。行全体がコメントの場合は、行の1桁目に、<comment_char>を指定することが必要です。

- END CHARMAP セクション・トレーラ

このトレーラは、マッピング文の終了を指定します。

サンプル charmap ファイルの一部を、次に示します。

```
CHARMAP
<code_set_name> "ISO8859-1"
<mb_cur_max> 1
<mb_cur_min> 1
<escape_char> \
<comment_char> #

<NUL> \x00
<SOH> \x01
<STX> \x02
<ETX> \x03
<EOT> \x04
<ENQ> \x05
<ACK> \x06
<alert> \x07
<backspace> \x08
<tab> \x09
<newline> \x0a
<vertical-tab> \x0b
<form-feed> \x0c
<carriage-return> \x0d
END CHARMAP
```

コマンド・リファレンス

HP C ランタイム・ライブラリ・ユーティリティで提供される、次のコマンドとその構文について説明します。

- GENCAT
- ICONV COMPILE
- ICONV CONVERT
- LOCALE COMPILE
- LOCALE LOAD
- LOCALE UNLOAD
- LOCALE SHOW CHARACTER_DEFINITIONS
- LOCALE SHOW CURRENT
- LOCALE SHOW PUBLIC
- LOCALE SHOW VALUE
- zic

GENCAT

メッセージ・テキスト・ソース・ファイルをメッセージ・カタログ・ファイルにマージします。

形式

```
GENCAT msgfile[...] catfile
```

パラメータ

msgfile

必須。

メッセージ・テキスト・ソース・ファイルの名前。デフォルトのファイル・タイプは.MSGX です。

catfile

必須。

メッセージ・カタログ出力ファイルの名前。catfile がすでに存在する場合には、既存のカタログのメッセージも含んだ新しいバージョンが作成されます。ファイル・タイプは.CAT でなければなりません。

修飾子

なし

説明

GENCAT コマンドは、1 つ以上の入力ソース・ファイル、および既存のカタログ・ファイル(ある場合)から、メッセージ・カタログを新しく作成します。メッセージ・カタログは、アプリケーション用のメッセージを含むバイナリ・ファイルです。エラー・メッセージ、画面表示、プロンプトなど、アプリケーションが出力するすべてのメッセージがここに含まれます。アプリケーションは、C ランタイム・ライブラリ・ルーチンのcatopen、catgets、およびcatcloseを使用して、メッセージ・カタログからメッセージを検索します。これらのルーチンについての詳細は、『HP C ランタイム・ライブラリ・リファレンス・マニュアル』を参照してください。

メッセージ・テキスト・ソース・ファイルは、ユーザが作成するテキスト・ファイルであり、ユーザのプログラムから出力されるメッセージをここに入れます。テキスト・ソース・ファイルには、任意のテキスト・エディタで、メッセージを入力できます。メッセージは、通常、プログラムの機能サブセットに相当するセットにグループ化できます。各メッセージは、そのセット内で一意の数字の識別子を持ちます。メッセージ・テキスト・ソース・ファイルには、セットおよび個別メッセージを処理するための、GENCATで認識されるコマンドを指定することもできます。

メッセージ・テキスト・ソース・ファイルは、いくつ指定しても構いません。GENCAT コマンドは、複数のソース・ファイルを、指定された順に1つずつ処理します。連続する各ソース・ファイルが、カタログを変更します。

*catfile*という名前のメッセージ・カタログが存在する場合、GENCATは、旧バージョンの内容を含むファイルを、新バージョンのファイルとして作成し、これを更新します。メッセージ・カタログが存在しない場合、GENCATは、*catfile*という名前のカタログを作成します。

*catfile*には、次のコマンドを指定できます。

- *message_number text*

*text*をメッセージとして、識別子*message_number* (メッセージ番号)とともに挿入します。この場合、次のガイドラインに従います。

- 番号は、各セット内で昇順でなければならない。番号を飛ばして指定することはできるが、GENCATセッションの中で、飛ばした番号を後で追加したり、既存の番号を置き換えることはできない。
- メッセージ・テキストが空で、スペースまたはタブのフィールド・セパレータがある場合、空白文字列がメッセージ・カタログに格納される。
- メッセージ・ソース行にメッセージ番号が指定されているが、フィールド・セパレータもメッセージ・テキストもない場合、その番号の既存のメッセージ (ある場合) が、カタログから削除される。

- *\$delset set_number*

*set_number*で示されるメッセージのセットを削除します。

- *\$quote character*

引用文字を*character*に設定します。詳細は、例を示した節を参照してください。

- *\$set set_number*

このコマンド以降に入力されるすべてのメッセージを、*set_number*で示されるセットに設定することを指定します。別の*\$set* コマンドを入力することによって、セットを変更できます。ただし、セット番号は昇順で入力しなければなりません。GENCATセッションの中で、番号の低いセットに戻ることはできません。コマンドが使用されない場合、デフォルトのセット番号は1です。

各初期キーワードまたは番号のあとに、空白がなければなりません。GENCAT コーティリティは、行の先頭にスペース、タブ、またはドル記号(\$)文字(スペース、タブ、または改行文字が続く場合)がある場合、その行を無視します。このため、この並びを使用して、*catfile*内でコメントを指定できます。空白行も無視されます。GENCATは、`$delset`、`$quote`、または`$set` コマンドに続く文字を無視するため、これらのコマンドの後にコメントを指定することもできます。

数字ではじまる行は、カタログに取り込まれるメッセージを表します。メッセージ ID 番号とメッセージ・テキストの間に、任意の数の空白を指定できます。ただし、メッセージ・テキストが引用符で区切られていない場合は、スペースまたはタブ文字を1つだけ指定することをおすすめします。メッセージ・テキストが引用符で区切られていない場合、GENCATは、空白もメッセージの一部として処理します。なお、メッセージ・テキストが引用符で区切られている場合には、GENCATは、メッセージ ID と先頭の引用符とのあいだのスペースまたはタブを、すべて無視します。

C 言語で認識されるようなエスケープ・シーケンスを、テキストの中で使用できます。エスケープ文字のバックスラッシュ(\)を使用すると、メッセージ・テキストに特殊文字を挿入できます。表 4-1 を参照してください。

表 4-1 特殊文字

エスケープ・シーケンス	文字
<code>\n</code>	改行
<code>\t</code>	水平タブ
<code>\v</code>	垂直タブ
<code>\b</code>	バックスペース
<code>\r</code>	キャリッジ・リターン
<code>\f</code>	改ページ
<code>\\</code>	バックスラッシュ文字(\)。メッセージ・テキストを次の行に継続するのに使用する。
<code>\ddd</code>	8進数 <code>ddd</code> に対応するシングル・バイト文字。1 ~ 3桁の8進数を指定できる。ただし、8進数に続く文字がやはり有効な8進数の場合は、先頭にゼロを指定する必要がある。たとえば、\$(ドル記号)の8進数は44である。メッセージに\$5.00を挿入するには、 <code>\445.00</code> ではなく、 <code>\0445.00</code> を使用する。そうでない場合には、5が、8進数の一部として解析される。

エラー

GENCATがエラーを報告した場合、コマンドに対して何らかの処理が行われることはなく、既存のカタログはそのまま変更されません。

例

1. `$set 10 Communication Error Messages`

この例は、ソース・ファイルの中で`$set` コマンドを使用し、メッセージ・グループに対してセット番号を割り当てます。

メッセージ・セット番号は 10 です。この`$set` コマンドと次の`$set` コマンドの間のすべてのメッセージに、メッセージ・セット番号 10 が割り当てられます(セット番号は昇順に割り当てられなければならないかもしれませんが、番号が連続している必要はありません)。

なお、`$set` コマンドにはコメントを指定できます。

2. `$delset 10 Communication Error Messages`

この例は、`$delset` コマンドを使用して、指定されたメッセージ・セット(この場合は 10) に属するすべてのメッセージを、カタログから削除します。

`$delset` コマンドは、同じソース・ファイル内の`$set` コマンドに関して、正しいセット番号の順序で指定する必要があります。なお、`$delset` コマンドにはコメントを指定できます。

3. `12 "file removed"`

この例は、メッセージ・テキストの入力、およびそれに対するメッセージ ID 番号の割り当て方法を示します。この場合、メッセージ ID 12 が、これに続くテキストに割り当てられます。

メッセージ ID 番号とメッセージ・テキストのあいだには、少なくとも 1 つのスペースまたはタブ文字が必要ですが、複数のスペースまたはタブを指定しても構いません。複数のスペースまたはタブを指定した場合、これらは、メッセージ・テキストが引用符で囲まれていると無視され、メッセージ・テキストが引用符で囲まれていなければテキストの一部と見なされます。

メッセージ番号は、1 つのメッセージ・セットの中で昇順でなければなりません。番号が連続している必要はありません。

メッセージ番号の後から行の終端までのすべてのテキストが、メッセージ・テキストとして取り込まれます。行の最終文字としてエスケープ文字のバックslash (\) を指定すると、メッセージ・テキストは次の行に続きます。次の例を参照してください。

```
This is the text associated with \  
message number 5.
```

この例に示す 2 行は、次の 1 行のメッセージを定義します。

```
This is the text associated with message number 5.
```

- ```

4. $quote " Use a double quote to delimit message text
 $set 10 Message Facility - Quote command messages
 1 "Use the $quote command to define a character \
 \n for delimiting message text" \n
 2 "You can include the \"quote\" character in a message \n \
 by placing a \\ (backslash) in front of it" \n
 3 You can include the "quote" character in a message \n \
 by having another character as the first nonspace \
 \n character after the message ID number \n
 $quote
 4 You can disable the quote mechanism by \n \
 using the $quote command without \n a character \
 after it \n

```

この例は、引用文字 (引用符) の効果を示します。

\$quote コマンドは、二重引用符(")を引用文字として定義します。引用文字は、メッセージ番号の後に指定されるスペース以外の最初の文字でなければなりません。2番目の引用文字以降のテキストは、すべて無視されます。

この例は、メッセージ・テキストに引用文字を指定する2通りの方法についても示しています。

- 引用文字の前にバックスラッシュ(\)を指定する。
- メッセージ番号の後の、スペース以外の最初の文字に、別の文字を使用する。これにより、そのメッセージに限って、引用文字を使用できなくなる。

この例は、次のことも示しています。

- 引用文字で囲まれた複数行に渡るメッセージを分割するには、やはりバックスラッシュ(\)が必要である。
- メッセージの中でバックスラッシュ(\)を表示するには、バックスラッシュ(\)を重ねて指定する必要がある。
- \nを使用すると、メッセージを改行文字で編集できる。
- \$quote コマンドを文字引数なしで使用すると、引用文字メカニズムが使用できなくなる。

---

# ICONV COMPILE

変換ソース・ファイルから変換テーブル・ファイルを作成します。変換テーブル・ファイルは、ICONV CONVERT コマンドが、ファイル内の文字を、あるコードセットから別のコードセットに変換するのに使用されます。

---

## 形式

ICONV COMPILE sourcefile tablefile

---

## パラメータ

sourcefile

必須。

変換ソース・ファイルの名前。デフォルトのファイル・タイプは.ISRC です。弊社は変換ソース・ファイルのファイル命名規則として、次の規則を使用します。

fromcodeset\_tocodeset.isrc

tablefile

必須。

作成する変換テーブル・ファイルの名前。デフォルトのファイル・タイプは.ICONV です。変換テーブル・ファイルのファイル命名規則は次のとおりです。

fromcodeset\_tocodeset.iconv

公用変換テーブル・ファイルは、論理名SYS\$I18N\_ICONVで定義されるディレクトリにあります。新しく作成した変換テーブル・ファイルを、システム単位で使用できるようにするには、これを同じディレクトリに格納します。

---

## 修飾子

/LISTING[=listfile]

ICONV COMPILE に対して、ソース・ファイル・リスト、およびコンパイル中に生じたエラー・メッセージを含む、リスト・ファイルの作成を指定します。ファイル名の指定が修飾子で省略された場合、デフォルトのリスト・ファイル名は、*sourcefile.LIS* となります。

---

 説明

ICONV コマンドは、状態独立の 1 ~ 4 バイト・コードセットをサポートします。状態依存のコードセットはサポートしません。

---

 注意
 

---

このインプリメンテーションにおける *tocodeset* エンコードには、インプリメンテーションの制約があります。 *tocodeset* 内の文字は、4 番目のバイトに 0xFF を使用してはなりません。

---

変換ソース・ファイルには、特定の変換についての文字変換規則が含まれます。

コードセット変換ソース・ファイルの形式は、次のように定義されます。

```

<fromcodeset_mb_cur_max> value
<fromcodeset_mb_cur_min> value
<tocodeset_mb_cur_max> value
<tocodeset_mb_cur_min> value
<fallback_code> value
<escape_char> value
<comment_char> value
<fromcodeset_range> value...value;value...value;...;value...value
ICONV_TABLE
fromvalue tovalue
fromvalue tovalue
. .
. .
. .
fromvalue tovalue
END ICONV_TABLE

```

ここで、シンボル<...>およびこれに対応する値は、コードセット定義であり、*fromvalue/tovalue* のペアは、文字変換規則です。

#### コードセット宣言

コードセット宣言は、文字変換規則の前に指定しなければなりません。各宣言は、1 桁目からはじまる山括弧で囲まれたシンボル、1 つ以上の空白（タブまたはスペース）、シンボルに割り当てられる値の順に構成されます。表 4-2 を参照してください。

表 4-2 コードセット宣言

| シンボル                     | 値                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <fromcodeset_mb_cur_max> | fromcodeset 内の文字の最大バイト数。デフォルトの値は 1。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <fromcodeset_mb_cur_min> | fromcodeset 内の文字の最小バイト数。この値は、fromcodeset_mb_cur_max 以下でなければならない。この値が指定されなかった場合、デフォルトの値として fromcodeset_mb_cur_max に設定される。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <tocodeset_mb_cur_max>   | tocodeset 内の文字の最大バイト数。デフォルトの値は 1。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <tocodeset_mb_cur_min>   | tocodeset 内の文字の最小バイト数。この値は、tocodeset_mb_cur_max 以下でなければならない。この値が指定されなかった場合、デフォルトの値として tocodeset_mb_cur_max に設定される。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <fallback_code>          | <p>&lt;fromcodeset_range&gt;に表示されているが、ICONV_TABLE および END ICONV_TABLE 間では指定されていない、<i>fromvalue</i> に対する <i>tovalue</i>。次の 3 種類の値のいずれかを指定する。</p> <ul style="list-style-type: none"> <li>• SAME — <i>tovalue</i> が <i>fromvalue</i> に等しいことを指定する。</li> <li>• ERROR — <i>fromvalue</i> から <i>tovalue</i> への変換がサポートされないことを指定する。ICONV_CONVERT は警告メッセージを出力し、読み込まれた残りのレコードを無視する。HP C ランタイム・ライブラリ・ルーチン <code>iconv</code> は、呼び出し側に対して "illegal character" エラーを返す。</li> <li>• ユーザ定義の <i>tovalue</i> — <i>fromvalue</i> は、指定されたユーザ定義の <i>tovalue</i> に変換される。<br/>ユーザ定義の <i>tovalue</i> はマルチバイト文字を表すことができるが、4 番目のバイトの値として、0xFF を使用できないという制約を伴う。&lt;fallback_code&gt;におけるユーザ定義の <i>tovalue</i> の設定は、文字変換規則値における設定に等しい。8 進、10 進、または 16 進数を使用できる。&lt;fallback_code&gt; が指定されなかった場合、デフォルトの値は SAME となる。</li> </ul> |
| <escape_char>            | 後に続く文字を特別な方法で解釈することを指定するためのエスケープ文字。デフォルトのエスケープ文字は、バックスラッシュ (\) である。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <comment_char>           | 行の 1 桁目に指定された場合、その行を無視することを指定する文字。デフォルトのコメント文字は、番号記号 (#)。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <fromokcodeset_range>    | fromcodeset のエンコード範囲。fromcodeset がマルチバイト・コードセットの場合にこの定義を指定する。fromcodeset が省略された場合、デフォルトの設定はシングル・バイト・コードセットであり、ICONV_COMPILE で作成されるテーブルは、シングル・バイトの fromcodeset 変換だけをサポートする。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

fromcodeset に対してコードセット・エンコード範囲を指定する場合、すべての文字ゾーンを指定しなければなりません。<fromcodeset\_range>の指定で文字ゾーンが抜けている場合、コードセット変換が正しく行われなことがあります。他のHP Cラン

タイム・ライブラリ (RTL) でサポートされる `fromcodeset` に対して、コードセット・エンコード範囲を正しく指定することは、きわめて重要です。この指定が正しく行われなければ、`iconv` と、他の HP C RTL でのコードセット・サポートが一致しません。

たとえば、EUCJP に対する `fromcodeset` 範囲は次のように指定されます。

```
<fromcodeset_range> \x0...\x7f;\x8e\xa1...\x8e\xfe;
 \xa1\xa1...\xfe\xfe;\x8f\xa1\xa1...\x8f\xfe\xfe
```

<fromcodeset\_range>値の設定は、文字変換規則値の設定と同じです。8進、10進、または16進数を使用できます。

#### 文字変換規則

文字変換規則は、1桁目から文字列 `ICONV_TABLE` が指定されている行と、1桁目から文字列 `END ICONV_TABLE` が指定されている行のあいだの、すべての行です。

文字変換規則は、1桁目からはじめる必要があります。

空白行、および1桁目にコメント文字が指定されている行は、無視されます。なお、コメントは省略可能です。

文字変換規則は、次の2種類の形式のいずれかで指定します。

```
fromvalue tovalue
fromvalue...fromvalue tovalue
```

*fromvalue* と *tovalue* の間には、1つ以上の空白 (タブまたはスペース) を指定します。

単一文字変換規則を定義するには、最初の形式を使用します。この例を次に示します。

```
\d32 \d101
\d37 \d106
```

文字変換規則の範囲を定義するには、2番目の形式を使用します。この形式の場合、最後に指定する *fromvalue* は、先頭の *fromvalue* 以上でなければなりません。範囲で定義される連続する *fromvalue* は、昇順で *tovalue* に変換されます。

たとえば、次の行について考えてみます。

```
\d223\d32...\d223\d35 \d129\d254
```



この行は、次のように解釈されます。

```
\d223\d32 \d129\d254
\d223\d33 \d129\d255
\d223\d34 \d130\d0
\d223\d35 \d130\d1
```

*fromvalue*および*tovalue*は、次のように設定されます。

- 10進定数は、エスケープ文字と小文字の *d* を頭に付けて、1 ~ 3桁の10進数で定義する。例: `\d42`。
- 8進定数は、エスケープ文字を頭に付けて、1 ~ 3桁の8進数で定義する。例: `\141`。
- 16進定数は、エスケープ文字と小文字の *x* を頭に付けて、1 ~ 2桁の16進数で定義する。例: `\x6a`。

各定数は、シングル・バイト値を表します。2つ以上の10進、8進、または16進定数を連結することによって、マルチバイト値を表すことができます。

---

#### 注意

---

定数を連結してマルチバイト値を構成する場合、その基数(10進、8進、または16進)は同じでなければなりません。変換ソース・ファイルには、Portable Character Set内の文字のみ使用できます。

---

ICONV CONVERT コマンドの説明も参照してください。

---

## エラー

処理中にエラーが発生した場合、ICONV COMPILE は *tablefile* 出力を作成しません。警告メッセージが出された場合は、有効なテーブル・ファイルが作成されます。ただし、警告はユーザ・エラーを示すものなので、返された警告メッセージをチェックする必要があります。

いくつかの ICONV COMPILE エラー・メッセージ、およびその説明を次に示します。

- %ICONV-E-INVFCSTRNG, syntax error in <fromcodeset\_range> definition  
このエラーは、<fromcodeset\_range>シンボルの定義が、決められた構文に従っていない場合に発生します。<fromcodeset\_range>シンボルはエンコード範囲を定義するもので、マルチバイト・コードセットでは必要です。
- %ICONV-E-INVSYNTAX, invalid file syntax  
このエラーは、ソース内の行が、決められた構文に従っていない場合に発生します。

- %ICONV-E-BADTABLE, bad table caused by invalid value for <fromcodeset\_range> definition

このエラーは、コードセット・エンコード範囲に、無効な値が指定された場合に発生します。エンコード範囲は、<fromcodeset\_range>シンボルで定義されます。

---

## 例

1. \$ ICONV COMPILE /LISTING EUCTW\_DECHANYU.ISRC EUCTW\_DECHANYU.ICONV

この例は、EUCTW コードセットを DECHANYU コードセットに変換するための、変換テーブル・ファイルの作成方法を示しています。リスト・ファイル EUCTW\_DECHANYU.LIS には、ソース・ファイルのリスト、およびコンパイラによって生じたエラー・メッセージが含まれます。

---

# ICONV CONVERT

ファイル内の文字を、あるコードセットから別のコードセットへ変換します。変換された文字は、出力ファイルに書き込まれます。

---

## 形式

```
ICONV CONVERT infile outfile
```

---

## パラメータ

*infile*  
必須。

変換する文字を格納したファイルの名前。/FROMCODE 修飾子はこのファイルの文字コードセットを指定します。

*outfile*  
必須。

ICONV CONVERT によって作成されるファイルの名前。/TOCODE 修飾子はこのファイルの文字のコードセットを指定します。

---

## 修飾子

/FROMCODE=*fromcodeset*  
必須。

*infile*の文字のコードセットを指定します。

/TOCODE=*tocodeset*  
必須。

*outfile*の文字コードセットを指定します。

---

## 説明

ICONV CONVERT コマンドは、*infile*内の文字を、/FROMCODE 修飾子で指定されるコードセットから、/TOCODE 修飾子で指定されるコードセットへ変換します。変換されたファイルは、*outfile*に書き込まれます。

変換は、次の2種類の方法のいずれかで行われます。

- 変換テーブル・ファイルを使用して、変換文字を検索する。これがデフォルトの方法であり、次の変換テーブル・ファイルの命名規則が適用される。変換テーブルは、DCL コマンドの ICONV COMPILE により作成される。
- 必要な変換を行う共用イメージ・ファイルを使用する。この方法は、テーブルを使用した変換が便利でない場合(たとえば、アルゴリズムの大きさに対して仮想アドレス空間が大きい場合)にも使用できる。また、テーブルを使用した変換ができない場合(たとえば、ISO2022 のような状態依存のコーディングの場合)にも使用できる。

コンバータのファイルの命名規則は次のとおりです。

fromcodeset\_tocodeset.iconv

---

**注意**

---

変換ファイルをシステムに追加する場合には、同じファイル命名規則を使用しなければなりません。

---

ICONV CONVERT は、現在のディレクトリでコンバータ・ファイルを検索します。現在のディレクトリ内でこのファイルが見つからなかった場合は、論理名 SYS\$I18N\_ICONV によって定義されるシステム・ディレクトリを検索します。

---

**例**

1. \$ ICONV CONVERT /FROMCODE=EUCTW /TOCODE=DECHANYU -  
\_ \$ FROMFILE.DAT TOFILE.DAT

この例は、EUCTW 文字から DECHANYU 文字への変換を示しています。ファイル FROMFILE.DAT 内の EUCTW 文字は、対応する DECHANYU 文字へ変換されます。変換された文字は、ファイル TOFILE.DAT に格納されます。

---

# LOCALE COMPILE

ロケール・ソース・ファイルを、バイナリのロケール・ファイルに変換します。バイナリのロケール・ファイルは、国際化の論理名の設定に依存する、ユーティリティおよびCルーチンによって使用されます。

---

## 形式

LOCALE COMPILE sourcefile

---

## パラメータ

sourcefile  
必須。

ロケールの各カテゴリを定義するロケール・ソース・ファイルの名前。ソース・ファイルのデフォルトのファイル・タイプは.LSRCです。ロケール・ソース・ファイルの形式の定義については、第2章を参照してください。

---

## 修飾子

/CHARACTER\_DEFINITIONS=filename  
/NOCHARACTER\_DEFINITIONS  
オプション。デフォルトの設定：/NOCHARACTER\_DEFINITIONS

ロケール用の文字セット記述ファイル(charmap)を指定します。このファイルは、文字を実際の文字エンコードにマップします。

charmapが指定されない場合には、ロケール・ソース・ファイルの中で、シンボル名(照合シンボル・キーワードで定義された照合シンボルを除く)を使用できません。

charmapファイル形式の定義については、第3章を参照してください。charmapのデフォルトのファイル・タイプは.CMAPです。

/DISPLAY=[NO]HOLE  
オプション。デフォルトの設定：/NOHOLE

中国に関するロケールおよびターミナルとともに使用して、4バイト文字がターミナル・ディスプレイ上で4つの印字位置(桁)を占めることを指定します。デフォルトの値(/DISPLAY=NOHOLE)は、4バイト文字が2つの印字位置を占めることを指定します。

`/IGNORE=WARNINGS`  
`/NOIGNORE`  
 オプション。デフォルトの設定：`/NOIGNORE`

LOCALE COMPILE が警告メッセージを出した場合でも，出力ファイルを作成することを指定します。警告メッセージは，結果として出力されたロケール・ファイルを使用する前に，修正が必要なユーザ・エラーを示していることがあるので，`/IGNORE` キーワードは十分に注意して使用してください。

`/LISTING[=filename]`  
`/NOLISTING`  
 オプション。バッチの場合のデフォルトの設定：`/LISTING`。会話型の場合のデフォルトの設定：`/NOLISTING`。

リスト・ファイルの名前を指定します。`/SHOW` 修飾子は，リスト・ファイルに含まれる情報の内容を制御します。ファイル名が省略された場合，デフォルトのファイル名は `sourcefile.LIS` になります。

`/OUTPUT=[filename]`  
`/NOOUTPUT`  
 オプション。デフォルトの設定：`/OUTPUT=sourcefile.LOCALE`

出力ファイルの名前を指定します。`/OUTPUT` 修飾子の指定が省略された場合，デフォルトの出力ファイル名は `sourcefile.LOCALE` になります。公用のロケールは，論理名 `SYS$I18N_LOCALE` で定義されるディレクトリに保存されます。出力ファイルが別の場所にある場合，ロケールはプライベートです。

`/NOOUTPUT` が指定された場合，コンパイルが正常に終了しても，コンパイラは出力ファイルを作成しません。

`/SHOW[=(keyword[,...])]`  
 オプション。デフォルトの設定：`/SHOW=(SOURCE,TERMINAL)`

`/SHOW` は `/LISTING` とともに使用して，リスト・ファイルに含まれる情報の内容を制御します。次のキーワードを指定できます。

| キーワード                     | 説明                                                       |
|---------------------------|----------------------------------------------------------|
| ALL                       | すべての情報を含む。                                               |
| BRIEF                     | シンボル・テーブルの要約を含む。                                         |
| [NO]CHARACTER_DEFINITIONS | charmap ファイルを含む，または含まない。                                 |
| NONE                      | 情報をいっさい出力しない。NONE が指定された場合，リスト・ファイルは，発生したエラー・メッセージだけを含む。 |
| [NO]SOURCE                | ソース・ファイルのリストを含む，または含まない。                                 |
| [NO]STATISTICS            | コンパイラ性能情報を含む，または含まない。                                    |
| [NO]SYMBOLS               | charmap シンボル・テーブルのリストを含む，または含まない。                        |

| キーワード        | 説明                      |
|--------------|-------------------------|
| [NO]TERMINAL | ターミナルにコンパイラ・メッセージを表示する。 |

---

## 説明

弊社が提供するロケールに加えて、新たなロケールをシステムに追加するには、LOCALE COMPILE コマンドを使用します。ロケールをコンパイルするのに、LOCALE COMPILE は次の 2 つのファイルを必要とします。

- ロケール用の文字セットを定義する charmap ファイル。charmap ファイルを指定しなければ、ロケール・ソース・ファイルでシンボル名を指定できない。この場合、処理されるカテゴリに応じて、LOCALE COMPILE はエラーまたは警告メッセージを出し、出力ファイルは作成されない (/IGNORE 修飾子も参照)。
- ロケール・ソース・ファイル。これは、LC\_CTYPE、LC\_COLLATE、LC\_MESSAGES、LC\_MONETARY、LC\_NUMERIC、LC\_TIME というロケール・カテゴリを、1 つ以上記述する。

---

## エラー

次に示すエラー・メッセージは、LOCALE COMPILE コマンドに関係があります。

- %LOCALE-E-CASEALRDY, case conversion already exists for 'character'  
ここで、character は、コードセットからの文字です。このエラーは、ロケール・コンパイラが LC\_CTYPE カテゴリを処理するときに発生します。これは、character に対して、複数の大文字/小文字変換が指定されたことを示しています。
- %LOCALE-E-PREOFMAP, premature end of file in charmap file  
このエラーは、charmap ファイルに END CHARMAP 文が指定されていない場合に発生します。
- %LOCALE-E-PREOFSRC, premature end of file in source file  
このエラーは、ロケール・ソース・ファイルで、END 文にエラーがある場合に発生します。
- %LOCALE-F-NOADDSYM, failed to add symbol to symbol table  
このエラーは、コンパイルを終了するのに十分なメモリがない場合に発生することがあります。プロセスで使用できるメモリ量を確認してください。

- %LOCALE-F-NOINITSYM, failed to initialize symbol table

このエラーは、コンパイルを終了するのに十分なメモリがない場合に発生することがあります。プロセスで使用できるメモリ量を確認してください。

---

### 例

1. `$ LOCALE COMPILE EN_GB_ISO8859-1 /CHARACTER_DEFINITIONS=ISO8859-1 -  
_$_ /LIST /SHOW=(CHARACTER_DEFINITIONS,SYMBOLS,STATISTICS)`

この例は、charmap ファイル ISO8859-1.CMAP を使用して、ソース・ファイル EN\_GB\_ISO8859-1.LSRC から、ロケール・ファイル EN\_GB\_ISO8859-1.LOCALE を作成する方法を示しています。このロケール・ファイルを使用するには、これを SYS\$I18N\_LOCALE ディレクトリにコピーし、LANG 論理名を "EN\_GB.ISO8859-1" に設定します。リスト・ファイルには、charmap ファイルのリスト、シンボル・テーブル、性能情報、およびコンパイラで生じたエラー・メッセージが含まれます。



---

# LOCALE LOAD

指定されたロケール名を、共用の読み込み専用グローバル・データとして、システムのメモリにロードします。

---

## 形式

LOCALE LOAD locale\_identifier

---

## パラメータ

locale\_identifier  
必須。

ロードするロケールを識別する文字列。次のいずれかを指定します。

- 公用のロケール名

これは、次の形式で指定します。

language\_country.codeset[@modifier]

LOCALE LOAD は、論理名 SYS\$I18N\_LOCALE で定義される位置で、公用のロケール・バイナリ・ファイルを検索します。デフォルトのファイル・タイプは .LOCALE です。指定された名前で使用されているピリオド(.)およびアットマーク(@)は、アンダスコア(\_)文字に置き換えられます。

たとえば、指定された名前が "zh\_CN.dechanzi@radical" の場合、LOCALE LOAD は、次のバイナリロケール・ファイルを検索します。

SYS\$I18N\_LOCALE:ZH\_CN\_DECHANZI\_RADICAL.LOCALE

- ファイルの名前

バイナリのロケール・ファイルを指定します。ファイル指定として有効であれば、どのようなファイル指定でもかまいません。デバイスまたはディレクトリが指定されていない場合、LOCALE LOAD は、現在の呼び出し側のデバイスおよびディレクトリを、デフォルトの設定として最初に適用します。これでファイルが見つからない場合、SYS\$I18N\_LOCALE 論理名で定義されるデバイスおよびディレクトリが、デフォルトの設定として使用されます。なお、デフォルトのファイル・タイプは .LOCALE です。

ワイルドカードは使用できません。また、バイナリのロケール・ファイルは、遠隔ノードに置くことはできません。

---

修飾子

なし

---

説明

LOCALE LOAD コマンドは、指定されたロケール名を、いくつかの共用読み込み専用グローバル・セクションとして、システムのメモリにロードします。ロードされたロケールにアクセスするプロセスはすべて、ロケールの1つのコピーを使用するため、システム・メモリ全体に対する需要が抑制されます。

この DCL コマンドは特権付きであり、通常はシステム管理者が実行します。この場合、次の特権が必要です。

- SYSGBL
- PRMGBL

---

例

1. `$ LOCALE LOAD JA_JP_DECKANJI`

これは、JA\_JP\_DECKANJI ロケールをロードする例です。

---

# LOCALE UNLOAD

指定されたロケール名を、システムのメモリからアンロードします。

---

## 形式

LOCALE UNLOAD locale

---

## パラメータ

locale  
必須。

アンロードするロケールを識別する文字列。このパラメータの指定形式については、LOCALE LOAD コマンドの説明を参照してください。

---

## 修飾子

なし

---

## 説明

LOCALE UNLOAD コマンドは、指定されたロケール名を、システムのメモリからアンロードします。UNLOAD コマンドが入力されたときに、プロセスがそのロケールにアクセスしている場合には、このアクセスが終了した後で、グローバル・セクションが削除されます。

この DCL コマンドは特権付きであり、通常はシステム管理者が実行します。この場合、次の特権が必要です。

- SYSGBL
- PRMGBL

---

### 注意

LOCALE LOAD コマンドでロードされたロケール・ファイルのみ、アンロードできます。

---

### 例

1. `$ LOCALE UNLOAD JA_JP_DECKANJI`

これは、JA\_JP\_DECKANJI ロケールをアンロードする例です。

---

# LOCALE SHOW CHARACTER\_DEFINITIONS

文字セット記述ファイル(charmap)をリストします。

---

## 形式

LOCALE SHOW CHARACTER\_DEFINITIONS

---

## パラメータ

なし

---

## 修飾子

なし

---

## 説明

LOCALE SHOW CHARACTER\_DEFINITIONS コマンドは、論理名 SYS\$I18N\_LOCALE で定義される公用ディレクトリ内の、文字セット記述ファイル(charmap)をすべてリストします。

charmap は、シンボル名、およびコード化文字セット内の文字の値を定義します。また、ロケールをコンパイルするとき、LOCALE\_COMPILE コマンドによって使用されます。charmap ファイルのファイル・タイプは.CMAP です。

---

## 例

1. \$ LOCALE SHOW CHARACTER\_DEFINITIONS

## LOCALE SHOW CHARACTER\_DEFINITIONS

```
[SYS$I18N.LOCALES.SYSTEM] DECHANYU
[SYS$I18N.LOCALES.SYSTEM] DECHANZI
[SYS$I18N.LOCALES.SYSTEM] DECKANJI
[SYS$I18N.LOCALES.SYSTEM] DECKOREAN
[SYS$I18N.LOCALES.SYSTEM] EUCJP
[SYS$I18N.LOCALES.SYSTEM] EUCTW
[SYS$I18N.LOCALES.SYSTEM] ISO8859-1
[SYS$I18N.LOCALES.SYSTEM] ISO8859-2
[SYS$I18N.LOCALES.SYSTEM] ISO8859-3
[SYS$I18N.LOCALES.SYSTEM] ISO8859-4
[SYS$I18N.LOCALES.SYSTEM] ISO8859-5
[SYS$I18N.LOCALES.SYSTEM] ISO8859-7
[SYS$I18N.LOCALES.SYSTEM] ISO8859-8
[SYS$I18N.LOCALES.SYSTEM] ISO8859-9
[SYS$I18N.LOCALES.SYSTEM] MITACTELEX
[SYS$I18N.LOCALES.SYSTEM] SDECKANJI
[SYS$I18N.LOCALES.SYSTEM] SJIS
```

この例は、SYS\$I18N\_LOCALE ディレクトリにあるすべての charmap ファイルを示しています。

---

# LOCALE SHOW CURRENT

各種の国際化論理名で定義された、現在の国際化環境の要約を表示します。

---

## 形式

LOCALE SHOW [CURRENT]

---

## パラメータ

なし

---

## 修飾子

なし

---

## 説明

LOCALE SHOW CURRENT コマンドは、ロケール・カテゴリごとの設定、および環境変数 LC\_ALL と LANG の値をリストします。

CURRENT キーワードはデフォルトの値のため、省略しても構いません。カテゴリを定義する論理名は、カテゴリと同じ名前です。たとえば、LC\_MESSAGES という論理名は、LC\_MESSAGES カテゴリの設定を定義します。表 4-3 は、ロケール・カテゴリについて説明しています。

表 4-3 ロケール・カテゴリ

| カテゴリ        | 説明                                        |
|-------------|-------------------------------------------|
| LC_COLLATE  | 照合シーケンスについての情報                            |
| LC_CTYPE    | 文字分類についての情報                               |
| LC_MESSAGES | プログラム・メッセージの言語、および yes/no プロンプトの形式についての情報 |
| LC_MONETARY | 通貨編集形式についての情報                             |
| LC_NUMERIC  | 数値編集形式についての情報                             |
| LC_TIME     | 日付/時刻についての情報                              |

ロケール・カテゴリは、次に示す順序で論理名を検索することによって、それぞれ定義されます。検索は、論理名が見つかるまで行われます。見つかった論理名が、有効

なロケール・ファイルを表していない場合、LOCALE SHOW は、すべてのカテゴリに対して、文字列 "C" を表示します。

1. LC\_ALL
2. テーブルの中で指定されているカテゴリに対応する論理名 (たとえば、有効なロケール・カテゴリとして LC\_NUMERIC が指定されている場合、LOCALE SHOW CURRENT コマンドは、カテゴリ名、およびこれが定義するロケール名を表示する)。
3. LANG
4. SYS\$LC\_ALL
5. SYS\$\*論理名で指定される、ロケール・カテゴリのシステムのデフォルトの値。たとえば、カテゴリ LC\_NUMERIC のデフォルトの値は、SYS\$LC\_NUMERIC 論理名で定義される。
6. SYS\$LANG

システム管理者は、サイト別システム・スタートアップ・ファイルで SYS\$\*論理を定義して、デフォルトのロケールを設定することもできます。定義が指定されない場合、プログラムは、組み込みのロケール "C" を使用して処理を行います。この場合、LOCALE SHOW CURRENT コマンドは、現在のロケール・カテゴリに対して、文字列 "C" を表示します。

---

## エラー

環境を定義する論理名が誤って定義されていても、警告メッセージが表示されることはありません。ただし、HP Cランタイム・ライブラリ・ルーチン `setlocale` を使用するアプリケーションが受け取るであろう実際の国際化環境が正確にリストされます (たとえば、下の例で `SPECIAL.LOCALE` ファイルが存在しない場合、`LC_MESSAGES` カテゴリの表示は `LC_MESSAGES="C"` となります)。

---

## 例

1. 

```
$ DEFINE LC_COLLATE EN_US.ISO8859-1 ! NOTE: the collate category in unquoted
$ DEFINE LANG EN_GB_ISO8859-1
$ DEFINE LC_MESSAGES PRIVATE$DISK:[APPL.LOCALES]SPECIAL.LOCALE
$ LOCALE SHOW CURRENT
```



```
LANG="EN_GB_ISO8859-1"
LC_CTYPE="EN_GB_ISO8859-1"
LC_COLLATE=EN_US_ISO8859-1
LC_TIME="EN_GB_ISO8859-1"
LC_NUMERIC="EN_GB_ISO8859-1"
LC_MONETARY="EN_GB_ISO8859-1"
LC_MESSAGES=PRIVATE$DISK: [APPL.LOCALES] SPECIAL.LOCALE;1
LC_ALL=
```

この例は、LC\_COLLATEとLC\_MESSAGESを除くすべてのロケール・カテゴリのデフォルトの値を、EN\_GB.ISO8859-1 という同じロケールに設定するプロセスを示しています。二重引用符で囲まれた設定は、その内容が、LANG、LC\_ALL、SYS\$LC\_ALL、SYS\$LANG のいずれかの論理名の設定に含まれていることを表します。これに対して、二重引用符で囲まれていない設定は、そのカテゴリに対する論理名が、国際化環境を定義していることを表します。この例は、ロケール・カテゴリが完全なファイル名で指定された場合、完全なファイル名が表示されることも示しています。

---

# LOCALE SHOW PUBLIC

システム上の公用のロケールをすべてリストします。

---

## 形式

LOCALE SHOW PUBLIC

---

## パラメータ

なし

---

## 修飾子

なし

---

## 説明

LOCALE SHOW PUBLIC コマンドは、システム上の公用のロケールをすべてリストします。公用のロケールのセットには、HP Cランタイム・ライブラリで提供される、システム組み込みのロケールに加えて、論理名 SYS\$I18N\_LOCALE で定義されるディレクトリにあるロケールがすべて含まれます。

---

## 例

1. \$ LOCALE SHOW PUBLIC

```
C (Built-in)
POSIX (Built-in)
[SYS$I18N.LOCALES.SYSTEM]EN_GB_ISO8859_1
[SYS$I18N.LOCALES.SYSTEM]EN_US_ISO8859_1
[SYS$I18N.LOCALES.SYSTEM]FR_CA_ISO8859_1
[SYS$I18N.LOCALES.SYSTEM]GRBAGE_LOCALE (bad file header checksum)
[SYS$I18N.LOCALES.SYSTEM]JA_JP_DECKANJI (Permanently Loaded)
```

この例は、SYS\$I18N\_LOCALE ディレクトリに3つのロケール・ファイルを持つシステムを示しています。CおよびPOSIXロケールはシステムに組み込まれているため、SYS\$I18N\_LOCALE ディレクトリにはありません。

この例は、公用ディレクトリに不良ファイルまたはロケールでないファイルがある場合の動作、および LOCALE LOAD コマンドでシステムのメモリにロードされたロケール・ファイルがある場合の動作も示しています。

---

## LOCALE SHOW VALUE

現在の国際化環境をもとに、1つ以上のキーワードの値を表示します。

---

### 形式

LOCALE SHOW VALUE name

---

### パラメータ

name

必須。複数の名前を指定できます。

次のいずれかの名前を指定します。

- キーワード
  - キーワードを指定すると、現在のロケールでのそのキーワードの値が表示されます。
  - 値が割り当てられていない整数キーワードの場合は、CHAR\_MAX (127) という値が表示されます。
  - キーワード値にセミコロン、二重引用符、バックスラッシュ、制御文字のいずれかが含まれる場合には、その前にエスケープ文字 (通常はバックスラッシュ) が挿入されます。
- カテゴリ
 

カテゴリを指定すると、そのカテゴリのすべてのキーワードの値が表示されます。

表 4-4 は、指定できるカテゴリとキーワードを示しています。

表 4-4 ロケール・カテゴリとキーワード

| カテゴリ     | キーワード | キーワードの説明 |
|----------|-------|----------|
| LC_CTYPE |       | 文字分類名    |
| LC_TIME  | DAY   | 曜日の完全名   |
|          | ABDAY | 曜日の短縮名   |
|          | MON   | 月の完全名    |
|          | ABMON | 月の短縮名    |

(次ページに続く)

表 4-4 (続き) ロケール・カテゴリとキーワード

| カテゴリ        | キーワード             | キーワードの説明                                                                          |
|-------------|-------------------|-----------------------------------------------------------------------------------|
|             | D_T_FMT           | 日付/時刻形式                                                                           |
|             | D_FMT             | 日付形式                                                                              |
|             | T_FMT             | 時刻形式                                                                              |
|             | T_FMT_AMPM        | 12 時間表示の時刻形式                                                                      |
|             | AM_PM             | 午前 (a.m.) および午後 (p.m.) を表す文字列の表示方法を定義する                                           |
|             | ERA               | ロケールにおける年 (年号) の数えかた, および表示方法を定義する                                                |
|             | ERA_D_FMT         | 年号の日付形式                                                                           |
|             | ERA_D_T_FMT       | 年号の日付/時刻形式                                                                        |
|             | ERA_T_FMT         | 年号の時刻形式                                                                           |
|             | ALT_DIGITS        | 数字の代替シンボルを定義する文字列                                                                 |
| LC_NUMERIC  | DECIMAL_POINT     | 小数点として使用する文字                                                                      |
|             | THOUSANDS_SEP     | 小数点の左側で数字を区切るのに使用する文字                                                             |
|             | GROUPING          | 小数点の左側で文字をどのように区切るかを定義する                                                          |
| LC_MONETARY | INT_CURR_SYMBOL   | 国際通貨記号を表す文字列                                                                      |
|             | CURRENCY_SYMBOL   | ローカル通貨記号として使用される文字列                                                               |
|             | MON_DECIMAL_POINT | 金額を書式化するとき小数点として使用される文字                                                           |
|             | MON_THOUSANDS_SEP | 小数点の左側で数字の位取りに使用する文字                                                              |
|             | POSITIVE_SIGN     | 正の金額を表すのに使用する文字列                                                                  |
|             | NEGATIVE_SIGN     | 負の金額を表すのに使用する文字列                                                                  |
|             | INT_FRAC_DIGITS   | 国際通貨記号を使用して金額を書式化する場合, 小数点の右側に表示される桁数                                             |
|             | FRAC_DIGITS       | ローカル通貨記号を使用して金額を書式化する場合, 小数点の右側に表示される桁数                                           |
|             | P_CS_PRECEDES     | 正の金額値について, ローカル通貨記号が数値の前にくる場合は 1, 数値の後にくる場合は 0 に設定する                              |
|             | N_CS_PRECEDES     | 負の金額値について, ローカル通貨記号が数値の前にくる場合は 1, 数値の後にくる場合は 0 に設定する                              |
|             | P_SEP_BY_SPACE    | 正の金額値について, 通貨記号と値のあいだにスペースがない場合は 0, スペースがある場合は 1, 記号と符号文字列のあいだにスペースがある場合は 2 に設定する |
|             | N_SEP_BY_SPACE    | 負の金額値について, 通貨記号と値のあいだにスペースがない場合は 0, スペースがある場合は 1, 記号と符号文字列のあいだにスペースがある場合は 2 に設定する |
|             | P_SIGN_POSN       | POSITIVE_SIGN文字列の位置を指定するために使用される整数                                                |
|             | N_SIGN_POSN       | NEGATIVE_SIGN文字列の位置を指定するために使用される整数                                                |
|             | MON_GROUPING      | 金額値を編集するときの数字の位取りを定義する                                                            |

(次ページに続く)

表 4-4 (続き) ロケール・カテゴリとキーワード

| カテゴリ        | キーワード   | キーワードの説明              |
|-------------|---------|-----------------------|
| LC_MESSAGES | YESSTR  | 現在のロケールの中で YES を表す文字列 |
|             | NOSTR   | 現在のロケールの中で NO を表す文字列  |
|             | YESEXPR | 現在のロケールの中で肯定応答を表す式    |
|             | NOEXPR  | 現在のロケールの中で否定応答を表す式    |

---

**注意**

---

現在のロケールの設定に影響を及ぼす環境変数が、無効なロケールを指定している場合、ロケール“C”が設定されます。

---

カテゴリの一部として、デフォルトの設定で表示されない他の有効なキーワードには、次のキーワードが含まれます。

- CHARMAP — ロケールが作成されたときに使用された charmap のファイル指定を表示する。
- CODE\_SET\_NAME — charmap ファイルの定義の対象となるコード化文字セット名を定義する。
- MB\_CUR\_MAX — マルチバイト文字内の最大バイト数を定義する。
- MB\_CUR\_MIN — コード化文字セット内の文字の最小バイト数を定義する。

---

## 修飾子

`/CATEGORY`

オプション。デフォルトの設定：カテゴリ名を表示しない。

各キーワードの前にカテゴリ名を表示します。

`/KEYWORD`

オプション。デフォルトの設定：キーワード名を表示しない。

キーワードの値の前にキーワード名を表示します。

---

## 説明

LOCALE SHOW VALUE コマンドは、現在の国際化環境をもとに、1 つ以上のキーワードの値を表示します。

---

エラー

```
%LOCALE-E-NOKEYFND, no keyword keyword-name found
```

*keyword-name*は、有効なキーワードではありません。表 4-4 にリストされているキーワードだけを指定してください。

---

例

1. \$ LOCALE SHOW VALUE NOEXPR

```
"^[nN][[:alpha:]]*"

```

修飾子を指定せずに LOCALE SHOW VALUE を実行すると、NOEXPR 文字列の値が表示されます。

2. \$ LOCALE SHOW VALUE /CATEGORY NOEXPR

```
LC_MESSAGES
"^[nN][[:alpha:]]*"

```

/CATEGORY を指定すると、NOEXPR 文字列の値の前に、カテゴリ名 (LC\_MESSAGES) が表示されます。

3. \$ LOCALE SHOW VALUE /KEYWORD NOEXPR

```
noexpr= "^[nN][[:alpha:]]*"

```

/KEYWORD を指定すると、キーワード値の前に、キーワード名が表示されます。

4. \$ LOCALE SHOW VALUE /KEYWORD /CATEGORY NOEXPR

```
LC_MESSAGES
noexpr= "^[nN][[:alpha:]]*"

```

/KEYWORD および/CATEGORY を指定すると、キーワード値の前に、カテゴリおよびキーワード名が表示されます。

---

## zic

指定されたタイム・ゾーンのソース・ファイルのデータを使用して、タイム・ゾーン変換情報を格納したバイナリ・ファイルを作成します。

---

### 形式

```
zic [-v] ["-L" leapseconds] [-d directory] [-y yearistype] infile
```

---

### パラメータ

infile  
必須。

zic が読み込むソース・ファイル。

---

### 修飾子

-v  
省略可能。

データ・ファイルに指定されている年が、時刻値によって表現可能な年の範囲をこえている場合は通知します。

"-L"  
省略可能。

指定された名前のファイルから、うるう秒情報を読み込みます。このオプションを使用しない場合は、うるう秒情報が出力ファイルに格納されません。

-d  
省略可能。

標準ディレクトリではなく、指定されたディレクトリに時間変換情報ファイルを作成します。

-y  
省略可能。

年のタイプを確認するときに、yearistype ではなく、指定されたコマンド・ファイルを使用します。



---

## 説明

zic コマンドを使用すると、ZIC コンパイラはコマンド行に指定されたファイルからテキストを読み込み、この入力に指定された時間変換情報ファイルを作成できます。

ファイル名が- の場合には、標準入力を読み込まれます。

入力行は複数のフィールドで構成されます。フィールドの間は任意の数のスペース文字で区切ることができます。入力行の先頭と末尾の空白は無視されます。入力行で、引用符で囲まれていないシャープ記号(#)はコメントの先頭を示し、この記号から行の末尾までがコメントであると解釈されます。スペース文字とシャープ文字をフィールドの一部として使用するときは、二重引用符(" ")で囲みます。コメントの後の空白行は無視されます。

空白行以外の行は、次の3種類のいずれかであると解釈されます。

- ルール行 (第 1.2.1 項参照)
- ゾーン行 (第 1.2.2 項参照)
- リンク行 (第 1.2.3 項参照)

---

## 例

1. `$ zic -v "-L" leapseco -d [-] myafrica`

ZIC コンパイラは myafrica というタイム・ゾーンのソース・ファイルをコンパイルします。指定されたパラメータをもとに、ZIC は次の処理を行います。

1. 表現可能な範囲をこえる年を通知します。
2. うるう秒を適用して出力ファイルを作成します。
3. 現在のディレクトリに結果を格納します。

日付/時刻関数についての詳細は、『HP C ランタイム・ライブラリ・リファレンス・マニュアル』を参照してください。



## 付録: 日本語ロケールについて

ここでは、日本語ロケールについて説明します。

### A.1 日本語ロケール・ファイル形式

ロケールの名称は、言語 (language)、地域 (テリトリ, territory)、コードセット (codeset) から構成されます。一般的には次のような形式で表現されます。

```
language_territory.codeset
```

DEC 漢字コードセットを使用した日本語ロケールの場合、言語名が ja、テリトリが JP、コードセット名が deckanji であり、ロケール名は ja\_JP.deckanji となります。

OpenVMS XPG4 ユーティリティで定義している日本語文字コードセットは次の 5 つです。

表 A-1 日本語文字コードセット

|              |                 |
|--------------|-----------------|
| deckanji     | DEC 漢字コード       |
| deckanji2000 | DEC 漢字 2000 コード |
| sdeckanji    | Super DEC 漢字コード |
| eucJP        | 日本語 EUC         |
| SJIS         | Shift JIS       |

OpenVMS は、これらの 4 つの文字コードに対応する日本語ロケール・データ・ファイルを提供します。各ロケールの名称は次の通りです。

| ロケール名              | ファイル名                     | 備考               |
|--------------------|---------------------------|------------------|
| ja_JP.deckanji     | JA_JP_DECKANJI.LOCALE     | DEC 漢字ロケール       |
| ja_JP.deckanji2000 | JA_JP_DECKANJI2000.LOCALE | DEC 漢字 2000 ロケール |
| ja_JP.sdeckanji    | JA_JP_SDECKANJI.LOCALE    | Super DEC 漢字ロケール |
| ja_JP.eucJP        | JA_JP_EUCJP.LOCALE        | 日本語 EUC ロケール     |
| ja_JP.SJIS         | JA_JP_SJIS.LOCALE         | Shift JIS ロケール   |

これらのファイルは、論理名 SYS\$I18N\_LOCALE で指定されるディレクトリに置かれます。

---

## 注意

---

上記の日本語ロケール・データ・ファイルはオプション・セーブセット VMSI18N073 をインストールすることで、システムに提供されます。VMSI18N073 セーブセットのインストールについては、日本語 HP OpenVMS の『インストール・ガイド』および『リリース・ノート』を参照してください。

---

ユーザが日本語ロケールを設定するには、論理名 LANG, LC\_ALL, あるいは LC\_ で始まる各ロケール・カテゴリを示す論理名に、上記ロケール名のいずれかを指定します。たとえば、以下の例では SuperDEC 漢字ロケールを設定しています。

```
$ DEFINE LANG "ja_JP.sdeckanji"
```

システム管理者が SYS\$ で始まる論理名を定義し、システムのデフォルトのロケールを設定することもできます。

---

## A.2 ICONV CONVERT ユーティリティ補足説明 – 日本語文字コード変換テーブル・ファイル

OpenVMS V6.2 では日本語文字コードについて次の変換テーブル・ファイルが SYS\$I18N\_ICONV に提供されます。

| fromcodeset | tocodeset |
|-------------|-----------|
| DECKANJI    | EUCJP     |
| DECKANJI    | SDECKANJI |
| DECKANJI    | SJIS      |
| EUCJP       | DECKANJI  |
| EUCJP       | SDECKANJI |
| EUCJP       | SJIS      |
| SDECKANJI   | DECKANJI  |
| SDECKANJI   | EUCJP     |
| SDECKANJI   | SJIS      |
| SJIS        | DECKANJI  |
| SJIS        | EUCJP     |
| SJIS        | SDECKANJI |

OpenVMS V7.0 では、上記に加えて次の変換テーブル・ファイルが提供されます。

| fromcodeset | tocodeset |
|-------------|-----------|
| DECKANJI    | ISO2022JP |
| ISO2022JP   | DECKANJI  |
| ISO2022JP   | SDECKANJI |
| SDECKANJI   | ISO2022JP |

ICONV コマンドは、状態独立の 1 ~ 4 バイトのコードセットをサポートします。状態依存のコードセットはサポートしません。

**注意**

変換テーブル・ファイルはオプション・セーブセット VMSI18N073 をインストールすることでシステムに提供されます。

日本語 OpenVMS V7.3 では、VMSI18N073 オプション・セーブセットとは独立に次の変換テーブル・ファイルが提供されます。

| fromcodeset  | tocodeset    |
|--------------|--------------|
| DECKANJI2000 | EUCJP        |
| DECKANJI2000 | ISO2022JP    |
| DECKANJI2000 | SJIS         |
| EUCJP        | DECKANJI2000 |
| ISO2022JP    | DECKANJI2000 |
| SJIS         | DECKANJI2000 |



## C

|                       |           |
|-----------------------|-----------|
| charmap(文字セット記述) ファイル | 1-4       |
| 構成要素                  | 3-4       |
| 説明                    | 3-1 ~ 3-6 |

## D

|                 |     |
|-----------------|-----|
| deckanji        | A-1 |
| deckanji2000    | A-1 |
| DEC 漢字 2000 コード | A-1 |
| DEC 漢字コード       | A-1 |

## E

|       |     |
|-------|-----|
| eucJP | A-1 |
|-------|-----|

## G

|             |     |
|-------------|-----|
| GENCAT コマンド | 4-2 |
|-------------|-----|

## I

|            |           |
|------------|-----------|
| ICONV コマンド |           |
| COMPILE    | 1-3, 4-7  |
| CONVERT    | 1-3, 4-13 |

## L

|                            |      |
|----------------------------|------|
| LC_COLLATE ロケール・カテゴリ       | 2-3  |
| LC_CTYPE ロケール・カテゴリ         | 2-7  |
| LC_MESSAGES ロケール・カテゴリ      | 2-11 |
| LC_MONETARY ロケール・カテゴリ      | 2-12 |
| LC_NUMERIC ロケール・カテゴリ       | 2-16 |
| LC_TIME ロケール・カテゴリ          | 2-17 |
| LOCALE コマンド                |      |
| COMPILE                    | 4-15 |
| LOAD                       | 4-19 |
| SHOW CURRENT               | 4-25 |
| SHOW PUBLIC                | 4-28 |
| SHOW VALUE                 | 4-30 |
| UNLOAD                     | 4-21 |
| LOCALE コマンド                |      |
| SHOW CHARACTER_DEFINITIONS | 4-23 |

## P

|                             |     |
|-----------------------------|-----|
| PCS(Portable Character Set) | 3-1 |
| Portable Character Set      |     |
| 「PCS」参照                     |     |

## S

|                 |     |
|-----------------|-----|
| sdeckanji       | A-1 |
| Shift JIS       | A-1 |
| SJIS            | A-1 |
| Super DEC 漢字コード | A-1 |

## X

|                                   |            |
|-----------------------------------|------------|
| X/Open Portability Guide, Issue 4 |            |
| 「XPG4 ローカライゼーション・ユーティリティ」参照       |            |
| XPG4 ローカライゼーション・ユーティリティ           |            |
| ICONV コマンド                        |            |
| COMPILE                           | 1-3, 4-7   |
| CONVERT                           | 4-13       |
| LOCALE コマンド                       |            |
| COMPILE                           | 4-15       |
| LOAD                              | 4-19       |
| SHOW CHARACTER_DEFINITIONS        | 4-23       |
| SHOW CURRENT                      | 4-25       |
| SHOW PUBLIC                       | 4-28       |
| SHOW VALUE                        | 4-30       |
| LOCALE コマンド                       |            |
| UNLOAD                            | 4-21       |
| GENCAT コマンド                       | 4-2        |
| zic コマンド                          | 4-34       |
| コマンド                              | 4-34       |
| 文字マップ(charmap) ファイル               | 3-1 ~ 3-6  |
| ロケール・ファイルの形式                      | 2-1 ~ 2-23 |

## Z

|                                            |           |
|--------------------------------------------|-----------|
| ZIC                                        | 1-5 ~ 1-8 |
| 「Zone Information Compiler(ZIC) ユーティリティ」参照 |           |
| zic コマンド                                   | 4-34      |
| ZIC ゾーン行                                   | 1-7 ~ 1-8 |
| ZIC リンク行                                   | 1-8       |
| ZIC ルール行                                   | 1-5 ~ 1-7 |

Zone Information Compiler(ZIC) ユーティリティ

|           |      |
|-----------|------|
| zic コマンド  | 1-5  |
| ZIC コンパイラ | 1-5  |
| 概要        | 1-1  |
| 修飾子       | 4-34 |
| パラメータ     | 4-34 |

## カ

---

### 概要

|                    |     |
|--------------------|-----|
| ZIC ユーティリティ        | 1-1 |
| ローカライゼーション・ユーティリティ | 1-1 |

## コ

---

### コマンド

|         |            |
|---------|------------|
| 構文の説明   | 4-1 ~ 4-35 |
| コマンドの説明 | 4-1 ~ 4-35 |

## 二

---

|                |     |
|----------------|-----|
| 日本語 EUC        | A-1 |
| 日本語文字コードセット    | A-1 |
| 日本語文字コード変換テーブル | A-2 |
| 日本語ロケール・ファイル形式 | A-1 |

## モ

---

|                  |  |
|------------------|--|
| 文字セット記述ファイル      |  |
| 「charmap ファイル」参照 |  |

## ロ

---

|                    |            |
|--------------------|------------|
| ローカライゼーション・ユーティリティ |            |
| 概要                 | 1-1        |
| ロケール・カテゴリ          | 1-4        |
| ロケール・ファイルの形式       |            |
| 説明                 | 2-1 ~ 2-23 |
| ロケール・カテゴリ          | 2-1, 2-2   |







HP OpenVMS  
HP C 国際化ユーティリティ・リファレンス・マニュアル

---

2007 年 12 月 発行

日本ヒューレット・パカード株式会社

〒 102-0076 東京都千代田区五番町 7 番地

電話 (03)3512-5700 (大代表)

