



SAS[®] Viya[™] 3.1 各国語サポート: リファレンスガイド

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. SAS® *Viya*™ 3.1 各国語サポート: リファレンスガイド. Cary, NC: SAS Institute Inc.

SAS® *Viya*™ 3.1 各国語サポート: リファレンスガイド

Copyright © 2016, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

October 2016

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

3.1-P1:nlsref

目次

1部 各国語サポートの概念 1

1章・各国語サポート	3
各国語サポートの概要	3
地域化と国際化の定義	4
2章・各国語サポート関連のロケール	5
各国語サポート関連のロケール概念の概要	5
ロケールの指定	6
3章・各国語サポート関連のエンコーディング	9
概要: 各国語サポート関連のエンコーディング	9
エンコーディングとトランスコーディングの違い	10
よく用いられるエンコーディング方法	10
各国語サポートエンコーディング関連の標準組織	13
EBCDIC エンコーディングの制約	13
照合順序	14
SAS データセットのエンコーディングの確認	16
SAS セッションのエンコーディングの動作	17
4章・各国語サポート関連のトランスコーディング	19
トランスコーディングの概要	19
トランスコーディングが必要となる場合	20
SAS データのトランスコーディングを指定する SAS オプション	20
トランスコーディングの留意点	21
互換性があるエンコーディングと互換性がないエンコーディング	21
トランスコーディングの防止	22
CVP エンジンを使用した文字データ切り捨てへの対応	23
5章・タイムゾーン	25
SAS タイムゾーン ID の概要	25
SAS でのタイムゾーンの指定	25

2部 各国語サポート関連の自動呼び出しマクロ 33

6章・各国語サポート関連の自動呼び出しマクロのディクショナリ	35
カテゴリ別の自動呼び出しマクロエントリ	35
ディクショナリ	35

3部 各国語サポート関連のデータセットオプション 39

7章・各国語サポート関連のデータセットオプションのディクショナリ	41
カテゴリ別のデータセットオプション	41
ディクショナリ	41

4部	各国語サポート関連の出力形式	45
8章	概要:各国語サポート関連の出力形式	47
	国際化対応の日付、日時出力形式	47
	通貨表現	53
	日付と時間のデフォルト幅の例外	60
9章	各国語サポート関連の出力形式のディクショナリ	63
	各国語サポート関連の出力形式のカテゴリ	66
	ディクショナリ	82
5部	各国語サポート関連の関数	277
10章	SAS 文字列関数の国際化の互換性	279
	SAS 文字列関数の国際化の互換性	279
11章	関数での VARCHAR サポート	297
	文字列関数の VARCHAR データタイプ	297
12章	各国語サポート関連の関数のディクショナリ	301
	カテゴリ別の関数	302
	ディクショナリ	307
6部	各国語サポート関連の入力形式	403
13章	各国語サポート関連の入力形式のディクショナリ	405
	カテゴリ別の入力形式	407
	ディクショナリ	414
7部	各国語サポート関連のマクロ関数	531
14章	各国語サポート関連のマクロ関数のディクショナリ	533
	カテゴリ別のマクロ	533
	ディクショナリ	534
8部	各国語サポート関連のシステムオプション	543
15章	各国語サポート関連のシステムオプションのディクショナリ	545
	カテゴリ別のシステムオプションエントリ	545
	ディクショナリ	546
9部	各国語サポート関連のコマンド、ステートメント、プロシジャオプション	567

16 章・各国語サポート関連のコマンド、ステートメントおよびプロシジャオプションのディクショナリ	569
カテゴリ別の各国語サポート関連のコマンド、ステートメント、およびプロシジャオプション	569
ディクショナリ	570
10 部 ロケールとトランスコーディングの値	597
17 章・LOCALE=システムオプションの値	599
PAPERSIZE と DFLANG オプションの LOCALE=の値	599
18 章・SAS 言語要素中のエンコーディング値	613
エンコーディング値を使用する SAS 言語要素の概要	613
データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値	613
11 部 付録	625
付録 1・タイムゾーン ID とタイムゾーン名	627
エリア: Africa (アフリカ)	627
エリア: America (アメリカ-北、中央、および南)	629
エリア: Antarctica (南極)	639
エリア: Asia (アジア)	639
エリア: Atlantic (大西洋)	643
エリア: Australia (オーストラリア)	644
エリア: その他	646
エリア: Europe (ヨーロッパ)	651
エリア: Pacific (太平洋)	656
推奨資料	659
用語集	661
キーワード	667

1 部

各国語サポートの概念

1 章	各国語サポート	3
2 章	各国語サポート関連のロケール	5
3 章	各国語サポート関連のエンコーディング	9
4 章	各国語サポート関連のトランスコーディング	19
5 章	タイムゾーン	25

1 章

各国語サポート

各国語サポートの概要	3
地域化と国際化の定義	4

各国語サポートの概要

各国語サポートは、ソフトウェアを各国の市場向けに正しく動作させるために必要な機能の集まりです。SAS には、地域の言語規則に準拠した SAS アプリケーションを確実に作成できるようにする、各国語サポート機能が組み込まれています。通常、英語で作成されるソフトウェアは、英語を話し、米国内で使用される規則に従ってフォーマットされたデータを使用するユーザーにとって使いやすくできています。ただし、これらのソフトウェアは、各国語サポートがなければ、世界の他の地域のユーザーにとっては使いにくい製品となります。SAS の各国語サポートでは、アジアやヨーロッパなどの地域のユーザーが、自国の言語および環境の下でデータを正常に処理できます。

各国語サポートは、マシン間を移動するデータに適用されます。たとえば、各国語サポートにより、ターゲットマシンで使用される正しい形式にデータを確実に変換できます。

テキスト文字列の操作は、言語および地域の SAS 設定の影響を受けます。このアクションにより、文字の大文字化と小文字化、文字の分類、データのスキャンなどの操作による結果を修正できます。SAS には、特定の国または国のグループに固有の文字である各国語文字を適正に表示および出力するための機能が組み込まれています。

各国語サポートを組み込んだソフトウェアアプリケーションにより、ソフトウェアの機能に関して、言語固有または文化固有の規則に依存することを回避できます。いくつかの例を示します。

- 文字列の操作
- 文字の分類
- 文字の比較ルール
- コードセット
- 日時出力形式
- ユーザーインターフェイス
- メッセージテキスト言語

- 数値および通貨の出力形式
- 並べ替え順序

地域化と国際化の定義

*国際化*は、1つの言語またはロケールに基づくという前提なしで行われる、ソフトウェア製品の設計処理で、地域化を容易にします。国際化により、文字列の並べ替えと日付、時間、数、通貨のフォーマティングのルールを含む、国際的な規則がサポートされます。また、一貫したユーザーエクスペリエンスを製品のどの言語版でも可能にします。国際化の略称はI18Nです。

アプリケーションのロジックが文化的規則(特定の地域の通貨や数値の出力形式など)をサポートするとしても、地域化されたバージョンのソフトウェアのみが現地語のユーザーインターフェイスとシステムメッセージを提供します。

*地域化*は、特定の対象となる環境や市場の言語、文化、その他の要件に製品を適合させる処理で、これによりユーザーは、その製品の使用時に、現地の言語と規則で作成された結果を参照できます。ユーザーインターフェイス、システムメッセージ、ドキュメントの翻訳も地域化に含まれます。

SAS 各国語サポート(NLS)の機能を使用して、SAS アプリケーションを地域化や国際化することができます。詳細については、[“SAS 文字列関数の国際化の互換性” \(279 ページ\)](#)を参照してください。

2 章

各国語サポート関連のロケール

各国語サポート関連のロケール概念の概要	5
ロケールの指定	6
SAS 起動時のロケールの指定法	6
SAS セッション中のロケールの指定法	7
言語切り替え	7

各国語サポート関連のロケール概念の概要

ロケールには、各地域の言語、データのフォーマティングなどの地域規則および文化が反映されています。この地域規則には、その国や地域の日付、時刻、数字のフォーマティングや通貨記号が含まれます。ページ順、用紙サイズ、住所、電話番号などもロケールに含まれます。

日付には、文化圏で受け入れられた地域規則に応じて、多くの表示形式があります。月は、数字または名前で表記される場合があります。名前は、完全に入力するか、略称が使用される場合があります。月、日および年の順序は、ロケールによって異なる可能性があります。

たとえば、"2016 年の 10 月の 3 日目"はロケールごとにさまざまな方法で表示されます。

ドイツ
03.10.16

イタリア
3/10/16

米国
10/03/16

言語はロケールの一部ですが、いずれか 1 つのロケールに固有ではありません。たとえばポルトガル語は、ポルトガルだけでなくブラジルでも話されていますが、それぞれの文化は異なります。ブラジルとポルトガルでは、データの表示形式に類似点があります。数字を表示するときに、整数部と小数部を区切るにはカンマ(,)を使用し、基数文字の左側にある桁を区切るにはドット(.)を使用します。ただし、2 つの異なるロケールで使用される通貨記号など、大きな違いもあります。ポルトガルではユーロを使っており、ユーロ記号(€)が必要です。一方、ブラジルではリアルを使っており、2 文字の通貨記号である R\$で表します。

また、1つの国で複数の公用語が使用されている場合もあります。カナダの公用語は英語とフランス語の2つです。LOCALE=システムオプションには、English_Canada および French_Canada という2つの値を指定できます。

通貨などの数値にもさまざまな表示形式があります。たとえば、小数点や基点文字をドット(.)で表す地域もあれば、カンマ(,)で表す地域もあります。3桁ごとの区切りは、地域によって、ドット、カンマ、さらには空白などさまざまです。貨幣の表記規則もロケールごとに異なります。たとえば、金額にドル記号や円記号を付けることもあります。

用紙サイズおよび測定単位もロケールの考慮事項に含まれます。標準の用紙サイズには、レター(8-1/2×11 インチ用紙)および A4 (210×297 ミリメートル用紙)があります。レター用紙サイズは、主に英語圏の国々で使用されています。A4はその他のほとんどのロケールで使用されています。ほとんどのロケールではセンチメートルを使用していますが、一部のロケールではインチを使用します。

時間は、一部の英語圏の国または地域では12時間表記を使用して表示され、他の英語圏では24時間表記を使用して時間値が表示されています。

ロケールの指定

SAS 起動時のローケルの指定法

LOCALE=システムオプションを使用すると、SAS 起動時に SAS セッションのロケールを指定できます。LOCALE=によって、次の SAS システムオプションも暗黙的に設定されます。

- DATESTYLE=
- DFLANG=
- ENCODING=
- LOCALEDATA
- MAPEBCDICTOASCII
- ODSLANGCHG
- PAPERSIZE=
- RSASIoTTRANSERROR
- TIMEZONE
- URLENCODING

注: ロケールは、POSIX 命名規則を使用しても指定できます。たとえば、POSIX の en_US は、SAS 値の English_UnitedStates に相当します。

LOCALE=オプションのデフォルト値は、各動作環境で同じです。詳細については、“[PAPERSIZE と DFLANG オプションの LOCALE=の値](#)” (599 ページ)を参照してください。

LOCALE=の値として English_UnitedStates を指定すると、SAS の起動時、次のオプションに指定されたデフォルト値が暗黙的に設定されます。

- **DATESTYLE=MDY**
- **DFLANG=English**
- **ENCODING=UTF-8**

- **PAPERSIZE=Letter**

起動時に、明示的に設定されたシステムオプションは暗黙的に設定されたオプションより優先されます。

起動時に、明示的に設定された PAPERSIZE=A4 は、LOCALE=オプションを用いて暗黙的に設定された PAPERSIZE=オプションより優先されます。詳細については、“[PAGESIZE= System Option](#)” (*SAS Viya System Options: Reference*) を参照してください。

SAS セッション中のロケールの指定法

LOCALE=システムオプションを使用すると、SAS セッション中に SAS セッションのロケールを指定できます。ただし、変更された LOCALE=の値を反映して暗黙的に変更されるのは、次のシステムオプションの値のみです。

- DFLANG=
- PAPERSIZE=

注: PAPERSIZE=オプションが影響を受けるのは、これを LOCALE に設定した場合に限ります。詳細については、“[PAGESIZE= System Option](#)” (*SAS Viya System Options: Reference*) を参照してください。

ENCODING=システムオプションの値が、変更された LOCALE=の値を反映して暗黙的に変更されることはありません。

LOCALE=オプションに Italian_Italy という値を割り当てると、次のオプションが、変更された LOCALE=システムオプションの値を反映して、SAS セッション中に暗黙的に再設定されます。

- DFLANG=Italian
- PAPERSIZE=A4

言語切り替え

SAS メッセージは、起動中に SAS 構成ファイルの設定で指定された言語で表示されます。言語切り替え機能を使用して別の言語で SAS メッセージを表示できます。言語切り替え機能には、ODSLANGCHG システムオプションを使用してアクセスできます。ODSLANGCHG が有効である場合、LOCALE システムオプションの値によって、プロシジャの出力、ユーザーインターフェイス要素および ODS フォントの言語が決まります。ODSLANGCHG が無効な場合、メッセージは起動中に設定された言語で表示されます。この機能はどのサーバーでも、メッセージを構成する文字がサポート対象のエンコーディングである限り、使用可能です。詳細については、“[ODSLANGCHG システムオプション](#)” (557 ページ) を参照してください。SAS 起動時に LOGLANGCHG オプションまたは ODSLANGCHG オプションを設定する際に、LSWLANG システムオプションにより、言語切り替え機能で使用する言語を指定します。LSWLANG システムオプションは、LOCALE システムオプションよりも優先度が高くなっています。LSWLANG オプションの値が有効な SAS 言語に指定された場合、SAS ログ出力は、その値により制御されます。それ以外の場合、SAS ログの言語は LOCALE=オプションにより決定されます。

LOGLANGCHG システムオプションは、SAS ログメッセージに関して言語切り替えを行うかどうかを制御します。このオプションは、SAS ログ出力におけるメッセージ切り替えの言語を制御します。LOGLANGCHG を指定した場合、SAS ログの言語は、LSWLANG オプションまたは LOCALE=により決定されます。詳細に

については、[“LOGLANGCHG システムオプション” \(552 ページ\)](#)を参照してください。

LSWLANG オプションは、LOGLANGCHG または ODSLANGCHG が有効である場合に、メッセージの言語を指定します。LSWLANG の値が LOCALE に設定されている場合、LOCALE=オプションにより、切り替え用の言語が決定されます。デフォルトは **LSWLANG=LOCALE** です。詳細については、[“LSWLANG システムオプション” \(554 ページ\)](#)を参照してください。

LOGLANGENG はトグルオプションであり、LOGLANGCHG および LSWLANG をオーバーライドし、これらの値をそれぞれ **LOGLANGCHG=ON** および **LSWLANG=EN** に設定します。その結果、LOG 出力は英語になります。NL 出力形式は変わりません。このオプションにより、ODSLANGCHG の設定が変更されます。**ODSLANGCHG=OFF** の場合、ODS 出力のシステムメッセージ言語は、SAS の構成によって決まります。**ODSLANGCHG=ON** の場合、LSWLANG の値が English に設定されるため、すべてのメッセージが英語になります。詳細については、[“LOGLANGENG システムオプション” \(553 ページ\)](#)を参照してください。

言語切り替えの詳細については、[Multilingual Computing with SAS® 9.4](#) を参照してください。

3 章

各国語サポート関連のエンコーディング

概要: 各国語サポート関連のエンコーディング	9
エンコーディングとトランスコーディングの違い	10
よく用いられるエンコーディング方法	10
各国語サポートエンコーディング関連の標準組織	13
EBCDIC エンコーディングの制約	13
照合順序	14
照合順序の概要	14
言語照合の指定	16
SAS データセットのエンコーディングの確認	16
PROC CONTENTS を使用して SAS データセットの内容を表示する	16
SAS セッションのエンコーディングの動作	17
SAS リリース別のエンコーディングサポート	17
出力処理	17
入力処理	17
外部ファイルの読み込みと書き込み	18

概要: 各国語サポート関連のエンコーディング

エンコーディングは、文字セットの各文字を一意的な数値表現にマッピングし、すべてのコードポイントからなるテーブルを構成します。コードページと呼ばれるこのテーブルは、順序に従って並べられた文字セットで、各文字に数値のインデックス(コードポイント値)が関連付けられます。コードページ上の文字の位置によって、2 桁の 16 進数が決まります。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

文字セットは、文字と記号のセットで、言語または言語のグループで使用されます。文字セットには、各国語文字(特定の国または国のグループに固有の文字)、特殊文字(句読点など)、アクセント記号のないラテン文字 A から Z、数字 0 から 9、コンピュータに必要なコントロール文字が含まれます。

エンコーディング方法は、数値表現を文字のセットに割り当てる一連のルールです。これらのルールによって、エンコーディングのサイズ(文字の数値表現の保存に使用されるビット数)と文字が表示されるコードページの範囲が決まります。エンコーディング方法は、コンピュータ業界で開発された標準に従って定義され

ています。多くの場合、エンコーディング方法は、コンピュータハードウェアベンダに固有です。

エンコーディングは、エンコーディング方法を文字セットに適用した結果です。

個々の文字には、使用するコードページによって、コードページの異なる位置が割り当てられます。たとえば、ユーロ記号(€)などがあります。ユーロ記号の Unicode コードポイントは U+20AC です。

エンコーディングとトランスコーディングの違い

エンコーディングは、SAS セッションのデフォルトの作業環境を確立します。UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

しかし、国際的な環境で作業する(たとえば、ドイツ語 EBCDIC でエンコードされた SAS データにアクセスする)場合、大文字 Ä のドイツ語 EBCDIC コードポイントは 16 進数の 4A です。SAS Viya は UTF-8 にしか対応していないので、ドイツ語 EBCDIC でエンコードされたデータセットを正しく解釈するためには、そのデータをトランスコードする必要があります。トランスコーディングとは、1 つのエンコーディングから別のエンコーディングにデータを変換する処理のことです。概念的な情報については、[4 章, “各国語サポート関連のトランスコーディング” \(19 ページ\)](#)を参照してください。

よく用いられるエンコーディング方法

エンコーディングは、エンコーディング方法をコード化された文字セットに適用した結果です。

エンコーディング方法 エンコードされた文字体系を生成するために、制定された業界ルールをコード化された文字に適用すること。そのようなルールでは、特定の文字の数値表現およびエンコーディングでのそのコード位置を保存するために必要とされるビット数が規定されます。ISO 2022 と UTF-8 は、エンコーディング方法の一例です。詳細については、[“各国語サポートエンコーディング関連の標準組織” \(13 ページ\)](#)を参照してください。

エンコーディング方法は、数値表現を文字のセットに割り当てる一連のルールです。これらのルールによって、エンコーディングのサイズ(文字の数値表現の保存に使用されるビット数)と文字が表示されるエンコーディングの範囲が決まります。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

次に、一般的なエンコーディング方法の一覧を示します。

Unicode は一般的でよく使用されている文字セットです。

Unicode

最大で 110,116 文字の組み合わせを提供します。基本的に Unicode は、世界中のすべての言語に対応できます。Unicode はコード化された文字セットです。HTTP および HTML プロトコルは、Unicode に基づいています。次の UTF エンコーディングは、Unicode 文字セットの一部です。

UTF-8

ラテンスク립ト言語、ギリシャ語、キリル文字、アラビア語、ヘブライ語を含む MBCS エンコーディング。日本語、中国語および韓国語などの東アジアの言語も含まれます。UTF-8 の文字の幅は、1~4 バイトまでです。UTF-8 では、1 から 128 のコード位置に ASCII 文字を残すことによって、ASCII との互換性を維持しています。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

UTF-16

現代のすべての記述体系に含まれる共通文字のほとんどを包含する 16 ビット形式です。ほとんどの文字は一樣に 2 バイトを使用して表現されますが、4 バイトを必要とする追加文字のために、サロゲートスペースと呼ばれる拡張スペースがあります。

UTF-32

各文字が 4 バイトを占める 32 ビット形式です。

ASCII (American Standard Code for Information Interchange、ASCII コード、情報交換用米国標準コード)

128 文字の組み合わせを提供する米国の 7 ビットエンコーディングです。このエンコーディングには、大文字と小文字の英語、アメリカ英語の句読点、基数が 10 の数値、少数のコントロール文字が含まれています。この 128 文字のセットは、他のほとんどのエンコーディングで共通です。ASCII はパーソナルコンピュータと UNIX コンピュータで使用されます。

EBCDIC (Extended Binary Coded Decimal Interchange Code)ファミリ

256 文字の組み合わせを提供する 8 ビットエンコーディングです。複数の EBCDIC ベースのエンコーディングが存在します。EBCDIC は、IBM メインフレームとほとんどの IBM ミッドレンジコンピュータで使用されています。EBCDIC では、ISO 646 規則に従って、EBCDIC エンコーディングと 7 ビット (および 8 ビット)ASCII ベースのエンコーディングの間の変換を行います。95 EBCDIC グラフィカル文字には 82 個の不変文字(空白スペースを含む)が含まれていて、ほとんどの EBCDIC シングルバイトコードページで同じコード位置を占めています。さらに 13 個の変換グラフィック文字を含んでいて、ほとんどの EBCDIC シングルバイトコードページでさまざまなコード位置を占めています。バリエーション文字の詳細については、[“EBCDIC エンコーディングの制約” \(13 ページ\)](#)を参照してください。

マルチバイト文字セット(MBCS) EBCDIC エンコーディングもあります。

ISO (国際標準化機構) 646 ファミリ

128 文字の組み合わせを提供する国際標準の 7 ビットエンコーディングです。ISO 646 エンコーディングファミリは、各国バリエーションとして 12 コードポイントを使用する点を除いて ASCII に似ています。12 の各国バリエーションは、特定の言語で必要とされる特殊文字を表します。

ISO 8859 ファミリと Windows ファミリ

すべての ASCII コードポイントをサポートし、12 コードポイントを追加して 256 文字の組み合わせを提供する ASCII の 8 ビット拡張です。公式には ISO-8859-1 という名前を持つ Latin1 は、ISO 8859 エンコーディングファミリの中で最も頻繁に使用されるメンバです。Latin1 には、ASCII 文字に加えて、アクセント記号付き文字、西ヨーロッパの言語で必要とされる他の文字、およびいくつかの特殊文字が含まれています。

その他のエンコーディング

ISO 8859 ファミリには、他の言語用に設計されたメンバが他にもあります。次の表に、ISO によって承認された他のエンコーディングの説明を示します。

表 3.1 ISO によって承認されたその他のエンコーディング

ISO 標準	エンコーディング名	説明
ISO 8859-1	ラテン諸語 1	米国および西ヨーロッパの言語
ISO 8859-2	ラテン諸語 2	中央および東ヨーロッパの言語
ISO 8859-3	ラテン諸語 3	南ヨーロッパの言語、マルタ語、エスペラント語
ISO 8859-4	バルト諸語	北ヨーロッパ
ISO 8859-5	キリル語	スラブ系言語
ISO 8859-6	アラビア語	アラビア語
ISO 8859-7	ギリシャ語	現代ギリシャ語
ISO 8859-8	ヘブライ語	ヘブライ語およびイディッシュ語
ISO 8859-9	トルコ語	トルコ語
ISO 8859-10	ラテン諸語 6	北欧(イヌイット語、サーミ語、アイスランド語)
ISO 8859-11	ラテン/タイ語	タイ語
ISO 8859-13	ラテン諸語 7	バルト語
ISO 8859-14	ラテン諸語 8	ケルト語
ISO 8859-15	ラテン諸語 9	西ヨーロッパ
ISO-8859-16	ラテン諸語 10	中央および東ヨーロッパの言語

さらに、東アジア言語用に多数のエンコーディング標準が開発されています。その一部を次の表に示します。

表 3.2 ISO によって承認された一部の東アジア言語のエンコーディング

標準	エンコーディング名	説明
GB 2312-80	簡体字中国語	中華人民共和国
CNS 11643	繁体字中国語	台湾
Big-5	繁体字中国語	台湾

標準	エンコーディング名	説明
KS C 5601	韓国産業標準	韓国
JIS	日本工業規格	日本
Shift-JIS	日本工業規格マルチバイトエンコーディング	日本

さまざまな言語とロケールをサポートする EBCDIC および Windows の標準には、他のエンコーディングがあります。

各国語サポートエンコーディング関連の標準組織

SAS によってサポートされるエンコーディングは、次の標準組織によって定義されます。

国際標準化機構(ISO)

標準の策定と関連活動を推進して、国家間での物資およびサービスの自由な流れを促進し、知的、科学的、技術的情報の交換を提唱します。ISO では、エンコーディングの標準も策定します。

米国規格協会(ANSI)

米国の標準規格の調整を計り、それらの標準に承認を与えます。国際規格を作成するために、ISO (国際標準化機構)と協力して活動します。

Unicode コンソーシアム

すべての文字に一意的番号を付与する Unicode 標準を作成および推進します。

EBCDIC エンコーディングの制約

!、#、\$、@、\、[、^、`、{、}、|、~の文字は、各種の EBCDIC バリエーションで異なるコード位置を持つ可能性があるため、バリエーション文字であると見なされます。これらの文字はすべてのエンコーディング内に存在していますが、それらの 16 進値はエンコーディングによって異なる場合があります(次の表を参照)。

表 3.3 バリエーション文字

文字	1047	838	870	1025	1141	1142	1143	1144	1145	1146	1147	1148
!	5A	5A	4F	4F	4F	4F	4F	4F	BB	5A	4F	4F
#	7B	7B	7B	7B	7B	4A	63	B1	69	7B	B1	7B
\$	5B	5B	5B	5B	5B	67	67	5B	5B	4A	5B	5B
@	7C	7C	7C	7C	B5	80	EC	B5	7C	7C	44	7C

文字	1047	838	870	1025	1141	1142	1143	1144	1145	1146	1147	1148
\	E0	E0	E0	E0	EC	E0	71	48	E0	E0	48	E0
[AD	49	4A	4A	63	9E	B5	90	4A	B1	90	4A
]	BD	59	5A	5A	FC	9F	9F	51	5A	BB	B5	5A
^	5F	69	5F	5F	5F	5F	5F	5F	BA	BA	5F	5F
`	79	79	79	79	79	79	51	DD	79	79	A0	79
{	C0	C0	C0	C0	43	9	43	44	C0	C0	51	C0
}	D0	D0	D0	D0	DC	47	47	54	D0	D0	54	D0
	4F	4F	6A	6A	BB	BB	BB	BB	4F	4F	BB	BB
~	A1	A1	A1	A1	DC	DC	DC	58	BD	BC	BD	A1

プログラミング言語でよく使用される文字の例は、{や\$です。

これらの文字は、バリエーション文字と呼ばれます。たとえば、ドイツ語のメインフレームユーザーがä(コードポイント C0)を入力すると、アメリカのコンパイラはコードポイント C0 を{と解釈します。

照合順序

照合順序の概要

*照合順序*は、文字を並べ替える順序です。たとえば、SORT プロシジャを実行すると、照合順序により他の文字に対する特定の文字の並べ替え順序(高い、低い、等しい)が決まります。

デフォルトの照合順序はバイナリ照合で、セッションエンコーディングのコードページにおける各文字の場所に従って文字を並べ替えます。並べ替え順序は、コードページ内のコードポイントの配置に直接対応します。UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

バイナリ照合はコンピュータにとって最も効率が高く、最速の種類の照合です。ただし、バイナリ照合のレポート内の文字の検索は、この方法がよくわからない場合には多少複雑です。たとえば、バイナリ照合のレポートは、大文字で始まる単語を小文字で始まる単語と区別して表示し、アクセント記号付きの文字で始まる単語はアクセント記号のない文字で始まる単語の後に表示されます。そのため、ASCII ベースのエンコーディングでは、大文字の **Z** は小文字の **a** より前に表示されます。同じように、EBCDIC ベースのエンコーディングでは、小文字の **z** は大文字の **A** より前に表示されます。

バイナリ照合より優先される代替照合順序を要求できます。代替照合順序を要求するには、次の順序のいずれかを指定します。

- 変換テーブル名
- エンコーディング値
- 言語照合

表 3.4 (15 ページ) は、異なる照合順序を使用して単語のリストを並べ替えた結果を示します。

表 3.4 異なる照合順序の結果

バイナリ	変換テーブル	エンコーディング値	言語
Aaron	aardvark	Aaron	aardvark
Aztec	azimuth	Aztec	Aaron
Zeus	Aaron	Zeus	azimuth
aardvark	Aztec	aardvark	Aztec
azimuth	cote	azimuth	cote
cote	coté	cote	côte
coté	côte	coté	coté
côte	côté	côte	côté
côté	zebra	côté	zebra
zebra	zèbre	zebra	zèbre
zèbre	Zeus	zèbre	Zeus

最初の列は、ASCII ベースのエンコーディングで表示される文字のバイナリ照合の結果です。大文字と小文字で始まる単語は異なるグループになるため、一貫したアルファベット順ではありません。たとえば、単語 Zeus は、ASCII ベースのエンコーディング内で文字に割り当てられたコードポイントでの判断により、aardvark より前に表示されます。

2 番目の列は、小文字と大文字の並べ替えを交互にする変換テーブルを指定した結果です。変換テーブルを使用した場合、単語 aardvark は Zeus より前に表示されます。ただし、変換テーブルにより小文字 **a** は大文字 **A** よりも低い重みの値が割り当てられるため、単語 azimuth は Aaron より前に表示されます。さらに、アクセント記号は左から右の順序で比較して並べ替えられます。たとえば、coté は côte より前に表示されます。

最後の列は、セッションロケール fr_FR (French_France) の言語照合の結果で、単語をアルファベット順にする照合アルゴリズムを使用します。このアルゴリズムでは、小文字で始まる単語が大文字で始まる単語より前に表示されます。さらに、言語照合によりフランス語のロケール仕様に基づいて、アクセントが右から左の順序で比較して並べ替えられます。

SAS では、言語照合の実装に International Components for Unicode (ICU) を採用しています。ICU とその Unicode 照合アルゴリズム (UCA) の実装は標準となっています。照合順序は、指定したロケールの ICU により提供されたデフォルトが使用されます。

言語照合の指定

言語照合は、言語のルールに従って文字を並べ替え、直感的で文化的に受け入れられる結果を作成します。結果は、辞書、電話帳、本の索引などの紙媒体で使用される照合と似ています。言語照合は、レポートやその他のデータ表示の作成と、システム間の互換性の実現に役立ちます。

SAS では International Components for Unicode (ICU)を採用しています。ICU は、Unicode 照合アルゴリズム(UCA)と互換性のある言語照合のルーチンを提供するオープンソースライブラリです。UCA は、Unicode 文字列を比較して並べ替えができる標準です。

`SORTSEQ=`システムオプションは言語照合をサポートしないため、言語照合を要求するには、`PROC SORT` ステートメントの `SORTSEQ=`オプションを使用する必要があります。たとえば、次のステートメントでは、`French_France` のロケールに従って言語的に照合する `SORT` プロシジャを実行します。

```
options locale=fr_FR;

proc sort data=myfiles.test sortseq=linguistic;
  by name;
run;
```

言語照合を要求すると、SAS セッションロケール用に ICU で提供されたデフォルトの言語照合アルゴリズムが使用されます。このアルゴリズムには、各地域の言語、データのフォーマティングなどの規則、文化が反映されています。アルゴリズムを変更するには、`LINGUISTIC` キーワードの後にオプションをカッコで囲んで指定します。たとえば、異なるロケールを指定できます。また、小文字を大文字より前にしたり、その逆の順序にする `CASE_FIRST=`オプションも指定できます。その他いろいろな指定が可能です。通常は、ICU によりさまざまな言語とロケールがデフォルトで関連付けられるため、オプションを指定する必要はありません。言語オプションの詳細については、“[照合順序オプション](#)” (571 ページ) の `SORTSEQ=`オプション、または *SAS Viya Data Management and Utility Procedures Guide* 内の `PROC SORT` ステートメントの `SORTSEQ=`オプションを参照してください。

言語的に並べ替えられた SAS データセットには、`CONTENTS` プロシジャ出力の照合順序 `LINGUISTIC` を表示するソートインジケータが含まれます。このソートインジケータとともに、データセットにより言語照合順序の完全な説明がファイルのディスクリプタ情報に記録されます。この情報は `CONTENTS` プロシジャ出力にも表示されます。

SAS データセットのエンコーディングの確認

PROC CONTENTS を使用して SAS データセットの内容を表示する

データセットのエンコーディングを表示するには、`PROC CONTENTS` を使用します。

```
proc contents data=sashelp.class; run;
```

`PROC CONTENTS` コードからの出力を次に示します。エンコーディングは `US-ASCII` です。

The CONTENTS Procedure

Data Set Name	SASHELP.CLASS	Observations	19
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	01/17/2016 20:12:45	Observation Length	40
Last Modified	01/17/2016 20:12:45	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label	Student Data		
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	us-ascii ASCII (ANSI)		

SAS セッションのエンコーディングの動作

SAS リリース別のエンコーディングサポート

SAS で作成されたデータセットには、ファイルのディスクリプタ部分のエンコーディング属性に UTF-8 が自動的に設定されます。

出力処理

SAS でデータセットを作成する場合、エンコーディングは次のように決定されません。

- 新しい出力ファイルを作成する場合、データは、現在のセッションエンコーディングである UTF-8 で書き込まれます。
- OUTREP=オプションを使用して新しい出力ファイルを作成すると、OUTREP=値によって指定した、オペレーティングシステムのデフォルトセッションエンコーディングを使用してデータはファイルに書き込まれます。詳細については、[“OUTREP=データセットオプション” \(44 ページ\)](#)を参照してください。
- 新しい出力ファイルで既存のファイルを置き換える場合、新しいファイルは既存のファイルのエンコーディングを継承します。別の動作環境にある既存のファイルを置き換える出力処理の場合、または既存のファイルにエンコーディングが指定されていない場合は、現在のセッションエンコーディングが使用されます。

入力処理

SAS での入力(読み込み)処理におけるエンコーディング動作を次に示します。

- ほとんどのユーザーは、入力ファイルにエンコーディングを指定しない、デフォルトの動作を選びます。
- セッションエンコーディングとファイルで指定されたエンコーディングの間に互換性がない場合、データはセッションエンコーディングにトランスコードされます。たとえば、ファイルに指定されているエンコーディングが LATIN1 の場合、SAS Viya によりファイルは LATIN1 から UTF-8 のセッションエンコーディングにトランスコードされます。

- ファイルにエンコーディングが指定されていない場合、SAS は、ファイルのデータ表記が現在のセッションと異なる場合にのみ、データをトランスコードします。

外部ファイルの読み込みと書き込み

SAS は、現在のセッションエンコーディングを使用して外部ファイルの読み込みと書き込みを行います。SAS では、外部ファイルのエンコーディングがセッションエンコーディングと同じであると想定します。SAS Viya はファイルのエンコーディングは UTF-8 であると想定します。たとえば、外部ファイルを読み込んで新しい SAS データセットを作成する場合、SAS は、外部ファイルのエンコーディングと現在のセッションのエンコーディングが同じであると想定します。エンコーディングが同じでない場合、外部データは、新しい SAS データセットに誤って書き込まれる可能性があります。適切な ENCODING オプションを指定する必要があります。次に、Shift_JIS エンコーディングの指定の例を示します。

```
filename in 'external-file'  
encoding='Shift-JIS';  
data mylib.contacts;  
infile in;  
length name $ 30 first $ 30 street $ 60 zip $ 10 city $ 30;  
input name first street zip city;  
run;
```

入出力処理を実行する SAS ステートメントの構文の詳細については、“[SAS データのトランスコーディングを指定する SAS オプション](#)” (20 ページ)を参照してください。

4 章

各国語サポート関連のトランスコーディング

トランスコーディングの概要	19
トランスコーディングが必要となる場合	20
SAS データのトランスコーディングを指定する SAS オプション	20
トランスコーディングの留意点	21
互換性があるエンコーディングと互換性がないエンコーディング	21
互換性があるエンコーディングと互換性がないエンコーディングの概要 .	21
改行文字と EBCDIC と ASCII 間のデータ転送	22
トランスコーディングの防止	22
CVP エンジンを使用した文字データ切り捨てへの対応	23

トランスコーディングの概要

トランスコーディングとは、1つのエンコーディングから別のエンコーディングにデータを変換する処理のことです。トランスコーディングは、SAS セッションエンコーディングとデータのエンコーディングが異なる場合に必要になります。多くの場合、異なるロケールとエンコーディングを使用する動作環境間でデータを移動するときにトランスコーディングが必要になります。UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

並べ替え順序は、コードページ内のコードポイントの配置に直接対応します。

たとえば、Latin1 エンコーディングを使用する Linux 動作環境で作成されたファイルを、UTF-8 エンコーディングを使用する SAS Viya に移動する場合などが考えられます。SAS Viya でファイルが処理される時、データは、Latin1 エンコーディングから UTF-8 エンコーディングに再マッピングされます。

トランスコーディングでは、言語間の翻訳は行いません。トランスコーディングで行われるのは文字の再マッピングです。

異なるエンコーディングを使用する動作環境間でデータを動的にトランスコードするには、明確なエンコーディング値を指定する必要があります。詳細については、[18 章, "SAS 言語要素中のエンコーディング値" \(613 ページ\)](#)を参照してください。

トランスコーディングが必要となる場合

データがトランスコードされる頻度が高いのは、次のような状況です。

- 異なるロケールまたは異なる動作環境で実行している、2つの異なる SAS セッション間でデータを共有する場合
- UTF-8 以外のエンコーディングの外部ファイルを読み込む場合

SAS データのトランスコーディングを指定する SAS オプション

さまざまな言語要素に次の SAS オプションを使用すると、トランスコーディングを実行したりデフォルトのエンコーディング動作を無効にして別の動作を指定したりできます。これらの要素によって、SAS ファイルまたは SAS アプリケーションに異なるエンコーディングを指定したり、トランスコーディングを抑止したりできます。

表 4.1 SAS データのトランスコーディングを指定する SAS オプション

オプション	使用箇所
CHARSET=	ODS MARKUP ステートメント
CORRECTENCODING=	DATASETS プロシジャの MODIFY ステートメント
ENCODING=	%INCLUDE、FILE、FILENAME、INFILE、ODS の各ステートメント、FILE および INCLUDE コマンド
ENCODING=	DATA ステップ
INENCODING=	LIBNAME ステートメント
ODSCHARSET=	XML の LIBNAME ステートメント
ODSTRANTAB=	XML の LIBNAME ステートメント
OUTENCODING=	LIBNAME ステートメント
XMLENCODING=	XML の LIBNAME ステートメント

これらのオプションに使用するサポート対象エンコーディング値の一覧については、「データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値」(613 ページ)を参照してください。

トランスコーディングの留意点

通常、トランスコーディングは問題なく実行されますが、データに影響を与え、不満足な結果をもたらす場合があります。次に、例を示します。

- エンコーディング同士が競合する可能性があります。つまり、2種類のエンコーディングにおいて、同じ文字に対して異なるコードポイントが使用されていたり、2つの異なる文字に対して同じコードポイントが使用されていたりする場合があります。
- 1つのエンコーディングの文字が、別のエンコーディングに存在しない可能性があります。たとえば、特定のエンコーディングにはドル記号(\$)の文字がない場合があります。ドル記号をサポートしていないエンコーディングにデータをトランスコーディングすると、その文字が出力されなかったり、表示されなかったりします。
- 1つのエンコーディングでの1文字に対応するバイト数が、別のエンコーディングでの同じ文字に対応するバイト数と異なる可能性があります。2バイト文字セット(DBCS)からSBCSにトランスコーディングする場合などにみられます。したがって、トランスコーディングにより文字値が切り捨てられる可能性があります。
- トランスコーディング中、データを元のエンコーディングにトランスコーディングして戻すことができないなどのエラーが発生した場合は、データが失われる可能性があります。つまり、更新処理のためにデータセットを開いても、オブザベーションが更新されない場合があります。ただし、入力(読み込み)処理のためにデータセットを開き、出力データセットを何も開かない場合、SASは警告を発行して表示します。処理が進行すると、PRINTプロシジャまたは他の読み込み操作を使用して、トランスコーディングされないデータを表示できます。
- CEDAには、いくつかの処理制限があります。たとえば、CEDAは更新処理をサポートしません。
- 不正なエンコーディングについては、SAS Viyaセッションでコピーされたり、置き換えられたりする場合、SAS 7またはSAS 8データセットにスタンプできます。不正なエンコーディングスタンプは、PROC DATASETSのMODIFYステートメントのCORRECTENCODING=オプションによって訂正できます。文字変数にバイナリデータが含まれると、トランスコーディングによってデータが破損する可能性があります。

互換性があるエンコーディングと互換性がないエンコーディング

互換性があるエンコーディングと互換性がないエンコーディングの概要

ASCIIはほとんどのエンコーディングの基礎であり続け、多くのパーソナルコンピュータ、ミニコンピュータおよびワークステーションで使用されています。ただし、IBMメインフレームではEBCDICエンコーディングが使用されます。そのため、ASCIIとEBCDICのマシンとデータには互換性がありません。あるエンコ

ーディングの一部またはすべての文字が別のエンコーディングの文字と異なる場合、トランスコーディングが必要になります。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

ただし、トランスコーディングを避けるため、データセットを作成して SAS がトランスコードしないエンコーディング値を指定することもできます。たとえば、ENCODING=データセットオプション、LIBNAME ステートメントの INENCODING=オプション/OUTENCODING=オプションのいずれかで次の値を使用した場合、トランスコーディングは実行されません。

- ANY を使用して、EBCDIC エンコーディングと ASCII エンコーディング間でもトランスコーディングを実行しないように指定します。

注: ANY は BINARY の指定と同等です。データはバイナリであるため、実際のエンコーディングには意味がありません。

- ASCIIANY を使用して、すべての ASCII ベースのエンコーディングと互換性があるデータセットを作成できます。
- EBCDICANY を使用して、すべての EBCDIC ベースのエンコーディングと互換性があるデータセットを作成できます。

混在したエンコーディングを含む SAS データセットを作成できます。たとえば、Latin1 と Latin2 の両方を含めることができます。入力処理または出力処理でのデータのトランスコードは避けます。デフォルトでは、データは現在のセッションエンコーディングにトランスコードされます。

SAS ファイルが SAS Viya のセッションエンコーディングである UTF-8 に一致しない場合、データをトランスコードする必要があります。

SAS ファイルと SAS セッションが互換性のあるエンコーディングを持つ場合、トランスコーディングは必要ありません。

改行文字と EBCDIC と ASCII 間のデータ転送

ASCII 動作環境で実行するソフトウェアでは、改行文字で行末を指定する必要があります。データを z/OS から ASCII エンコーディングをサポートするマシンに転送する場合、EBCDIC 復帰改行文字が認識されないため、特に HTML 出力でフォーマティングの問題が発生する可能性があります。z/OS では、次の 2 セットの EBCDIC ベースのエンコーディングをサポートします。

- 名前に EBCDIC が含まれるエンコーディングは、ASCII 改行文字に対して EBCDIC 改行文字の従来のマッピングを使用します。そのため、データが 1 列で表示される可能性があります。
- 名前に Open Edition が含まれるエンコーディングは、改行文字を行末文字として使用します。ASCII を使用する動作環境にデータを転送する場合、EBCDIC 復帰改行文字を ASCII 改行文字にマッピングします。このマッピングにより、ASCII アプリケーションが行末を適切に解釈できるため、フォーマティングがより正確になります。

トランスコーディングの防止

エンコーディング値によっては、SAS がトランスコードしないデータセットを作成できるものがあります。入力処理または出力処理でのデータのトランスコー

ドは避けた方がよいでしょう。代わりに混在したエンコーディングのデータを含む SAS ライブラリを作成してください。たとえば、Latin1 と Latin2 の両方を含めることができます。

たとえば、ENCODING=データセットオプション、ILIBNAME ステートメントの NENCIDING=オプション/OUTENCODING=オプションのいずれかで次の値を使用する場合は、トランスコーディングを回避できます。

- ANY を使用して、EBCDIC エンコーディングと ASCII エンコーディング間でもトランスコーディングを実行しないように指定します。
注: ANY は BINARY の指定と同等です。データはバイナリであるため、実際のエンコーディングには意味がありません。
- ASCIIANY を使用すると、ASCII ベースのエンコーディング間ではトランスコーディングが不要と指定されます。
- EBCDICANY を使用すると、EBCDIC ベースのエンコーディング間ではトランスコーディングが不要と指定されます。

詳細については、“ENCODING=データセットオプション” (41 ページ) および “INENCODING=/OUTENCODING=オプション” (587 ページ) を参照してください。

TRANSCODE=オプションを使用して、特定の列のデータに対してトランスコーディングを行わずに、データセットの残りの文字データをトランスコードすることができます。詳細については、“PROC SQL の TRANSCODE=列修飾子” (590 ページ) を参照してください。

CVP エンジンを使用した文字データ切り捨てへの対応

データセットのエンコーディングが SAS セッションのエンコーディングと一致しない場合、データセット内の文字データのエンコーディングを変更すると、データが占める容量が増えるかも知れません。たとえば、SAS セッションが UTF-8 エンコーディングを使っているとき、Windows cp 1252(WLATIN1)エンコーディングのデータセットを読み込んだ場合、WLATIN1 では 1 バイトしか要しなかった文字が、UTF-8 では 2 バイトまたは 3 バイトを要する可能性があります。データの切り捨てを避けるには、文字変数ごとに、元のデータの幅より 1.5 倍広い幅が必要になります。

トランスコーディングを必要とする SAS データファイルを処理する場合は、CVP (文字変数埋め込み)エンジンに文字変数の長さの拡張を要求して、文字データの切り捨てが発生しないようにできます (変数の長さは、各変数の値を保存するために使用するバイト数です)。文字データの切り捨ては、1 つのエンコーディングにおける 1 文字のバイト数が、別のエンコーディングにおける同じ文字のバイト数と異なる場合に発生する可能性があります。

たとえば、1 バイト文字セットを、2 バイト文字セットやマルチバイト文字セットにトランスコードする場合などです。SBCS では各文字を 1 バイトで表すのに対し、DBCS では各文字を 1 バイトまたは 2 バイトで表します。MBCS は、1 から 4 バイトのさまざまな長さで文字を表します。たとえば、Wlatin2 から UTF-8 などの Unicode エンコーディングにトランスコーディングする場合、変数の長さ(バイト単位)は値を保持するのに不十分であり、結果として文字データが切り捨てられることとなります。

CVP エンジンを使用して、トランスコーディングの前に変数の長さが拡張されるように拡張量を指定し、それからデータを処理します。CVP エンジンは、トラン

スコーディングに向けてデータの準備をする中間エンジンと考えてください。長さが増えた後は、デフォルトのベースエンジンなどの主エンジンを使用して、実際にファイルを処理します。

CVP エンジンは、SAS データファイル限定の読み取り専用エンジンです。文字変数の拡張要求(たとえば、LIBNAME ステートメントを使用)は、次のいずれかの方法で実行できます。

- CVP エンジンを明示的に指定し、変数の長さの 1.5 倍のデフォルト拡張を使用します。
- LIBNAME の CVPBYTES=オプションまたは CVPMULTIPLIER=オプションにより、CVP エンジンを暗黙的に指定します。これらのオプションにより、拡張量を指定します。さらに、CVPENGINE=オプションを使用して、SAS ファイルを処理する主エンジンを指定できます。デフォルトは、デフォルトの SAS エンジンです。

たとえば、次の LIBNAME ステートメントでは、CVP エンジンを明示的に割り当てています。文字変数の長さは、長さを 1.5 倍にするデフォルト拡張を使用して増やします。たとえば、長さが 10 の文字変数の新しい長さは 15 になり、長さが 100 の文字変数の新しい長さは 150 になります。

```
libname expand cvp ' SAS data-library';
```

注: 拡張量は、どのような拡張にでも対応できるよう十分大きくする必要があります。大きさが不十分な場合は切り捨てが発生します。エラー情報は SAS ログに出力されます。

注: WHERE 式を使用してオブザベーションのサブセットを条件付きで選択する処理の場合は、CVP エンジンを使用するとパフォーマンスに影響することがあります。CVP エンジンを使用しない場合のファイルの処理は、CVP エンジンを使用する場合のファイルの処理より高速です。たとえば、CVP エンジンを使用する場合は、データセットにインデックスがあっても、WHERE 式を最適化するためにそのインデックスは使用されません。

注: CVP エンジンが自動的に出力形式や入力形式を調整することはありません。出力形式をデータ変数にアタッチした結果、データの切り捨てが生じる可能性があります。出力形式は、手動またはプログラムで調整してください。

注: CVP は読み取り専用のデータコピーを作成します。データの永続的なコピーを保存するには、新規にデータセットを作成する必要があります。

詳細と例については、“CVPBYTES=オプション、CVPENGINE=オプション、CVPMULTIPLIER=オプション”(577 ページ)を参照してください。

5 章 タイムゾーン

SAS タイムゾーン ID の概要	25
SAS でのタイムゾーンの指定	25
定義	25
SAS でのタイムゾーンについて	26
タイムゾーン処理のタスクと SAS 言語要素	27
タイムゾーンの例	30

SAS タイムゾーン ID の概要

国際的な顧客に対応するためには、ローカルおよび国際の時間値と日付値を処理する必要があります。この章では、タイムゾーンと次の処理方法について説明します。

- タイムゾーン
- 夏時間(DST)
- ローカル時間
- SAS 日時値

SAS でのタイムゾーンの指定

定義

夏時間(DST)

Summer Time とも呼ばれる *Daylight Saving Time (DST)* は、夏の日の長い時期に時計を 1 時間繰り上げて夕方の日照を有効に利用し、秋には元に戻すという方法です。北半球の夏時間は 3 月から 4 月までの間に始まり、9 月から 11 月までの間に終了します。北半球の標準時間は 9 月から 11 月の間に始まり、3 月と 4 月の間に終了します。南半球の夏時間は 9 月から 11 月までの間に始まり、3 月から 4 月までの間に終了します。南半球の標準時間は 3 月と 4 月の間に始まり、9 月から 11 月の間に終了します。北半球の多くの国では、夏時間を採用しています。夏時間は、すべての国で採用されているわけではありません。

Local time

TIMEZONE=システムオプションが設定されなかった場合に使用されるタイムゾーンのローカル時間を指定します。

Time zone

同一の標準時間を使用する地球上のエリア。通常はローカル時間と呼ばれます。タイムゾーンには、政治的な境界や地理的な境界があり、地域住民の便宜を考慮して調整される場合があります。インドや中国などのように、地理的に大きな国で1つのタイムゾーンを使用している国もあれば、ロシアやUSなどのように、複数のタイムゾーンを持つ国もあります。

タイムゾーン ID

フォワードスラッシュ(/)で区切られた地域とエリア。Asia/Tokyo は、タイムゾーン ID の1つです。タイムゾーン ID はJava タイムゾーンと互換性があります。タイムゾーン ID の一覧については、[付録 1, "タイムゾーン ID とタイムゾーン名" \(627 ページ\)](#)を参照してください。

タイムゾーン名

タイムゾーンを示す3文字または4文字を指定します。タイムゾーン名の一覧については、[付録 1, "タイムゾーン ID とタイムゾーン名" \(627 ページ\)](#)を参照してください。

タイムゾーンオフセット

+|-hh:mm または+|-hhmm という形式で示した、協定世界時(UTC)からタイムゾーンまでの時差。

ユーザーのローカル時間

TIMEZONE=システムオプションで指定されたタイムゾーンのローカル時間を指定します。

協定世界時(UTC)

基準子午線(英国のグリニッジ付近)の時間を指定します。UTC は、ISO 8601 基本形式 `yyyymmddThhmmss+|-hhmm`、または ISO 8601 拡張形式 `yyyy-mm-ddThh:mm:ss+|-hh:mm` を使用した日時値です。

SAS でのタイムゾーンについて

SAS 日時値は、ローカル時間の 1960 年 1 月 1 日 00:00:00 から経過秒数です。たとえば、英国における 1960 年 1 月 1 日 00:00:00 の SAS 日時値は 0 です。日本における 1960 年 1 月 1 日 00:00:00 の SAS 日時値も同様に 0 になります。英国と日本の間には 9 時間の時差があります。同じ瞬間の時刻であれば、この日時値は英国と日本の両方で 0 にはできません。絶対時間と日時値に対応するために、SAS は UTC 日付値と、タイムゾーンの時間値および日時値をサポートしています。

タイムゾーンの指定には、タイムゾーン ID またはタイムゾーン名を使用できません。タイムゾーン ID は、`region/area` という形式で地域とエリアを指定します。America/New_York は、タイムゾーン ID の1つです。タイムゾーン名は、タイムゾーンを指定します。東部標準時間(Eastern Standard Time)に使用される EST は、タイムゾーン名の1つです。タイムゾーンが夏時間に切り替わるとき、夏時間として別のタイムゾーン名が用意されています。PST は太平洋標準時間です。PDT は太平洋夏時間です。EET は東欧時間です。EEST は東欧夏時間です。タイムゾーン ID を指定すると、SAS によって夏時間かどうかが決まります。地域/エリアが標準時間または夏時間であるかをユーザーが判断する必要はありません。ゾーン ID とタイムゾーン名のリストについては、[付録 1, "タイムゾーン ID とタイムゾーン名" \(627 ページ\)](#)を参照してください。

タイムゾーンを指定すると、データセットや SAS カタログの作成または変更時にタイムゾーン固有のタイムスタンプをデータセットや SAS カタログに追加しま

す。また、タイムゾーン固有のタイムスタンプは、SAS の起動時には SAS ログに追加され、実行時には出力に追加されます。SAS タイムスタンプ定数は、ISO 8601 標準のタイムスタンプを指定する際に使用できます。次に、SAS タイムスタンプ定数を使用したタイムスタンプを示します。

```
tstamp='2013-05-17T09:15:30-05:00'dt;
```

SAS タイムスタンプ定数を指定すると、現在のタイムゾーンとタイムゾーンオフセットを考慮して、タイムスタンプがローカル日時値に変換されます。

タイムゾーンの設定は、TIME()関数、TODAY()関数、DATE()関数、DATETIME()関数に影響します。タイムゾーンが設定されると、日付値と日時値はそのタイムゾーンの値になります。

タイムゾーン処理のタスクと SAS 言語要素

タイムゾーンの設定

タイムゾーンの設定には、TIMEZONE=システムオプションを使用します。

```
options timezone='asia/tokyo';
```

このオプション値には、タイムゾーン名またはタイムゾーン ID を使用します。値は引用符で囲みます。タイムゾーン ID とタイムゾーン名については、[付録 1, "タイムゾーン ID とタイムゾーン名" \(627 ページ\)](#)を参照してください。詳細については、["TIMEZONE=システムオプション" \(560 ページ\)](#)を参照してください。

タイムゾーン ID オフセットの特定

TZONEOFF()関数を使用して、タイムゾーン名またはタイムゾーン ID のオフセットを特定します。

- TZONEOFF()関数は、現在のタイムゾーンのタイムゾーンオフセットを返します。
- TZONEOFF('time-zone-ID')関数は、'time-zone-ID'のタイムゾーンオフセットを返します。

このプログラムは、現在のタイムゾーン(EST)および東京のタイムゾーンオフセットを返します。

```
data _null_;
  o1=tzoneoff();
  o2=tzoneoff('asia/tokyo');
  put o1 time.;
  put o2 time.;
run;
```

SAS ログ内の出力を次に示します。

```
-5:00:00 9:00:00
```

2 つのタイムゾーン間の時差を特定するために、ABS()関数を使用できます。

```
diff=abs(tzoneoff('america/new_york') - tzoneoff('asia/tokyo'));
```

詳細については、["TZONEOFF 関数" \(385 ページ\)](#)を参照してください。

タイムゾーン ID またはタイムゾーン名の特定

次の関数を使用して、タイムゾーン名またはタイムゾーン ID を特定します。

- TZONEID()関数は現在のタイムゾーン ID を返します。
- TZONENAME()関数は、タイムゾーン ID と SAS 日時値の両方、または SAS 日時値のみに基づいて、現在のタイムゾーン名を返します。

次に、関数の使用例を示します。2013年3月10日は夏時間の開始日です。

```
options timezone='America/Chicago';
data _null_;
  tzid=tzoneid();
  put 'Current time zone is ' tzid;
  tzn=tzonename('america/los_angeles');
  put 'Time zone for Los Angeles: ' tzn;
  tznST=tzonename('america/los_angeles','10mar2013:01:00:00'dt);
  put 'Time zone for Los Angeles standard time: ' tznST;
  tznDT=tzonename('america/los_angeles','10mar2013:02:00:00'dt);
  put 'Time zone for Los Angeles daylight time: ' tznDT;
  tznSDT=tzonename('10mar2013:02:00:00'dt);
  put 'Time zone name for this SAS datetime: ' tznSDT;
run;
```

出力は次のようになります。

```
Current time zone is AMERICA/CHICAGO Time zone for Los Angeles: PST Time zone
for Los Angeles standard time: PST Time zone for Los Angeles daylight time: PDT
Time zone name for this SAS datetime: CDT
```

詳細については、“[TZONEID 関数](#)” (382 ページ)および“[TZONENAME 関数](#)” (384 ページ)を参照してください。

SAS と UTC 間での日時値の変換

次の関数を使用して、SAS 日時値を UTC 日時値へ、UTC 日時値を SAS 日時値へ変換します。

- TZONES2U()は、SAS 日時値を UTC 日時値へ変換します。
- TZONEU2S()関数は、UTC 日時値を SAS 日時値へ変換します。

次に、関数の使用例を示します。

```
options timezone='est';
data _null_;
  put ' The time zone is EST';
  diff=abs(tzoneoff('america/new_york') - tzoneoff('europe/london'));
  put ' New York-London difference: ' diff time.;
  diff=abs(tzoneoff('america/new_york') - tzoneoff('asia/tokyo'));
  put ' New York-Tokyo difference: ' diff time.;
  put ' The SAS datetime is 2013-03-15T09:15:00+00:00 ';
  put ' ';
  put ' Change a SAS datetime to a UTC value ';
  put ' ';
  put ' The time zone offset +00:00 is for London ';
  put ' Subtract the 5 hours for the EST time zone offset';
  stu1=tzones2u('2013-03-15T09:15:00+00:00'dt);
  put ' STU1 Using E8601DX: ' stu1 e8601dx.;
  put ' ';
  put ' 2013-03-15 9:15 AM in Tokyo is 2013-03-14 7:15 PM in New York';
  put ' Subtract the 5 hours for the EST time zone offset';
  stu2=tzones2u('2013-03-15T09:15:00+00:00'dt, 'Asia/Tokyo');
```

```

put 'STU2 Using E8601DX:' stu2 e8601dx.;
put '';
put 'Change a UTC to a SAS datetime value. ';
put '';
put '+00:00 is the time zone offset for London.';
put 'Subtract the 5 hours for the EST time zone offset';
uts1=tzoneu2s('2013-03-15T09:15:00+00:00'dt);
put 'UTS1 Using DATETIME:' uts1 datetime.;
put '';
put '9:15:00+00:00 is 18:15:00 in Tokyo. ';
put 'Subtract the 5 hours for the EST time zone offset';
uts2=tzoneu2s('2013-03-15T09:15:00+00:00'dt, 'Asia/Tokyo');
put 'UTS2 Using DATETIME:' uts2 datetime.;
run;

```

出力は次のようになります。

```

The time zone is EST New York-London difference: 5:00:00 New York-Tokyo
difference:14:00:00 The SAS datetime is 2013-03-15T09:15:00+00:00 Change a SAS
datetime to a UTC value The time zone offset +00:00 is for London Subtract the
5 hours for the EST time zone offset STU1 Using E8601DX:
2013-03-15T04:15:00-05:00 2013-03-15 9:15 AM in Tokyo is 2013-03-14 7:15 PM in
New York Subtract the 5 hours for the EST time zone offset STU2 Using E8601DX:
2013-03-14T14:15:00-05:00 Change a UTC to a SAS datetime value.+00:00 is the
time zone offset for London.Subtract the 5 hours for the EST time zone offset
UTS1 Using DATETIME:14MAR13:23:15:00 9:15:00+00:00 is 18:15:00 in
Tokyo.Subtract the 5 hours for the EST time zone offset UTS2 Using DATETIME:
15MAR13:13:15:00

```

詳細については、“TZONES2U 関数” (386 ページ)を参照してください。

タイムゾーンに基づいた ISO 8601 表記の時間値と日時値の書き込み

これらの出力形式では、ISO 8601 基本表記と拡張表記の時間値と日時値を書き出します。時間、UTC オフセット、またはその両方が、TIMEZONE=システムオプションの値に基づきます。

- B8601TX.および E8601TX.出力形式では、UTC 時間値をユーザーのローカル時間に変換し、その時間値をタイムゾーンオフセット付きの形式で書き出します。
- B8601DX.および E8601DX.出力形式では、UTC 日時値をユーザーのローカル時間に変換し、その日時値をタイムゾーンオフセットを使用して書き出します。
- B8601LX.および E8601LX.出力形式では、ユーザーのローカルタイムゾーンの UTC オフセットを使用して、ローカル日時値を書き出します。

このプログラムは、タイムゾーン ID の Australia/Sydney に基づき、時間値と日時値をフォーマットします。

```

options timezone='Australia/Sydney';
data _null_;
st='18:33:40't;
sdt='2013-03-17T14:30:22+00:00'dt;
put 'Time B8601TX:' st b8601tx.;
put 'Time E8601TX:' st e8601tx.;
put 'UTC B8601DX:' sdt b8601dx.;
put 'UTC E8601DX:' sdt e8601dx.;
put 'UTC B8601LX:' sdt b8601lx.;

```

```
put 'UTC E8601LX:' sdt e8601lx.;
run;
```

出力は次のようになります。

```
Time B8601TX: 043340+1000 Time E8601TX:04:33:40+10:00 UTC B8601DX:
20130318T123022+1100 UTC E8601DX: 2013-03-18T12:30:22+11:00 UTC B8601LX:
20130318T013022+1100 UTC E8601LX: 2013-03-18T01:30:22+11:00
```

詳細については、“B8601DX 出力形式” (82 ページ)、 “B8601LX 出力形式” (83 ページ)、 “B8601TX 出力形式” (85 ページ)、 “E8601DX 出力形式” (91 ページ)、 “E8601LX 出力形式” (92 ページ)、 “E8601TX 出力形式” (94 ページ)を参照してください。

タイムゾーンオフセットを使用した SAS 日時値の書き出し

次の出力形式では、TIMEZONE=システムオプションの値に基づき、UTC オフセットを使用して SAS 日時値を書き出します。

- NLDATMZ.出力形式では、SAS 日時値を *ddmmmyyyy:hh:mm:ss +|-hhmm* 形式で書き出します。
- NLDATMTZ.出力形式では、SAS 時間値を *hh:mm:ss +|-hhmm* 形式で書き出します。
- NLDATMWZ.出力形式では、SAS 日時値から週の曜日、月、日、年、午前|午後の部分を、*day-of-week, month-name dd, yyyy AM|PM +hhmm* 形式で書き出します。

このプログラムは、タイムゾーンオフセットを使用して、SAS 時間と日時値をフォーマットします。

```
options timezone='Indian/Maldives';
data _null_;
  st='18:33:40't;
  sdt='2013-03-17T14:30:22+00:00'dt;
  put 'Time NLDATMTZ:' st nldatmtz.;
  put 'SAS datetime NLDATMZ:' sdt nldatmz.;
  put 'SAS datetime NLDATMWZ:' sdt nldatmwz.;
run;
```

出力は次のようになります。

```
Time NLDATMTZ:18:33:40 +0500 SAS datetime NLDATMZ:17Mar2013:19:30:22 +0500 SAS
datetime NLDATMWZ:Sunday, March 17, 2013 07:30:22 PM +0500
```

詳細については、“NLDATMTZ 出力形式” (137 ページ)、 “NLDATMZ 出力形式” (150 ページ)、 および“NLDATMWZ 出力形式” (140 ページ)を参照してください。

タイムゾーンの例

この例では、ロサンゼルスから東京までのフライトの到着時間を使用します。

```
/* Set the time zone */
options timezone='America/Los_Angeles';
data depart;
/* Set the departure time */
  depart='2013-05-17T09:15:00-08:00'dt;
  put 'Depart Los Angeles:' depart nldatmwz.;
```

```

/* Set the flight time */
ftime='13:00't;
put 'Flight time=' ftime time.;
utc=depart+ftime;
put 'Arrive PST=' utc nldatmwz.;
put 'Arrive UTC=' utc nldatmwz.;
run;
/* Set the time zone for Tokyo */
options timezone='Asia/Tokyo';
data arrive;
set depart;
put 'Arrive in Tokyo ' utc nldatmwz.;
run;

```

出力は次のようになります。

```

39 /* Set the time zone */ 40 options timezone='America/Los_Angeles'; 41
data depart; 42 /* Set the departure time */ 43
depart='2013-05-17T09:15:00-08:00'dt; 44 put 'Depart Los Angeles: ' depart
nldatmwz.; 45 /* Set the flight time */ 46 ftime='13:00't; 47 put
'Flight time=' ftime time.; 48 utc=depart+ftime; 49 put 'Arrive PST='
utc nldatmwz.; 50 put 'Arrive UTC=' utc nldatmwz.; 51 run; Depart Los
Angeles: Friday, May 17, 2013 10:15:00 AM -0700 Flight time=13:00:00 Arrive
PST=Friday, May 17, 2013 11:15:00 PM -0700 Arrive UTC=Friday, May 17, 2013
11:15:00 PM -0700 NOTE: The data set WORK.DEPART has 1 observations and 3
variables.NOTE: DATA statement used (Total process time): real time
0.01 seconds cpu time 0.01 seconds 52 /* Set the time zone for
Tokyo */ 53 options timezone='Asia/Tokyo'; 54 data arrive; 55 set
depart; 56 put 'Arrive in Tokyo ' utc nldatmwz.; 57 run; Arrive in Tokyo
Friday, May 17, 2013 11:15:00 PM +0900 NOTE: There were 1 observations read
from the data set WORK.DEPART.NOTE: The data set WORK.ARRIVE has 1 observations
and 3 variables.NOTE: DATA statement used (Total process time): real
time 0.01 seconds cpu time 0.01 seconds

```


2 部

各国語サポート関連の自動呼び出しマクロ

6 章	各国語サポート関連の自動呼び出しマクロのディクショナリ.....	35
-----	----------------------------------	----

6 章

各国語サポート関連の自動呼び出しマクロのディクショナリ

カテゴリ別の自動呼び出しマクロエントリ	35
ディクショナリ	35
%KLOWCASE と %QKLOWCAS 自動呼び出しマクロ	35
%KTRIM と %QKTRIM 自動呼び出しマクロ	36
%KVERIFY 自動呼び出しマクロ	37

カテゴリ別の自動呼び出しマクロエントリ

次の表に、SAS 各国語サポート自動呼び出しマクロの概要を示します。詳細については、それぞれの各国語サポート関連のマクロに関する情報を参照してください。

カテゴリ	言語要素	説明
DBCS	%KLOWCASE と %QKLOWCAS 自動呼び出しマクロ (p. 35)	大文字を小文字に変更します。
	%KTRIM と %QKTRIM 自動呼び出しマクロ (p. 36)	末尾の空白を取り除きます。
	%KVERIFY 自動呼び出しマクロ (p. 37)	式に固有の最初の文字の位置を返します。

ディクショナリ

%KLOWCASE と %QKLOWCAS 自動呼び出しマクロ

大文字を小文字に変更します。

カテゴリ: DBCS

要件 MAUTOSOURCE システムオプション

構文

%KLOWCASE (text | text expression)

%QKLOWCAS (text | text expression)

詳細

%KLOWCASE と %QKLOWCAS マクロは、大文字の英字を小文字に変更します。次のリストにある特殊文字またはニーモニック演算子が引数に含まれている可能性がある場合は、%QKLOWCAS を使用します。

%KLOWCASE は、引数に引用符が含まれる場合でも、引用符なしの結果を返します。%QKLOWCAS は、次の特殊文字とニーモニック演算子がマスクされた結果を生成します。そのためマクロプロセッサは、それらの結果を、マクロ言語の要素ではなくテキストとして解釈します。

& % ' " () + - * / < > = ~ ^ ~ ; , blank AND OR NOT EQ NE LE LT GE GT IN

自動呼び出しマクロは SAS ライブラリに含まれます。このライブラリはサイトにインストールされていないか、またはサイト固有のバージョンの可能性がります。このマクロにアクセスできない場合、またはライブラリがサイト固有のバージョンかどうか確認する場合は、オンサイトの SAS サポート担当者にお問い合わせください。

%KTRIM と %QKTRIM 自動呼び出しマクロ

末尾の空白を取り除きます。

カテゴリ: DBCS

要件 MAUTOSOURCE システムオプション

構文

%KTRIM (text | text expression)

%QKTRIM (text | text expression)

詳細

KTRIM マクロと QKTRIM マクロは、末尾の空白を取り除きます。引数が次に示す特殊文字やニーモニック演算子を含む場合は、%QKTRIM を使用します。

QKTRIMS は、次の特殊文字とニーモニック演算子がマスクされた結果を生成します。そのためマクロプロセッサは、それらの結果を、マクロ言語の要素ではなくテキストとして解釈します。

& % ' " () + - * / < > = ~ ? ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN

自動呼び出しマクロは SAS ライブラリに含まれます。このライブラリはサイトにインストールされていないか、またはサイト固有のバージョンの可能性がります。このマクロにアクセスできない場合、またはライブラリがサイト固有のバージョンかどうか確認する場合は、オンサイトの SAS サポート担当者にお問い合わせください。

%KVERIFY 自動呼び出しマクロ

式に固有の最初の文字の位置を返します。

カテゴリ: DBCS

要件 MAUTOSOURCE システムオプション

構文

%KVERIFY (source, excerpt)

構文

source

excerpt に存在しない文字を確認する、テキストまたはテキスト式です。

excerpt

%KVERIFY で source の確認に使用される文字のセットを定義する、テキストまたはテキスト式です。

詳細

%KVERIFY は、excerpt に存在しない source 内の最初の文字の位置を返します。source 内のすべての文字が excerpt に存在する場合、%KVERIFY は 0 を返します。

自動呼び出しマクロは SAS ライブラリに含まれます。このライブラリはサイトにインストールされていないか、またはサイト固有のバージョンの可能性があります。このマクロにアクセスできない場合、またはライブラリがサイト固有のバージョンかどうかを確認する場合は、オンサイトの SAS サポート担当者にお問い合わせください。

3 部

各国語サポート関連のデータセットオプション

7 章	各国語サポート関連のデータセットオプションのディクショナリ	41
-----	---------------------------------------	----

7章

各国語サポート関連のデータセットオプションのディクショナリ

カテゴリ別のデータセットオプション	41
ディクショナリ	41
ENCODING=データセットオプション	41
OUTREP=データセットオプション	44

カテゴリ別のデータセットオプション

各国語サポートは、選択したデータセットオプションのデータセットコントロールカテゴリに影響を与えます。次の表に、各データセットオプションの概要を示します。詳細については、各データセットオプションのディクショナリエントリを参照してください。

カテゴリ	言語要素	説明
データセットコントロール	ENCODING=データセットオプション (p. 41)	SAS データセットの読み取りまたは書き込みに使用するエンコーディングを手動で指定します。
	OUTREP=データセットオプション (p. 44)	出力 SAS データセットのデータ表記を指定します。

ディクショナリ

ENCODING=データセットオプション

SAS データセットの読み取りまたは書き込みに使用するエンコーディングを手動で指定します。

該当要素: DATA ステップおよび PROC ステップ

カテゴリ: データセットコントロール

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

ENCODING= ANY | ASCIIANY | EBCDICANY | *encoding-value*

説明

ANY

トランスコーディングを行わないように指定します。

注 ANY は BINARY の指定と同等です。データはバイナリであるため、実際のエンコーディングには意味がありません。

ASCIIANY

混在したエンコーディングが ASCII エンコーディングの場合にトランスコーディングを行わないように指定します。

EBCDICANY

混在したエンコーディングが EBCDIC エンコーディングの場合にトランスコーディングを行わないように指定します。

encoding-value

エンコーディング値を指定します。

参照項目: [3章, “各国語サポート関連のエンコーディング” \(9 ページ\)](#)

詳細

ENCODING=の値は、SAS データセットに現在のセッションエンコーディングとは異なるエンコーディングがあることを示します。データは、データセットから読み取られるときに、指定したエンコーディングからセッションエンコーディングにトランスコードされます。また、データセットに書き込まれるときに、セッションエンコーディングから指定したエンコーディングにトランスコードされます。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

入力処理

デフォルトでは、入力処理のエンコーディングは次のように決まります。

- セッションエンコーディングとファイルに指定されたエンコーディングが異なる場合、データはセッションエンコーディングにトランスコードされます。
- ファイルにエンコーディングが指定されておらず、ファイルのデータ表記が現在のセッションのエンコーディングとは異なる場合、データは現在のセッションのエンコーディングにトランスコードされます。

出力処理

デフォルトでは、出力処理のエンコーディングは次のように決まります。

- データは現在のセッションのエンコーディングを使用してファイルに書き込まれます。ただし、OUTREP=データセットオプション、LIBNAME ステートメントの OUTENCODING=オプション、または ENCODING=データセットオプションを使用して別の出力表記が指定された場合を除きます。
- 既存のファイルが新しいファイルで置き換えられる場合、新しいファイルは既存のファイルのエンコーディングを継承します。

- 既存のファイルが、別の動作環境で作成されたか、エンコーディングが指定されていない新しいファイルで置き換えられる場合、新しいファイルは現在のセッションのエンコーディングを使用します。

注: 文字メタデータとデータ出力は、データセットの作成時に使用されたエンコーディングと異なるエンコーディングを指定すると文字化けします。この例では、出力するデータセットは内部的に ASCII としてエンコードされます。ただし、データセットオプションでは EBCDIC エンコーディングが指定されています。SAS は、データを EBCDIC から ASCII にトランスコードしようと試みますが、データはすでに ASCII です。その結果、データが文字化けしません。

```
data a;
x=1;
abc='abc';
run;
proc print data=a (encoding="ebcdic");
run;
```

注: 次の ENCODING=の値は無効です。

- UCS2
- UCS4
- UTF16
- UTF32

比較

- セッションエンコーディングは、ENCODING=システムオプションまたは LOCALE=システムオプションを使用して指定されます。それぞれの動作環境にはデフォルトのエンコーディングがあります。UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。
- LIBNAME ステートメントの INENCODING=オプション(入力ファイル用)および OUTENCODING=オプション(出力ファイル用)を使用して SAS ライブラリ用のエンコーディングを指定できます。LIBNAME ステートメントオプションと ENCODING=データセットオプションを両方とも指定すると、データセットオプションが使用されます。

例

例 1: エンコーディングが混在し、トランスコーディングが抑制された SAS データセットの作成

データセットオプション ENCODING=ANY を指定すると、混在したエンコーディングが含まれる SAS データセットを作成し、入力処理または出力処理でトランスコーディングを抑制できます。

この例では、新しいデータセット MYFILES.MIXED に、Latin1 エンコーディングを使用するデータと Latin2 エンコーディングを使用するデータが含まれます。データセットの処理時に、トランスコーディングは行われません。たとえば、Latin1 セッションエンコーディングの正しい Latin1 文字と Latin2 セッションエンコーディングの正しい Latin2 文字が表示されます。

```
libname myfiles 'SAS data-library';
data myfiles.mixed (encoding=any);
set work.latin1;
```

```
set work.latin2;
run;
```

例 2: 特定のエンコーディングを使用した SAS データセットの作成

出力処理では、現在のセッションエンコーディングを手動で指定できます。このアクションは、SAS Viya 以外の SAS 環境で使用するよう作成されたデータセットなどの場合に必要になることがあります。

たとえば、現在のセッションエンコーディングが UTF-8 の場合、ENCODING=WLATIN2 を指定してエンコーディング Wlatin2 を使用するデータセットを作成できます。次のステートメントでは、セッションエンコーディングのかわりに Wlatin2 エンコーディングを使用して新しいデータセットにデータを書き込むように指示しています。エンコーディングはまた、ファイルのディスクリプタ部分にも指定されます。

```
libname myfiles 'SAS data-library';
data myfiles.difencoding (encoding=wlatin2);
.
.
.
run;
```

関連項目:

- [概念の説明:3 章, “各国語サポート関連のエンコーディング” \(9 ページ\)](#)

ステートメントとコマンドのオプション:

- [“ENCODING=オプション” \(583 ページ\)](#)
- [“INENCODING=/OUTENCODING=オプション” \(587 ページ\)](#)

システムオプション:

- [“ENCODING システムオプション” \(549 ページ\)](#)
- [“LOCALE システムオプション” \(550 ページ\)](#)

OUTREP=データセットオプション

出力 SAS データセットのデータ表記を指定します。

該当要素: DATA ステップおよび PROC ステップ

カテゴリ: データセットコントロール

参照項目: [“OUTREP= Data Set Option” \(SAS Viya Data Set Options: Reference\).](#)

構文

OUTREP= *format*

関連項目:

[“OUTREP= Data Set Option” \(SAS Viya Data Set Options: Reference\)](#)

4 部

各国語サポート関連の出力形式

8 章	概要:各国語サポート関連の出力形式	47
9 章	各国語サポート関連の出力形式のディクショナリ	63

8 章

概要:各国語サポート関連の出力形式

国際化対応の日付、日時出力形式	47
通貨表現	53
通貨の概要	53
米国ドルの通貨表現	53
ユーロ	53
カスタマイズした通貨表現	54
各国語の通貨記号/国際標準の通貨コードを使用した、地域化された通貨表現	55
各国語の通貨記号/国際標準の通貨コードを使用した、独自の通貨表現 ..	57
例: 各国語の通貨記号および国際標準の通貨コードを使用した出力形式での通貨の表現	58
日付と時間のデフォルト幅の例外	60

国際化対応の日付、日時出力形式

SAS では、最も一般的に使用される英語の日付出力形式に相当する、国際化対応出力形式がサポートされています。いずれの場合でも、出力形式は対応する英語の出力形式と同様に機能します。最大幅、最小幅、デフォルトの幅のみが異なります。

表 8.1 国際化対応の日付、日時出力形式

言語	英語出力形式	国際化対応出力形式	最小	最大	デフォルト
アフリカーンス語(AFR)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	10
	WEEKDATX.	NLDATEW.	10	200	29

言語	英語出力形式	国際化対応出力形式	最小	最大	デフォルト
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
カタロニア語(CAT)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
クロアチア語(CRO)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
チェコ語(CSY)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
デンマーク語(DAN)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9

言語	英語出力形式	国際化対応出力形式	最小	最大	デフォルト
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
オランダ語(NLD)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WORDDATX.	NLDATE.	10	200	20
	WEEKDAY.	NLDATEWN.	4	200	9
フィンランド語(FIN)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
フランス語(FRA)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20

言語	英語出力形式	国際化対応出力形式	最小	最大	デフォルト
ドイツ語(DEU)	DATE.	NLDATE.	105	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
ハンガリー語(HUN)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
イタリア語(ITA)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
マケドニア語(MAC)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16

言語	英語出力形式	国際化対応出力形式	最小	最大	デフォルト
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDDATX.	EURDFWDX.	1	32	1
	WORDDATX.	NLDATEWN.	4	200	9
ノルウェー語(NOR)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
ポーランド語(POL)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	20	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
ポルトガル語(PTG)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATEWX.	10	200	20
ロシア語(RUS)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30

言語	英語出力形式	国際化対応出力形式	最小	最大	デフォルト
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
スペイン語(ESP)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
スロベニア語(SLO)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9
	WORDDATX.	NLDATE.	10	200	20
スウェーデン語(SVE)	DATE.	NLDATE.	10	200	20
	DATETIME.	NLDATM.	10	200	30
	DOWNAME.	NLDATEWN.	4	200	9
	MONNAME.	NLDATEMN.	4	200	9
	MONYY.	NLDATEYM.	6	200	16
	WEEKDATX.	NLDATEW.	10	200	29
	WEEKDAY.	NLDATEWN.	4	200	9

言語	英語出力形式	国際化対応出力形式	最小	最大	デフォルト
	WORDDATX.	NLDATE.	10	200	20

通貨表現

通貨の概要

通貨は、その国特有の交換手段です。SAS では、通貨の書き出しと読み込みに使用する出力形式と入力形式を提供します。

米国ドルの通貨表現

DOLLARw.d 出力形式と入力形式が、最初に、米国通貨の読み込みと書き出しを行うために導入されました。DOLLARw.d では次の記号を使用します。

- 米国通貨の先頭に通貨記号としてドル記号(\$)を使用します。
- カンマ(,)を桁区切り文字として、ピリオド(.)を小数点区切り文字として使用します。

例:

\$12,345.00

DOLLARXw.d はまた、通貨の先頭にドル記号(\$)を付け、桁区切り文字としてピリオド(.)、小数点区切り文字としてカンマ(,)を使用して書き出します。ピリオドとカンマが逆の通貨表現は、ヨーロッパの多くの国で使用される規則です。

例:

\$12.345,00

DOLLAR 出力形式および入力形式には次の制限があります。

- すべての通貨記号をサポートしません。
- ヨーロッパ諸国には、ピリオドとカンマが逆の通貨表現を使用しない国があります。
- 通貨記号の表現はコンピュータによって異なります(EBCDIC ベースのコンピュータと ASCII ベースのコンピュータでは文字の表示が異なります)。

ユーロ

EUROw.d 出力形式および入力形式は、1999 年に設立された欧州通貨同盟(EMU)によって制定されたユーロ通貨をサポートするために導入されました。

EUROw.d

例:

```
options locale=English_UnitedKingdom;
data _null_;
x=12345;
put x euro10.2;
run;
```

出力:

E12,345.00

EURO 出力形式はロケール依存ではありません。カンマ(,)を桁区切り文字として、ピリオド(.)を小数点区切り文字として使用します。

カスタマイズした通貨表現

カスタマイズした通貨表現を作成するには、FORMAT プロシジャが使用できます。次の例では、オーストラリアドル、スイスフランおよびイギリスポンドに固有の出力形式を作成します。FORMAT プロシジャの詳細については、*SAS Viya Data Management and Utility Procedures Guide* を参照してください。

例のコード 8.1 通貨表現をカスタマイズする SAS コード

```
proc format;

  picture aud low-<0='0,000,000,009.00'
    (prefix='-AU$' mult=100)
    0-high='0,000,00,009.00 '
    (prefix='AU$' mult=100);

  picture sfr low-<0='0,000,000,009.00'
    (prefix='-SFr.' mult=100)
    0-high='0,000,00,009.00 '
    (prefix='-SFr.' mult=100);

  picture bpd low-<0='0,000,000,009.00'
    (prefix='-BPd.' mult=100)
    0-high='0,000,00,009.00 '
    (prefix='BPd.' mult=100);

run;
data currency;
input aud sfr bpd 12.2;
datalines;
12345 12345 12345
0 0 0
-12345 -12345 -12345
;

proc print data=currency noobs;
var aud sfr bpd;
format aud aud. sfr sfr. bpd bpd.;
title 'Unique Currency Formats';
run;
```

aud	sfr	bpd
AU\$12,345.00	SFr. 12,345.00	BPd. 123.45
AU\$0.00	SFr. 0.00	BPd. 0.00
-AU\$12,345.00	-SFr. 12,345.00	-BPd. 123.45

通貨表現をカスタマイズすると柔軟性は向上しますが、プログラムでの対応が必要です。

各国語の通貨記号/国際標準の通貨コードを使用した、地域化された通貨表現

NLMNYw.d および NLMNYIw.d の出力形式と入力形式は、地域化された通貨を次の 2 つの形式で表現するために導入されました。

地域化された各国語通貨表現

ロケールの地域規則と慣習が反映されます。各国語のフォーマットは、NLMNYw.d 出力形式および入力形式を使用して指定します。NLMNYw.d 出力形式および入力形式を使用する場合、さらに LOCALE=オプションも使用してロケールを指定する必要があります。

例:

```
options locale=english_UnitedStates;
data _null_;
  x=12345;
  put x nlmny15.2;
run;
```

出力:

\$12,345.00

次に、一部の通貨の各国語表現を示します。

表 8.2 地域化された各国語通貨表現

LOCALE=	通貨	各国語表現
English_UnitedStates	米国ドル	\$12,345.00
French_Canada	カナダドル	12 345,00 \$
French_France	ユーロ	12 345,00 €
French_Switzerland	スイスフラン	SFr.12'345.00
German_Germany	ユーロ	12.345,00 €
German_Luxembourg	ユーロ	12.345,00 €
Spanish_Spain	ユーロ	12.345,00 €
Spanish_Venezuela	ベネズエラボリバル	VEF12.345,00

地域化された表示には、通貨表示に関する各国固有の地域規則が反映されます。たとえば、これらの EMU 諸国の中には、同じユーロ通貨を使用しているも通貨の表記が異なる場合があります。French_France は、桁区切り文字は使用せず、小数点区切り文字としてカンマを使用しますが、German_Germany および Spanish_Spain は桁区切り文字としてピリオドを、小数点区切り文字としてカンマを使用します。

国際標準の通貨コードを使用した、地域化された通貨表現

ISO 標準 4217 に準拠します。国際標準の通貨コードを使用した形式は、NLMNYIw.d 出力形式および入力形式を使用して指定します。国際標準の通貨コードを使用した形式は一般に、世界各国の通貨の対比を示すために使用されます。たとえば、航空券、貿易、株価などで使用されます。NLMNYIw.d 出力形式および入力形式を使用する場合、さらに LOCALE=オプションも使用してロケールを指定する必要があります。"International"を示す文字"I"が、出力形式名および入力形式名に付加されています。次に、例を示します。

```
options locale=english_UnitedStates;
data _null_;
x=12345; put x nlmnyi15.2;
run;
```

```
Output:
USD12,345.00
```

例:

国際標準の通貨コードを使用した通貨表現の一部を次に示します。

表 8.3 国際標準の通貨コードを使用した、ロケール別の通貨表現(ISO 標準 4217)

LOCALE=	通貨	国際標準の通貨コードを使用した表現
English_UnitedStates	米国ドル	USD12,345.00
French_Canada	カナダドル	12 345,00 CAD
French_France	ユーロ	12 345,00 EUR
French_Luxembourg	ユーロ	12.345,00 EUR
German_Germany	ユーロ	12.345,00 EUR
German_Switzerland	スイスフラン	CHF 12'345.00
Spanish_Spain	ユーロ	12.345,00 EUR
Spanish_Venezuela	ベネズエラボリバル	VEF12.345,00

国際標準の通貨コードを使用した表示には、通貨表示に関する各国固有の地域規則も反映されます。たとえば、すべてのロケールで桁区切り文字にカンマ、小数点区切り文字にピリオドが使用されるわけではありません。ISO 通貨コードが使用される場合、その位置はロケールにより異なります。EMU 諸国が通貨の後に通貨コードを付けるのに対し、English_UnitedStates、German_Switzerland および Spanish_Venezuela では通貨の前に ISO コードが付けられます。

ISO 標準 4217 通貨コードの全リストについては、<http://www.xe.com/iso4217.php> を参照してください。

各国語の通貨記号/国際標準の通貨コードを使用した地域化された通貨表現を使用する場合、LOCALE=システムオプションの値に依存することが最大の制限となります。

各国語の通貨記号/国際標準の通貨コードを使用した、独自の通貨表現

NLMNLISOW.d および NLMNIIOW.d の出力形式と入力形式は、LOCALE=オプションを併用しなくても各通貨を独自に表現できるように導入されました。各通貨は、ISO 標準 4217 通貨コードで示されます。

独自の各国語通貨表現

固有の ISO 通貨コードで指定します。各国語のフォーマットは、NLMNLISOW.d 出力形式および入力形式を使用して指定します。次の例では、USD が米国ドルの ISO 通貨コードです。

注: NLMNLISOW.d 出力形式および入力形式を使用する場合は、LOCALE=オプションは省略可能です。

例: `data _null_ ; x=12345; put x nlmnlusd15.2; run;`

出力:

US\$12,345.00 次に、一部の通貨の各国語表現を示します。

この例では、ロケールは fr_FR として指定されます。

`data _null_ ; x=12345; put x nlmnleur15.2; run;`

出力:

€12,345.00

表 8.4 ISO 通貨コードを使用した、独自の通貨表現

ISO 通貨コード	通貨	各国語表現
USD	米国ドル	US\$12,345.00
CAD	カナダドル	CA\$12,345.00
EUR	ユーロ	€12,345.00
CHF	スイスフラン	SFr.12,345.00
EUR	ユーロ	€12,345.00
EUR	ユーロ	€12,345.00
EUR	ユーロ	€12,345.00
VEB	ベネズエラボリバル	BsF.12,345.00

通貨記号または通貨コードは、多くの通貨で先頭に付けられます。また、桁区切り文字にはカンマ、小数点区切り文字にはピリオドが使用されます。ローカル通貨の通貨記号が現在の SAS セッションのエンコーディングでサポートされていない場合、NLMNLxxxw.d 出力形式では、3 文字の ISO 通貨コードを使用して値をフォーマットします。

国際標準の通貨コードを使用した、独自の通貨表現

固有の ISO 通貨コードで指定します。国際化対応出力形式は、NLMNIIOW.d 出力形式および入力形式を使用して指定します。国際標準の通貨コードを使

用した形式は一般に、世界各国の通貨の対比を示すために使用されます。たとえば、航空券、貿易、株価などで使用されます。たとえば、航空券、貿易、株価などで使用されます。"International"を示す文字"I"が、出力形式名および入力形式名に付加されています。次の例では、USD が米国ドルの ISO 通貨コードです。

注: NLMNISOw.d 出力形式および入力形式を使用する場合、LOCALE=オプションを使用してロケールを指定しません。

例: `data _null_ ; x=12345; put x nlmniusd15.2; run;`出力: **USD12,345.00**

国際標準の通貨コードを使用した通貨表現の一部を次に示します。

表 8.5 国際標準の通貨コードを使用した、ISO 通貨コード別の通貨表現

ISO 通貨コード	通貨	国際標準の通貨コードを使用した表現
USD	米国ドル	USD12,345.00
CAD	カナダドル	CAD12,345.00
EUR	ユーロ	EUR12,345.00
CHF	スイスフラン	CHF12,345.00
EUR	ユーロ	EUR12,345.00
EUR	ユーロ	EUR12,345.00
EUR	ユーロ	EUR12,345.00
VEB	ベネズエラボリバル	不明

国際標準の通貨コードを使用した表示では、該当する ISO コードが通貨の先頭に付けられます。また、桁区切り文字にはカンマ、小数点区切り文字にはピリオドが使用されます。

例: 各国語の通貨記号および国際標準の通貨コードを使用した出力形式での通貨の表現

この SAS プログラムでは、選択したアジア太平洋諸国について米国ドルに対する換算レートを使用します。出力では、各国の通貨が各国語の通貨記号および国際標準の通貨コードを使用した出力形式を使用して表記されます。

例のコード 8.2 各国語の通貨記号および国際標準の通貨コードを使用した出力形式で通貨を表記する SAS コード

```
data curr;
input ex_date mmdyy. usd aud hkd jpy sgd 12.2;
datalines;
```



```

061704 1.00000 1.45349 7.79930 110.110 1.71900 1

;

proc print data=curr noobs label;

    var ex_date usd aud hkd jpy sgd;

    format ex_date mmddyy. usd nlmnlusd15.2 aud nlmnlaud15.2 hkd
nlmnlhkd15.2
        jpy nlmnljpy15.2 sgd nlmnlsgd15.2;
2

    label ex_date='Date' usd='US' aud='Australia' hkd='Hong Kong'

        jpy='Japan' sgd='Singapore';

    title 'Exchange Rates for Selected Asian-Pacific Countries
(Localized Currency Codes)';

proc print data=curr noobs label;

    var ex_date usd aud hkd jpy sgd;

    format ex_date mmddyy. usd nlmniusd15.2 aud nlmniaud15.2 hkd
nlmnihkhd15.2
        jpy nlmnijpy15.2 sgd nlmnisgd15.2;
3

    label ex_date='Date' usd='US' aud='Australia' hkd='Hong Kong'

        jpy='Japan' sgd='Singapore';

    title 'Exchange Rates for Selected Asian-Pacific Countries
(International Currency Codes)';

run;

```

1. 2004年6月17日の時点の各換算レートが、データとしてSASプログラムに指定されます。
2. NLMNLISO 出力形式は、INPUT ステートメントで指定された数値データ項目にそれぞれ適用されます。これらの出力形式は、該当する各国語の出力形式で通貨を表示します。
3. NLMNIISO 出力形式は、INPUT ステートメントで指定された数値データ項目にそれぞれ適用されます。これらの出力形式は、該当する国際標準の通貨コードを使用した出力形式で通貨を表示します。

図 8.1 各国語の通貨記号および国際標準の通貨コードを使用した出力形式の出力

Exchange Rates for Selected Asian-Pacific Countries (Localized Currency Codes)					
2					
13:30 Monday, June 21, 2004					
Date	US	Australia	Hong Kong	Japan	Singapore
06/17/04	US\$1.00	AUS1.45	HK\$7.80	JPY110.11	SG\$1.72

Exchange Rates for Selected Asian-Pacific Countries (International Currency Codes)					
3					
13:30 Monday, June 21, 2004					
Date	US	Australia	Hong Kong	Japan	Singapore
06/17/04	USD1.00	AUD1.45	HKD7.80	JPY110.11	SGD1.72

日付と時間のデフォルト幅の例外

次の表に UTF-8 に対する推奨される幅を示します。

エンコーディング	ロケール	出力形式	デフォルトの幅	推奨される幅(>=)
UTF-8	CS_CS	nltimap	10	11
	EN_GB	nldatmw	30	32
		nltimap	10	16
	EN_US	nltimap	10	11
	JA_JP	nldatmw	30	43
		nldateyq	16	20
		nldatmyq	16	20
		nldatmwz	40	49
		nltimap	10	21
	KO_KR	nldatew	20	21
		nldatmw	30	46
		nldatmwz	40	52
		nltimap	10	22
	RU_RU	nldatew	20	21
		nldatmw	30	32
	ZH_CN	nldatew	20	25
		nldatmw	30	49
		nldatmwz	40	55
		nltimap	10	21

エンコーディング	ロケール	出力形式	デフォルトの幅	推奨される幅(>=)
	ZT_HK	nldatew	20	25
		nldatmw	30	49
		nldatmwz	40	55
		nltimap	10	21
	ZT_TW	nldatew	20	25
		nldatmw	30	49
		nldatmwz	40	55
		nltimap	10	21

9 章

各国語サポート関連の出力形式の ディクショナリ

各国語サポート関連の出力形式のカテゴリ	66
ディクショナリ	82
B8601DX 出力形式	82
B8601LX 出力形式	83
B8601TX 出力形式	85
BESTDOTX 出力形式	86
\$CPTDW 出力形式	88
\$CPTWD 出力形式	89
\$EBCDIC 出力形式	90
E8601DX 出力形式	91
E8601LX 出力形式	92
E8601TX 出力形式	94
EURO 出力形式	95
EUROX 出力形式	98
MINGUO 出力形式	100
NENGO 出力形式	102
NLBEST 出力形式	103
NLDATE 出力形式	105
NLDATEL 出力形式	106
NLDATEM 出力形式	107
NLDATEMD 出力形式	108
NLDATEMDL 出力形式	109
NLDATEMDM 出力形式	110
NLDATEMDS 出力形式	111
NLDATEMN 出力形式	112
NLDATES 出力形式	113
NLDATEW 出力形式	114
NLDATEWN 出力形式	115
NLDATEYM 出力形式	116
NLDATEYML 出力形式	118
NLDATEYMM 出力形式	118
NLDATEYMS 出力形式	119
NLDATEYQ 出力形式	120
NLDATEYQL 出力形式	121
NLDATEYQM 出力形式	122
NLDATEYQS 出力形式	123
NLDATEYR 出力形式	124
NLDATEYW 出力形式	125
NLDATM 出力形式	126
NLDATMAP 出力形式	127
NLDATMDT 出力形式	128

NLDATML 出力形式	129
NLDATMM 出力形式	130
NLDATMMD 出力形式	130
NLDATMMDL 出力形式	131
NLDATMMDM 出力形式	132
NLDATMMDS 出力形式	133
NLDATMMN 出力形式	134
NLDATMS 出力形式	135
NLDATMTM 出力形式	136
NLDATMTZ 出力形式	137
NLDATMW 出力形式	138
NLDATMWN 出力形式	139
NLDATMWZ 出力形式	140
NLDATMYM 出力形式	141
NLDATMYML 出力形式	142
NLDATMYMM 出力形式	143
NLDATMYMS 出力形式	143
NLDATMYQ 出力形式	144
NLDATMYQL 出力形式	145
NLDATMYQM 出力形式	146
NLDATMYQS 出力形式	147
NLDATMYR 出力形式	148
NLDATMYW 出力形式	149
NLDATMZ 出力形式	150
NLMNIAED 出力形式	151
NLMNIAUD 出力形式	152
NLMNIBGN 出力形式	153
NLMNIBRL 出力形式	154
NLMNICAD 出力形式	155
NLMNICHF 出力形式	156
NLMNICNY 出力形式	157
NLMNICZK 出力形式	158
NLMNIDKK 出力形式	159
NLMNIEEK 出力形式	160
NLMNIEGP 出力形式	161
NLMNIEUR 出力形式	162
NLMNIGBP 出力形式	163
NLMNIHKD 出力形式	164
NLMNIHRK 出力形式	165
NLMNIHUF 出力形式	166
NLMNIIDR 出力形式	167
NLMNIILS 出力形式	168
NLMNIINR 出力形式	169
NLMNIJPY 出力形式	170
NLMNIKRW 出力形式	171
NLMNILTL 出力形式	172
NLMNILVL 出力形式	173
NLMNIMOP 出力形式	174
NLMNIMXN 出力形式	175
NLMNIMYR 出力形式	176
NLMNINOK 出力形式	177
NLMNINZD 出力形式	178
NLMNIPLN 出力形式	179
NLMNIRUB 出力形式	180
NLMNISEK 出力形式	181
NLMNISGD 出力形式	182

NLMNITHB 出力形式	183
NLMNITRY 出力形式	184
NLMNITWD 出力形式	185
NLMNIUSD 出力形式	186
NLMNIZAR 出力形式	187
NLMNLAED 出力形式	188
NLMNLAUD 出力形式	189
NLMNLBGN 出力形式	190
NLMNLBRL 出力形式	191
NLMNLCAD 出力形式	192
NLMNLCHF 出力形式	193
NLMNLCNY 出力形式	194
NLMNLCZK 出力形式	195
NLMNLDKK 出力形式	196
NLMNLEEK 出力形式	197
NLMNLEGP 出力形式	198
NLMNLEUR 出力形式	199
NLMNLGBP 出力形式	200
NLMNLHKD 出力形式	201
NLMNLHRK 出力形式	202
NLMNLHUF 出力形式	203
NLMNLIDR 出力形式	204
NLMNLILS 出力形式	205
NLMNLINR 出力形式	206
NLMNLJPY 出力形式	207
NLMNLKRW 出力形式	208
NLMNLLTL 出力形式	209
NLMNLLVL 出力形式	210
NLMNLMOP 出力形式	211
NLMNLMXN 出力形式	212
NLMNLMYR 出力形式	213
NLMNLNOK 出力形式	214
NLMNLNzd 出力形式	215
NLMNLPLN 出力形式	216
NLMNLRUB 出力形式	217
NLMNLSEK 出力形式	218
NLMNLSGD 出力形式	219
NLMNLTHB 出力形式	220
NLMNLTRY 出力形式	221
NLMNLTWD 出力形式	222
NLMNLUSD 出力形式	223
NLMNLZAR 出力形式	224
NLMNY 出力形式	225
NLMNYI 出力形式	226
NLNUM 出力形式	227
NLNUMI 出力形式	229
NLPCT 出力形式	230
NLPCTI 出力形式	232
NLPCTN 出力形式	233
NLPCTP 出力形式	234
NLPVALUE 出力形式	235
NLSTRMON 出力形式	236
NLSTRQTR 出力形式	237
NLSTRWK 出力形式	239
NLTIMAP 出力形式	240
NLTIME 出力形式	241

\$UCS2B 出力形式	242
\$UCS2BE 出力形式	243
\$UCS2L 出力形式	244
\$UCS2LE 出力形式	246
\$UCS2X 出力形式	247
\$UCS2XE 出力形式	248
\$UCS4B 出力形式	249
\$UCS4BE 出力形式	251
\$UCS4L 出力形式	252
\$UCS4LE 出力形式	253
\$UCS4X 出力形式	254
\$UCS4XE 出力形式	256
\$UESC 出力形式	257
\$UESCE 出力形式	258
\$UNCR 出力形式	259
\$UNCRE 出力形式	260
\$UPAREN 出力形式	261
\$UPARENE 出力形式	262
\$UTF8X 出力形式	263
\$UTF8XE 出力形式	264
WEEKU 出力形式	265
WEEKV 出力形式	267
WEEKW 出力形式	268
YEN 出力形式	270
YYWEEKU 出力形式	271
YYWEEKV 出力形式	272
YYWEEKW 出力形式	274

各国語サポート関連の出力形式のカテゴリ

一部の出力形式は SAS 内でのみ実行され、一部の出力形式は SAS 内および CAS エンジン上で実行されます。出力形式カテゴリとして CAS を指定した場合、出力形式は SAS 内および CAS サーバー上で実行されます。出力形式カテゴリとして CAS を指定しない場合、出力形式は SAS 内でのみ実行されます。たとえば、NLSDATE 出力形式は SAS 内および CAS サーバー上で実行されるため、カテゴリとして CAS を指定します。\$UNCR 形式出力は SAS でしか実行されないため、カテゴリとして CAS を指定しません。

各国語サポートに関するカテゴリ:

カテゴリ	説明
BIDI テキストの操作	データ変数から双方向のデータ値を書き出すように指示します。
CAS	この出力形式を CAS サーバー上で実行するように指示します。
文字	文字変数から文字データ値を書き出すように指示します。
通貨の換算	1 つの通貨を別の通貨に変換するように指示します。

カテゴリ	説明
DBCS	アジア言語で使用される 2 バイト文字セットを変換するように指示します。
ヘブライ語のテキストの操作	データ変数からヘブライ語のデータを読み込むように指示します。
国際化対応日時	日付、時刻および日時を表す変数からデータ値を書き出すように指示します。
数値	数値変数から数値データ値を書き出すように指示します。

カテゴリ	言語要素	説明
CAS	EURO 出力形式 (p. 95)	先頭にユーロ(E)記号を付け、3桁ごとにカンマを入れ、小数部分との区切りにはピリオドを使用して数値を出力します。
	EUROX 出力形式 (p. 98)	先頭にユーロ(E)記号を付け、3桁ごとにピリオドを入れ、小数部分との区切りにはカンマを使用して数値を出力します。
	NENGO 出力形式 (p. 102)	SAS 日付値を日本の日付として e.yyymmdd 形式で書き出します。
	NLBEST 出力形式 (p. 103)	ロケールに基づいて最適な数値表記を書き出します。
	NLDATE 出力形式 (p. 105)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から日付を書き出します。
	NLDATEL 出力形式 (p. 106)	SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。
	NLDATEM 出力形式 (p. 107)	SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。
	NLDATEMD 出力形式 (p. 108)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から月の名前と月の日を書き出します。
	NLDATEMDL 出力形式 (p. 109)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。
	NLDATEMDM 出力形式 (p. 110)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。
	NLDATEMDS 出力形式 (p. 111)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。
	NLDATEMN 出力形式 (p. 112)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から月の名前を書き出します。
	NLDATES 出力形式 (p. 113)	SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。

カテゴリ	言語要素	説明
	NLDATEW 出力形式 (p. 114)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から日付と曜日を書き出します。
	NLDATEWN 出力形式 (p. 115)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から曜日を書き出します。
	NLDATEYM 出力形式 (p. 116)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と月の名前を書き出します。
	NLDATEYML 出力形式 (p. 118)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。
	NLDATEYMM 出力形式 (p. 118)	SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して月と年を日付値として書き出します。
	NLDATEYMS 出力形式 (p. 119)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。
	NLDATEYQ 出力形式 (p. 120)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と四半期を書き出します。
	NLDATEYQL 出力形式 (p. 121)	SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して年と四半期値(Q1-Q4)を書き出します。
	NLDATEYQM 出力形式 (p. 122)	SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して年と四半期値(Q1-Q4)を書き出します。
	NLDATEYQS 出力形式 (p. 123)	SAS 日付値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して年と四半期値(1-4)を書き出します。
	NLDATEYR 出力形式 (p. 124)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年を書き出します。
	NLDATEYW 出力形式 (p. 125)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と週を書き出します。
	NLDATM 出力形式 (p. 126)	SAS 日時値を、指定されたロケールの日時値に変換し、その日時値から日時を書き出します。
	NLDATMAP 出力形式 (p. 127)	SAS 日時値を、指定されたロケールの日時値に変換し、その日時値から午前または午後付きの日時を書き出します。
	NLDATMDT 出力形式 (p. 128)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前、月の日および年を書き出します。
	NLDATML 出力形式 (p. 129)	SAS 日時値を、指定されたロケールの日時文字列に変換し、月、日、年、時間を書き出します。
	NLDATMM 出力形式 (p. 130)	SAS 日時値を、指定されたロケールの日付文字列に変換し、月名と時間の短縮形を使用して月、日、年、時間を書き出します。

カテゴリ	言語要素	説明
	NLDATMMD 出力形式 (p. 130)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前と月の日を書き出します。
	NLDATMMDL 出力形式 (p. 131)	SAS 日時値を、指定されたロケールの日付文字列に変換し、長い形式で月と日を書き出します。
	NLDATMMDM 出力形式 (p. 132)	SAS 日時値を、指定されたロケールの日付文字列に変換し、短縮形を使用して月と日を書き出します。
	NLDATMMDS 出力形式 (p. 133)	SAS 日時値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して月と日を書き出します。
	NLDATMMN 出力形式 (p. 134)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前を書き出します。
	NLDATMS 出力形式 (p. 135)	SAS 日時値を、指定されたロケールの日付文字列に変換し、MM/dd/yyyy の形式で書き出します。
	NLDATMTM 出力形式 (p. 136)	SAS 日時値の時間部分を、指定されたロケールの時間値に変換し、その時間値から時刻を書き出します。
	NLDATMTZ 出力形式 (p. 137)	ロケールの SAS 日時値の時間部分を、時刻とタイムゾーンに変換します。
	NLDATMW 出力形式 (p. 138)	SAS 日時値を、曜日および日時形式の、ロケール依存の日付文字列に変換します。
	NLDATMWN 出力形式 (p. 139)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から曜日を書き出します。
	NLDATMWZ 出力形式 (p. 140)	指定されたロケールの SAS 日時値を、曜日、日時、およびタイムゾーン値に変換します。
	NLDATMYM 出力形式 (p. 141)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と月の名前を書き出します。
	NLDATMYML 出力形式 (p. 142)	SAS 日時値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。
	NLDATMYMM 出力形式 (p. 143)	SAS 日時値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。
	NLDATMYMS 出力形式 (p. 143)	SAS 日時値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して月と年を書き出します。
	NLDATMYQ 出力形式 (p. 144)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と四半期を書き出します。
	NLDATMYQL 出力形式 (p. 145)	SAS 日時値を、指定されたロケールの日付文字列に変換し、四半期(1-4)値、年の順に書き出します。
	NLDATMYQM 出力形式 (p. 146)	SAS 日時値を、指定されたロケールの日付文字列に変換し、四半期(1-4)、年の順に書き出します。

カテゴリ	言語要素	説明
	NLDATMYQS 出力形式 (p. 147)	SAS 日時値を、指定されたロケールの日付文字列に変換し、年と四半期(1-4)を数字と区切り文字を使用して書き出します。
	NLDATMYR 出力形式 (p. 148)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年を書き出します。
	NLDATMYW 出力形式 (p. 149)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と週の名前を書き出します。
	NLDATMZ 出力形式 (p. 150)	SAS 日時値を、タイムゾーンおよび日時形式の、ロケール依存の日付文字列に変換します。
	NLMNY 出力形式 (p. 225)	指定されたロケールで、ローカル通貨を使用してローカル表示形式で通貨データを書き出します。
	NLMNYI 出力形式 (p. 226)	指定されたロケールで、通貨データを国際表示形式で書き出します。
	NLNUM 出力形式 (p. 227)	指定されたロケールで、数値をローカル表示形式で書き出します。
	NLNUMI 出力形式 (p. 229)	指定されたロケールで、数値を国際表示形式で書き出します。
	NLPCT 出力形式 (p. 230)	指定されたロケールで、百分率のデータをローカル表示形式で書き出します。
	NLPCTI 出力形式 (p. 232)	指定されたロケールで、百分率のデータを国際表示形式で書き出します。
	NLPCTN 出力形式 (p. 233)	百分率を書き出します。負の値にはマイナス記号を使用します。
	NLPCTP 出力形式 (p. 234)	ロケール固有の数値を百分率として書き出します。
	NLPVALUE 出力形式 (p. 235)	指定されたロケールで、p 値をローカル表示形式で書き出します。
	NLSTRMON 出力形式 (p. 236)	指定されたロケールの月名を書き出します。
	NLSTRQTR 出力形式 (p. 237)	指定されたロケールの四半期の形式で数値を書き出します。
	NLSTRWK 出力形式 (p. 239)	指定されたロケールの曜日の形式で数値を書き出します。
	NLTIMAP 出力形式 (p. 240)	SAS 時間値を、指定されたロケールの時間値に変換し、その時間値から午前または午後付きの時間値を書き出します。NLTIMAP は、SAS 日時値も変換します。
	NLTIME 出力形式 (p. 241)	SAS 時間値を、指定されたロケールの時間値に変換し、その時間値から時刻を書き出します。NLTIME は、SAS 日時値も変換します。

カテゴリ	言語要素	説明
	YEN 出力形式 (p. 270)	円記号、桁区切り、小数点付きの数値を出力します。
	YYWEEKU 出力形式 (p. 271)	曜日情報を除外し、U アルゴリズムを使用して、10 進数の週番号を書き出します。
	YYWEEKV 出力形式 (p. 272)	曜日情報を除外し、V アルゴリズムを使用して、10 進数の週番号を書き出します。
	YYWEEKW 出力形式 (p. 274)	曜日情報を除外し、W アルゴリズムを使用して、10 進数の週番号を書き出します。
ISO 8601	B8601DX 出力形式 (p. 82)	協定世界時(UTC)日時値をユーザーのローカル日付と時間に調整します。その後、ローカルの日付と時間は、ISO 8601 規格の日時とタイムゾーンの基本表記 <code>yyyymmddThhmmss+hhmm</code> を使用して書き出されます。
	B8601LX 出力形式 (p. 83)	ISO 8601 規格の基本表記 <code>yyyymmddThhmmss+ -hhmm</code> を使用して、ローカル時間と UTC 間のタイムゾーンオフセット差を追加することにより、日時値をローカル時間として書き出します。
	B8601TX 出力形式 (p. 85)	協定世界時(UTC)値をユーザーのローカル時間に調整します。その後、ISO 8601 規格の基本時間表記 <code>hhmmss+ -hhmm</code> を使用して、ローカル時間を書き出します。
	E8601DX 出力形式 (p. 91)	協定世界時(UTC)日時値をユーザーのローカル日付と時間に調整します。その後、ローカルの日付と時間は、ISO 8601 規格の日時とタイムゾーンの拡張表記 <code>yyyy-mm-ddThh:mm:ss+hh:mm</code> を使用して書き出されます。
	E8601LX 出力形式 (p. 92)	ISO 8601 規格の拡張表記 <code>yyyy-mm-ddThh:mm:ss+ -hh:mm</code> を使用して、ローカル時間と UTC 間のタイムゾーンオフセット差を追加することにより、日時値をローカル時間として書き出します。
	E8601TX 出力形式 (p. 94)	協定世界時(UTC)値をユーザーのローカル時間に調整します。その後、ローカルの間は、ISO 8601 規格の日時とタイムゾーンの拡張表記 <code>hh:mm:ss+ -hh:mm</code> を使用して読み込まれます。
数値	BESTDOTX 出力形式 (p. 86)	最適な表記を SAS が選択し、ピリオド(.)を小数点として使用することを指定します。
	EURO 出力形式 (p. 95)	先頭にユーロ(E)記号を付け、3 桁ごとにカンマを入れ、小数部分との区切りにはピリオドを使用して数値を出力します。
	EUROX 出力形式 (p. 98)	先頭にユーロ(E)記号を付け、3 桁ごとにピリオドを入れ、小数部分との区切りにはカンマを使用して数値を出力します。
	NLBEST 出力形式 (p. 103)	ロケールに基づいて最適な数値表記を書き出します。
	NLMNIAED 出力形式 (p. 151)	アラブ首長国連邦の国際表示形式で通貨データを書き出します。

カテゴリ	言語要素	説明
	NLMNIAUD 出力形式 (p. 152)	オーストラリアの国際表示形式で通貨データを書き出します。
	NLMNIBGN 出力形式 (p. 153)	ブルガリアの国際表示形式で通貨データを書き出します。
	NLMNIBRL 出力形式 (p. 154)	ブラジルの国際表示形式で通貨データを書き出します。
	NLMNICAD 出力形式 (p. 155)	カナダの国際表示形式で通貨データを書き出します。
	NLMNICHF 出力形式 (p. 156)	リヒテンシュタインとスイスの国際表示形式で通貨データを書き出します。
	NLMNICNY 出力形式 (p. 157)	中国の国際表示形式で通貨データを書き出します。
	NLMNICZK 出力形式 (p. 158)	チェコ共和国の国際表示形式で通貨データを書き出します。
	NLMNIDKK 出力形式 (p. 159)	デンマーク、フェロー諸島およびグリーンランドの国際表示形式で通貨データを書き出します。
	NLMNIEEK 出力形式 (p. 160)	エストニアの国際表示形式で通貨データを書き出します。
	NLMNIEGP 出力形式 (p. 161)	エジプトの国際表示形式で通貨データを書き出します。
	NLMNIEUR 出力形式 (p. 162)	オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインの国際表示形式で通貨データを書き出します。
	NLMNIGBP 出力形式 (p. 163)	英国の国際表示形式で通貨データを書き出します。
	NLMNIHKD 出力形式 (p. 164)	香港の国際表示形式で通貨データを書き出します。
	NLMNIHRK 出力形式 (p. 165)	クロアチアの国際表示形式で通貨データを書き出します。
	NLMNIHUF 出力形式 (p. 166)	ハンガリーの国際表示形式で通貨データを書き出します。
	NLMNIIDR 出力形式 (p. 167)	インドネシアの国際表示形式で通貨データを書き出します。
	NLMNIILS 出力形式 (p. 168)	イスラエルの国際表示形式で通貨データを書き出します。
	NLMNIINR 出力形式 (p. 169)	インドの国際表示形式で通貨データを書き出します。
	NLMNIJPY 出力形式 (p. 170)	日本の国際表示形式で通貨データを書き出します。

カテゴリ	言語要素	説明
	NLMNIKRW 出力形式 (p. 171)	韓国の国際表示形式で通貨データを書き出します。
	NLMNILTL 出力形式 (p. 172)	リトアニアの国際表示形式で通貨データを書き出します。
	NLMNILVL 出力形式 (p. 173)	ラトビアの国際表示形式で通貨データを書き出します。
	NLMNIMOP 出力形式 (p. 174)	マカオの国際表示形式で通貨データを書き出します。
	NLMNIMXN 出力形式 (p. 175)	メキシコの国際表示形式で通貨データを書き出します。
	NLMNIMYR 出力形式 (p. 176)	マレーシアの国際表示形式で通貨データを書き出します。
	NLMNINOK 出力形式 (p. 177)	ノルウェーの国際表示形式で通貨データを書き出します。
	NLMNINZD 出力形式 (p. 178)	数値
	NLMNIPLN 出力形式 (p. 179)	ポーランドの国際表示形式で通貨データを書き出します。
	NLMNIRUB 出力形式 (p. 180)	ロシアの国際表示形式で通貨データを書き出します。
	NLMNISEK 出力形式 (p. 181)	スウェーデンの国際表示形式で通貨データを書き出します。
	NLMNISGD 出力形式 (p. 182)	シンガポールの国際表示形式で通貨データを書き出します。
	NLMNITHB 出力形式 (p. 183)	タイの国際表示形式で通貨データを書き出します。
	NLMNITRY 出力形式 (p. 184)	トルコの国際表示形式で通貨データを書き出します。
	NLMNITWD 出力形式 (p. 185)	台湾の国際表示形式で通貨データを書き出します。
	NLMNIUSD 出力形式 (p. 186)	プエルトリコと米国の国際表示形式で通貨データを書き出します。
	NLMNIZAR 出力形式 (p. 187)	南アフリカの国際表示形式で通貨データを書き出します。
	NLMNLAED 出力形式 (p. 188)	アラブ首長国連邦のローカル表示形式で通貨データを書き出します。
	NLMNLAUD 出力形式 (p. 189)	オーストラリアのローカル表示形式で通貨データを書き出します。

カテゴリ	言語要素	説明
	NLMNLBGN 出力形式 (p. 190)	ブルガリアのローカル表示形式で通貨データを書き出します。
	NLMNLBRL 出力形式 (p. 191)	ブラジルのローカル表示形式で通貨データを書き出します。
	NLMNLCAD 出力形式 (p. 192)	カナダのローカル表示形式で通貨データを書き出します。
	NLMNLCHF 出力形式 (p. 193)	リヒテンシュタインとスイスのローカル表示形式で通貨データを書き出します。
	NLMNLCNY 出力形式 (p. 194)	中国のローカル表示形式で通貨データを書き出します。
	NLMNLCZK 出力形式 (p. 195)	チェコ共和国のローカル表示形式で通貨データを書き出します。
	NLMNLDKK 出力形式 (p. 196)	デンマーク、フェロー諸島およびグリーンランドのローカル表示形式で通貨データを書き出します。
	NLMNLEEK 出力形式 (p. 197)	エストニアのローカル表示形式で通貨データを書き出します。
	NLMNLEGP 出力形式 (p. 198)	エジプトのローカル表示形式で通貨データを書き出します。
	NLMNLEUR 出力形式 (p. 199)	オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインのローカル表示形式で通貨データを書き出します。
	NLMNLGBP 出力形式 (p. 200)	英国のローカル表示形式で通貨データを書き出します。
	NLMNLHKD 出力形式 (p. 201)	香港のローカル表示形式で通貨データを書き出します。
	NLMNLHRK 出力形式 (p. 202)	クロアチアのローカル表示形式で通貨データを書き出します。
	NLMNLHUF 出力形式 (p. 203)	ハンガリーのローカル表示形式で通貨データを書き出します。
	NLMNLIDR 出力形式 (p. 204)	インドネシアのローカル表示形式で通貨データを書き出します。
	NLMNLILS 出力形式 (p. 205)	イスラエルのローカル表示形式で通貨データを書き出します。
	NLMNLINR 出力形式 (p. 206)	インドのローカル表示形式で通貨データを書き出します。

カテゴリ	言語要素	説明
	NLMNLJPY 出力形式 (p. 207)	日本の国際表示形式で通貨データを書き出します。
	NLMNLKRW 出力形式 (p. 208)	韓国のローカル表示形式で通貨データを書き出します。
	NLMNLLTL 出力形式 (p. 209)	リトアニアのローカル表示形式で通貨データを書き出します。
	NLMNLLVL 出力形式 (p. 210)	ラトビアのローカル表示形式で通貨データを書き出します。
	NLMNLMOP 出力形式 (p. 211)	マカオのローカル表示形式で通貨データを書き出します。
	NLMNLMXN 出力形式 (p. 212)	メキシコのローカル表示形式で通貨データを書き出します。
	NLMNLMYR 出力形式 (p. 213)	マレーシアのローカル表示形式で通貨データを書き出します。
	NLMNLNOK 出力形式 (p. 214)	ノルウェーのローカル表示形式で通貨データを書き出します。
	NLMNLNZD 出力形式 (p. 215)	ニュージーランドのローカル表示形式で通貨データを書き出します。
	NLMNLPLN 出力形式 (p. 216)	ポーランドのローカル表示形式で通貨データを書き出します。
	NLMNLRUB 出力形式 (p. 217)	ロシアのローカル表示形式で通貨データを書き出します。
	NLMNLSEK 出力形式 (p. 218)	スウェーデンのローカル表示形式で通貨データを書き出します。
	NLMNLSGD 出力形式 (p. 219)	シンガポールのローカル表示形式で通貨データを書き出します。
	NLMNLTHB 出力形式 (p. 220)	タイのローカル表示形式で通貨データを書き出します。
	NLMNLTRY 出力形式 (p. 221)	トルコのローカル表示形式で通貨データを書き出します。
	NLMNLTWD 出力形式 (p. 222)	台湾のローカル表示形式で通貨データを書き出します。
	NLMNLUSD 出力形式 (p. 223)	プエルトリコと米国のローカル表示形式で通貨データを書き出します。
	NLMNLZAR 出力形式 (p. 224)	南アフリカのローカル表示形式で通貨データを書き出します。

カテゴリ	言語要素	説明
	NLMNY 出力形式 (p. 225)	指定されたロケールで、ローカル通貨を使用してローカル表示形式で通貨データを書き出します。
	NLMNYI 出力形式 (p. 226)	指定されたロケールで、通貨データを国際表示形式で書き出します。
	NLNUM 出力形式 (p. 227)	指定されたロケールで、数値をローカル表示形式で書き出します。
	NLNUMI 出力形式 (p. 229)	指定されたロケールで、数値を国際表示形式で書き出します。
	NLPCT 出力形式 (p. 230)	指定されたロケールで、百分率のデータをローカル表示形式で書き出します。
	NLPCTI 出力形式 (p. 232)	指定されたロケールで、百分率のデータを国際表示形式で書き出します。
	NLPCTN 出力形式 (p. 233)	百分率を書き出します。負の値にはマイナス記号を使用します。
	NLPCTP 出力形式 (p. 234)	ロケール固有の数値を百分率として書き出します。
	NLPVALUE 出力形式 (p. 235)	指定されたロケールで、p 値をローカル表示形式で書き出します。
	NLSTRMON 出力形式 (p. 236)	指定されたロケールの月名を書き出します。
	NLSTRQTR 出力形式 (p. 237)	指定されたロケールの四半期の形式で数値を書き出します。
	NLSTRWK 出力形式 (p. 239)	指定されたロケールの曜日の形式で数値を書き出します。
	YEN 出力形式 (p. 270)	円記号、桁区切り、小数点付きの数値を出力します。
日付と時間	B8601DX 出力形式 (p. 82)	協定世界時(UTC)日時値をユーザーのローカル日付と時間に調整します。その後、ローカルの日付と時間は、ISO 8601 規格の日時とタイムゾーンの基本表記 <code>yyyymmddThhmmss+hhmm</code> を使用して書き出されます。
	B8601LX 出力形式 (p. 83)	ISO 8601 規格の基本表記 <code>yyyymmddThhmmss+ -hhmm</code> を使用して、ローカル時間と UTC 間のタイムゾーンオフセット差を追加することにより、日時値をローカル時間として書き出します。
	B8601TX 出力形式 (p. 85)	協定世界時(UTC)値をユーザーのローカル時間に調整します。その後、ISO 8601 規格の基本時間表記 <code>hhmmss+ -hhmm</code> を使用して、ローカル時間を書き出します。
	E8601DX 出力形式 (p. 91)	協定世界時(UTC)日時値をユーザーのローカル日付と時間に調整します。その後、ローカルの日付と時間は、ISO 8601 規格の日時とタイムゾーンの拡張表記 <code>yyyy-mm-ddThh:mm:ss+hh:mm</code> を使用して書き出されます。

カテゴリ	言語要素	説明
	E8601LX 出力形式 (p. 92)	ISO 8601 規格の拡張表記 yyyy-mm-ddThh:mm:ss+ -hh:mm を使用して、ローカル時間と UTC 間のタイムゾーンオフセット差を追加することにより、日時値をローカル時間として書き出します。
	E8601TX 出力形式 (p. 94)	協定世界時(UTC)値をユーザーのローカル時間に調整します。その後、ローカル時間は、ISO 8601 規格の日時とタイムゾーンの拡張表記 hh:mm:ss+ -hh:mm を使用して読み込まれます。
	MINGUO 出力形式 (p. 100)	SAS 日付値を台湾の日付として yyyymmdd 形式で書き出します。
	NENGO 出力形式 (p. 102)	SAS 日付値を日本の日付として e.yyymmdd 形式で書き出します。
	NLDATE 出力形式 (p. 105)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から日付を書き出します。
	NLDATEL 出力形式 (p. 106)	SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。
	NLDATEM 出力形式 (p. 107)	SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。
	NLDATEMD 出力形式 (p. 108)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から月の名前と月の日を書き出します。
	NLDATEMDL 出力形式 (p. 109)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。
	NLDATEMDM 出力形式 (p. 110)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。
	NLDATEMDS 出力形式 (p. 111)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。
	NLDATEMN 出力形式 (p. 112)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から月の名前を書き出します。
	NLDATES 出力形式 (p. 113)	SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。
	NLDATEW 出力形式 (p. 114)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から日付と曜日を書き出します。
	NLDATEWN 出力形式 (p. 115)	SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から曜日を書き出します。
	NLDATEYM 出力形式 (p. 116)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と月の名前を書き出します。
	NLDATEYML 出力形式 (p. 118)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。

カテゴリ	言語要素	説明
	NLDATEYMM 出力形式 (p. 118)	SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して月と年を日付値として書き出します。
	NLDATEYMS 出力形式 (p. 119)	SAS 日付値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。
	NLDATEYQ 出力形式 (p. 120)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と四半期を書き出します。
	NLDATEYQL 出力形式 (p. 121)	SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して年と四半期値(Q1-Q4)を書き出します。
	NLDATEYQM 出力形式 (p. 122)	SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して年と四半期値(Q1-Q4)を書き出します。
	NLDATEYQS 出力形式 (p. 123)	SAS 日付値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して年と四半期値(1-4)を書き出します。
	NLDATEYR 出力形式 (p. 124)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年を書き出します。
	NLDATEYW 出力形式 (p. 125)	SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と週を書き出します。
	NLDATM 出力形式 (p. 126)	SAS 日時値を、指定されたロケールの日時値に変換し、その日時値から日時を書き出します。
	NLDATMAP 出力形式 (p. 127)	SAS 日時値を、指定されたロケールの日時値に変換し、その日時値から午前または午後付きの日時を書き出します。
	NLDATMDT 出力形式 (p. 128)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前、月の日および年を書き出します。
	NLDATML 出力形式 (p. 129)	SAS 日時値を、指定されたロケールの日時文字列に変換し、月、日、年、時間を書き出します。
	NLDATMM 出力形式 (p. 130)	SAS 日時値を、指定されたロケールの日付文字列に変換し、月名と時間の短縮形を使用して月、日、年、時間を書き出します。
	NLDATMMD 出力形式 (p. 130)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前と月の日を書き出します。
	NLDATMMDL 出力形式 (p. 131)	SAS 日時値を、指定されたロケールの日付文字列に変換し、長い形式で月と日を書き出します。
	NLDATMMDM 出力形式 (p. 132)	SAS 日時値を、指定されたロケールの日付文字列に変換し、短縮形を使用して月と日を書き出します。
	NLDATMMDS 出力形式 (p. 133)	SAS 日時値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して月と日を書き出します。

カテゴリ	言語要素	説明
	NLDATMMN 出力形式 (p. 134)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前を書き出します。
	NLDATMS 出力形式 (p. 135)	SAS 日時値を、指定されたロケールの日付文字列に変換し、MM/dd/yyyy の形式で書き出します。
	NLDATMTM 出力形式 (p. 136)	SAS 日時値の時間部分を、指定されたロケールの時間値に変換し、その時間値から時刻を書き出します。
	NLDATMTZ 出力形式 (p. 137)	ロケールの SAS 日時値の時間部分を、時刻とタイムゾーンに変換します。
	NLDATMW 出力形式 (p. 138)	SAS 日時値を、曜日および日時形式の、ロケール依存の日時文字列に変換します。
	NLDATMWN 出力形式 (p. 139)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から曜日を書き出します。
	NLDATMWZ 出力形式 (p. 140)	指定されたロケールの SAS 日時値を、曜日、日時、およびタイムゾーン値に変換します。
	NLDATMYM 出力形式 (p. 141)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と月の名前を書き出します。
	NLDATMYML 出力形式 (p. 142)	SAS 日時値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。
	NLDATMYMM 出力形式 (p. 143)	SAS 日時値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。
	NLDATMYMS 出力形式 (p. 143)	SAS 日時値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して月と年を書き出します。
	NLDATMYQ 出力形式 (p. 144)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と四半期を書き出します。
	NLDATMYQL 出力形式 (p. 145)	SAS 日時値を、指定されたロケールの日付文字列に変換し、四半期(1-4)値、年の順に書き出します。
	NLDATMYQM 出力形式 (p. 146)	SAS 日時値を、指定されたロケールの日付文字列に変換し、四半期(1-4)、年の順に書き出します。
	NLDATMYQS 出力形式 (p. 147)	SAS 日時値を、指定されたロケールの日付文字列に変換し、年と四半期(1-4)を数字と区切り文字を使用して書き出します。
	NLDATMYR 出力形式 (p. 148)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年を書き出します。
	NLDATMYW 出力形式 (p. 149)	SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と週の名前を書き出します。
	NLDATMZ 出力形式 (p. 150)	SAS 日時値を、タイムゾーンおよび日時形式の、ロケール依存の日時文字列に変換します。

カテゴリ	言語要素	説明
	NLTIMAP 出力形式 (p. 240)	SAS 時間値を、指定されたロケールの時間値に変換し、その時間値から午前または午後付きの時間値を書き出します。NLTIMAP は、SAS 日時値も変換します。
	NLTIME 出力形式 (p. 241)	SAS 時間値を、指定されたロケールの時間値に変換し、その時間値から時刻を書き出します。NLTIME は、SAS 日時値も変換します。
	WEEKU 出力形式 (p. 265)	U アルゴリズムを使用して、10 進数の週番号を書き出します。
	WEEKV 出力形式 (p. 267)	V アルゴリズムを使用して、10 進数の週番号を書き出します。
	WEEKW 出力形式 (p. 268)	W アルゴリズムを使用して、10 進数の週番号を書き出します。
	YYWEEKU 出力形式 (p. 271)	曜日情報を除外し、U アルゴリズムを使用して、10 進数の週番号を書き出します。
	YYWEEKV 出力形式 (p. 272)	曜日情報を除外し、V アルゴリズムを使用して、10 進数の週番号を書き出します。
	YYWEEKW 出力形式 (p. 274)	曜日情報を除外し、W アルゴリズムを使用して、10 進数の週番号を書き出します。
ヘブライ語のテキストの操作	\$CPTDW 出力形式 (p. 88)	IBM-PC (cp862)でエンコードされたヘブライ語テキスト中の文字列を処理し、その文字列を Windows のヘブライ語エンコーディング(cp 1255)で書き出します。
	\$CPTWD 出力形式 (p. 89)	Windows でエンコード(cp1255)された文字列を処理し、その文字列をヘブライ語 DOS (cp862)エンコーディングで書き出します。
文字	\$EBCDIC 出力形式 (p. 90)	ネイティブな形式の文字データを EBCDIC 表現に変換します。
	\$UCS2B 出力形式 (p. 242)	現在の SAS セッションのエンコーディングを使用する文字列を処理し、ビッグエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を書き出します。
	\$UCS2BE 出力形式 (p. 243)	ビッグエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。
	\$UCS2L 出力形式 (p. 244)	現在の SAS セッションのエンコーディングを使用する文字列を処理し、リトルエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を書き出します。
	\$UCS2LE 出力形式 (p. 246)	リトルエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ	言語要素	説明
	\$UCS2X 出力形式 (p. 247)	現在の SAS セッションのエンコーディングを使用する文字列を処理し、ネイティブエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を書き出します。
	\$UCS2XE 出力形式 (p. 248)	ネイティブエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。
	\$UCS4B 出力形式 (p. 249)	現在の SAS セッションのエンコーディングを使用する文字列を処理し、ビッグエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を書き出します。
	\$UCS4BE 出力形式 (p. 251)	ビッグエンディアン、32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。
	\$UCS4L 出力形式 (p. 252)	現在の SAS セッションのエンコーディングを使用する文字列を処理し、リトルエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を書き出します。
	\$UCS4LE 出力形式 (p. 253)	リトルエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。
	\$UCS4X 出力形式 (p. 254)	現在の SAS セッションのエンコーディングを使用する文字列を処理し、ネイティブエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を書き出します。
	\$UCS4XE 出力形式 (p. 256)	ネイティブエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)で文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。
	\$UESC 出力形式 (p. 257)	現在の SAS セッションでエンコードされる文字列を処理し、その文字列を Unicode エスケープ(UESC)表記で書き出します。
	\$UESCE 出力形式 (p. 258)	Unicode エスケープ(UESC)表記の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。
	\$UNCR 出力形式 (p. 259)	現在の SAS セッションでエンコードされる文字列を処理し、その文字列を数値文字表現(NCR)で書き出します。
	\$UNCRE 出力形式 (p. 260)	数値文字表現(NCR)の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。
	\$UPAREN 出力形式 (p. 261)	現在の SAS セッションでエンコードされる文字列を処理し、その文字列を Unicode カッコ(UPAREN)表記で書き出します。
	\$UPARENE 出力形式 (p. 262)	Unicode カッコ(UPAREN)内の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ	言語要素	説明
	\$UTF8X 出力形式 (p. 263)	現在の SAS セッションのエンコーディングを使用する文字列を処理し、UTF-8 (Universal Transformation Format)エンコーディングの文字列を書き出します。
	\$UTF8XE 出力形式 (p. 264)	UTF-8 (Universal Transformation Format)の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

ディクショナリ

B8601DX 出力形式

協定世界時(UTC)日時値をユーザーのローカル日付と時間に調整します。その後、ローカルの日付と時間は、ISO 8601 規格の日時とタイムゾーンの基本表記 `yyyymmddThhmmss+hhmm` を使用して書き出されます。

カテゴリ:	日付と時間 ISO 8601
配置:	左
サポート:	ISO 8601 Element 5.4.1, complete representation

構文

B8601DX*w*.

構文の説明

w
出力幅を指定します。

デフォルト 26

範囲 20-35

詳細

UTC 値は、イギリスのグリニッジの基準子午線に基づく日付と時間を示します。SAS は、この出力形式を使用して、日時値を UTC 値に変換したり、TIMEZONE=システムオプションの値を使用することで、ユーザーローカルの日付と時間を決定したりします。TIMEZONE=オプションが設定されていない場合、ユーザーローカルの日付と時間は、ローカルの日付と時間に基づきます。B8601DX 出力形式は、SAS 日時値を次の ISO 8601 規格の基本日時表記を使用して書き出します。

- `yyyymmddThhmmss+hhmm`

yyyy
4 桁の年です。

mm
01 から 12 までの 2 桁の月です(ゼロ埋め込み)。

dd
01 から 31 までの 2 桁の日です(ゼロ埋め込み)。

hh
00 から 23 までの 2 桁の時間です(ゼロ埋め込み)。

mm
00 から 59 までの 2 桁の分です(ゼロ埋め込み)。

ss
00 から 59 までの 2 桁の秒です(ゼロ埋め込み)。

+|-hhmm
基準子午線時間からの時間と分の符号付きオフセットです。オフセットは *+|-hhmm* (つまり、+または-を伴う 4 文字)である必要があります。
+は基準子午線の東部タイムゾーンに、-は基準子午線の西部タイムゾーンにそれぞれ使用します。たとえば、+0200 は基準子午線の東部での 2 時間差を表し、-0600 は基準子午線の西部での 6 時間差を表します。

制約: 短い形式 *+|-hh* はサポートされていません。

例

最初の例では、時間とタイムゾーンオフセットの決定にローカル時間を使用します。2 番目の例では、タイムゾーンを America/Adak に変更します。それはハワイ-アリューシャン標準時です。

ステートメント	結果
<pre>data _null_; t='01Feb2013T12:34:56'dt; put t b8601dx.; run;</pre>	20130201T073456-0500
<pre>options timezone='America/Adak'; data _null_; t='01Feb2013T12:34:56'dt; put t b8601dx.; run;</pre>	20130201T023456-1000

関連項目:

[“Working with Dates and Times By Using the ISO 8601 Basic and Extended Notations” \(SAS Viya Formats and Informats: Reference\)](#)

B8601LX 出力形式

ISO 8601 規格の基本表記 *yyyymmddThhmmss+|-hhmm* を使用して、ローカル時間と UTC 間のタイムゾーンオフセット差を追加することにより、日時値をローカル時間として書き出します。

カテゴリ: 日付と時間
ISO 8601

配置: 右
サポート: ISO 8601 Elements 5.3.3 and 5.3.4.2

構文

B8601LX w

構文の説明

w

出力幅を指定します。

デフォルト 26

範囲 20-35

詳細

B8601LX 出力形式は、ISO 8601 規格の基本日時表記を使用し、時間値を調整せずに書き出してローカル SAS セッションに対応した UTC タイムゾーンオフセットを追加します。

- $yyyymmddThhmmss+|-hhmm$

$yyyy$

4桁の年です。

mm

01 から 12 までの 2 桁の月です(ゼロ埋め込み)。

dd

01 から 31 までの 2 桁の日です(ゼロ埋め込み)。

hh

00 から 23 までの 2 桁の時間です(ゼロ埋め込み)。

mm

00 から 59 までの 2 桁の分です(ゼロ埋め込み)。

ss

00 から 59 までの 2 桁の秒です(ゼロ埋め込み)。

$+|-hhmm$

基準子午線時間からの時間と分の符号付きオフセットです。オフセットは $+|-hhmm$ (つまり、+または-を伴う 4 文字)である必要があります。

+は基準子午線の東部タイムゾーンに、-は基準子午線の西部タイムゾーンにそれぞれ使用します。たとえば、+0200 は基準子午線の東部での 2 時間差を表し、-0600 は基準子午線の西部での 6 時間差を表します。

制約: 短い形式 $+|-hh$ はサポートされていません。

例

この PUT ステートメントは東部標準タイムゾーンに対応した時間を書き出します。

```
blx='01Feb2013T12:34:56'dt;
put blx b8601lx.;
```

blx の値	結果
1675341296	20130201T123456-0500

関連項目:

[“Working with Dates and Times By Using the ISO 8601 Basic and Extended Notations” \(SAS Viya Formats and Informats: Reference\)](#)

B8601TX 出力形式

協定世界時(UTC)値をユーザーのローカル時間に調整します。その後、ISO 8601 規格の基本時間表記 *hhmmss+|-hhmm* を使用して、ローカル時間を書き出します。

カテゴリ:	日付と時間 ISO 8601
配置:	右
サポート:	ISO 8601 Elements 5.3.3 and 5.3.4

構文

B8601TX*w*.

構文の説明

w
出力幅を指定します。

デフォルト 14

範囲 9-20

詳細

UTC 値は、イギリスのグリニッジの基準子午線に基づく時間を示します。SAS は、この出力形式を使用して、時間値を UTC 値に変換したり、TIMEZONE=システムオプションを使用することで、ユーザーのローカル時間を決定したりします。TIMEZONE=オプションが設定されていない場合、ユーザーのローカル時間は、ローカル時間に基づきます。B8601TX 出力形式は、SAS 日時値を次の ISO 8601 規格の基本時間表記を使用して書き出します。

- *hhmmss+|-hhmm*

hh
00 から 23 までの 2 桁の時間です(ゼロ埋め込み)。

mm
00 から 59 までの 2 桁の分です(ゼロ埋め込み)。

ss
00 から 59 までの 2 桁の秒です(ゼロ埋め込み)。

+|-hhmm

基準子午線時間からの時間と分の符号付きオフセットです。オフセットは +|-hhmm (つまり、+または-を伴う 4 文字) である必要があります。

+は基準子午線の東部タイムゾーンに、-は基準子午線の西部タイムゾーンにそれぞれ使用します。たとえば、+0200 は基準子午線の東部での 2 時間差を表し、-0600 は基準子午線の西部での 6 時間差を表します。

制約: 短い形式+|-hh はサポートされていません。

SAS では UTC 時間を B8601TZ 入力形式を使用して読み込む際に、オフセット調整後の時間が 24 時間を超えていたり 00 時間未満である場合、オフセット調整後の時間が 000000 と 240000 の間になるように値を調整します。B8601TX 出力形式をこの時間範囲外の時間に適用する場合、時間はアスタリスクを使用してフォーマットされ、その値が範囲外であることが示されます。

例

最初の例では、時間とタイムゾーンオフセットの決定にローカル時間を使用します。2 番目の例では、タイムゾーンを America/Adak に変更します。それはハワイ-アリューシャン標準時です。

ステートメント	結果
<pre>data _null_; t='12:34:56't; put t b8601tx.; run;</pre>	073456-0500
<pre>options timezone='America/Adak'; data _null_; t='12:34:56't; put t b8601tx.; run;</pre>	023456-1000

関連項目:

[“Working with Dates and Times By Using the ISO 8601 Basic and Extended Notations” \(SAS Viya Formats and Informats: Reference\)](#)

BESTDOTX 出力形式

最適な表記を SAS が選択し、ピリオド(.)を小数点として使用することを指定します。

カテゴリ: 数値

配置: 右

構文

BESTDOTX*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 1-32

ヒント 0 から.01 の数字(端点を除く)を出力する場合、過度に丸めないように 7 以上のフィールド幅を使用します。0 から-01 の数字(端点を除く)を出力する場合は、8 以上のフィールド幅を使用します。

詳細

NLDECSEPARATOR システムオプションを無効にすると、BEST w と BESTDOTX w の出力形式のデータの処理方法は同じになります。NLDECSEPARATOR システムオプションを有効にすると、BEST と BESTDOTX の出力形式の結果は異なります。この違いについて、次の表に示します。

LOCALE オプション	ロケールのデフォルトの小数点区切り文字	NLDECSEPARATOR オプション	BEST で使用される区切り文字 w.	BESTDOTX で使用される区切り文字 w.
en_US	ピリオド	無効(デフォルト)	ピリオド	ピリオド
		有効	ピリオド	ピリオド
fr_FR	カンマ	無効(デフォルト)	ピリオド	ピリオド
		有効	カンマ	ピリオド

例

次のコードでは、NLSDECSEPARATOR オプションを有効にして、BESTDOTX 出力形式を使用することで、結果を作成しています。ロケールは fr_FR です。

ステートメント	結果
options nld locale=fr_FR;	1,2
data _null_;	1.2
x=1.2;	
call symput('macx', put(x, BESTDOTX.));	
put x; /* Result is printed as "1,2" */	
y=put(x, BESTDOTX.); /*Result is printed as "1.2" */	
run;	
&put &macx;	1.2
data _null_;	1,2
bx=&macx; /* Succeeds because the value of &macx is 1.2 */	
put bx;	
run;	

関連項目:

- [“BEST Format” \(SAS Viya Formats and Informats: Reference\)](#)
- [“NLDECSEPARATOR システムオプション” \(556 ページ\)](#)

\$CPTDW 出力形式

IBM-PC (cp862)でエンコードされたヘブライ語テキスト中の文字列を処理し、その文字列を Windows のヘブライ語エンコーディング(cp 1255)で書き出します。

カテゴリ: ヘブライ語のテキストの操作

配置: 左

構文

\$CPTDW_w.

説明

w

出力フィールドの幅を指定します。

デフォルト 200

範囲 1-32767

比較

\$CPTDW_w.出力形式では、\$CPTWD_w.出力形式とは逆の処理を行います。

例

次の例では、入力値として"808182x"を使用します。

ステートメント	結果
	----+----1----+
put text \$cptdw3.;	808182x

関連項目:**出力形式:**

- [“\\$CPTWD 出力形式” \(89 ページ\)](#)

入力形式:

- [“\\$CPTDW 入力形式” \(414 ページ\)](#)
- [“\\$CPTWD 入力形式” \(415 ページ\)](#)

\$CPTWD 出力形式

Windows でエンコード(cp1255)された文字列を処理し、その文字列をヘブライ語 DOS (cp862)エンコーディングで書き出します。

カテゴリ: ヘブライ語のテキストの操作

配置: 左

構文

\$CPTWD*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 200

範囲 1-32767

比較

\$CPTWD*w*.出力形式では、\$CPTDW*w*.出力形式とは逆の処理を行います。

例

次の例では、入力値として“אבא”を使用します。

ステートメント	結果
	----+----1----+----2----+
put text \$cptwd3.;	אבא,

関連項目:

出力形式:

- [“\\$CPTDW 出力形式” \(88 ページ\)](#)

入力形式:

- [“\\$CPTDW 入力形式” \(414 ページ\)](#)
- [“\\$CPTWD 入力形式” \(415 ページ\)](#)

\$EBCDIC 出力形式

ネイティブな形式の文字データを EBCDIC 表現に変換します。

カテゴリ: 文字

配置: 左

注: サポートされているセッションエンコーディングは UTF-8 のみです。\$EBCDIC 出力形式は 7 ビット ASCII 文字でのみ動作します。

構文

\$EBCDIC*w*.

構文の説明

w

出力幅を指定します。

デフォルト 1

範囲 1-32767

詳細

EBCDIC がネイティブな形式の場合、変換は実行されません。

ASCII システムでは、\$EBCDIC*w*.出力形式は、SAS 起動時に指定された LOCALE= オプションのデフォルトエンコーディング値に基づいています。たとえば、ロケールが en_US ロケールに設定されていた場合、\$EBCDIC*w*.出力形式で使用されるデフォルトのエンコーディングは Open_ed-1047 です。ロケールが de_DE (German_Germany)である場合、\$EBCDIC*w*.出力形式で使用されるデフォルトのエンコーディングは Open_ed-1141 です。ロケールおよびエンコーディング値のリストについては、[表 17.2 \(606 ページ\)](#)を参照してください。

MAPEBCDIC2ASCII システムオプションを使用して、EBCDIC と ASCII 間での文字のマッピングに使用される変換テーブルを指定できます。詳細については、[“MAPEBCDIC2ASCII=システムオプション” \(555 ページ\)](#)を参照してください。

比較

- ASCII システムで \$EBCDIC*w*.を使用すると、ASCII 文字データは EBCDIC に変換されます。
- その他のすべてのシステムでは、\$EBCDIC*w*. は \$CHAR*w*.出力形式と同じ結果になります。

例

```
put name $ebcdic3.;
```


name の値	ロケール値が en_US の場合	ロケール値が fr_FR の場合	ロケール値が de_DE の場合
ABC	C1C2C3	C1C2C3	C1C2C3
{ä}	C043D0	514354	43C0DC
[@]	AD7CBD	9044B5	63B5FC

上記の結果は、EBCDIC 形式のコード値を 16 進表現で出力したものです。16 進数の 2 桁がバイナリデータの 1 バイトに相当し、1 バイトが 1 文字に相当します。

E8601DX 出力形式

協定世界時(UTC)日時値をユーザーのローカル日付と時間に調整します。その後、ローカルの日付と時間は、ISO 8601 規格の日時とタイムゾーンの拡張表記 `yyyy-mm-ddThh:mm:ss+hh:mm` を使用して書き出されます。

カテゴリ: 日付と時間
ISO 8601

配置: 左

サポート: ISO 8601 Element 5.4.1, complete representation

構文

E8601DX*w*.

構文の説明

w
出力幅を指定します。

デフォルト 26

範囲 20-35

詳細

UTC 値は、イギリスのグリニッジの基準子午線に基づく日付と時間を示します。SAS は、この出力形式を使用して、日時値を UTC 値に変換したり、TIMEZONE= システムオプションの値を使用することで、ユーザーローカルの日付と時間を決定したりします。TIMEZONE=オプションが設定されていない場合、ユーザーローカルの日付と時間は、ローカルの日付と時間に基づきます。E8601DX 出力形式は、SAS 日時値を次の ISO 8601 規格の基本日時表記を使用して書き出します。

- `yyyy-mm-ddThh:mm:ss+hh:mm`

yyyy
4 桁の年です。

mm
01 から 12 までの 2 桁の月です(ゼロ埋め込み)。

dd

01 から 31 までの 2 桁の日です(ゼロ埋め込み)。

hh

00 から 23 までの 2 桁の時間です(ゼロ埋め込み)。

mm

00 から 59 までの 2 桁の分です(ゼロ埋め込み)。

ss

00 から 59 までの 2 桁の秒です(ゼロ埋め込み)。

+|-hhmm

基準子午線時間からの時間と分の符号付きオフセットです。オフセットは +|-hh:mm (つまり、+または-を伴う 5 文字)である必要があります。

+は基準子午線の東部タイムゾーンに、-は基準子午線の西部タイムゾーンにそれぞれ使用します。たとえば、+02:00 は基準子午線の東部での 2 時間差を表し、-06:00 は基準子午線の西部での 6 時間差を表します。

制約: 短い形式+|-hh はサポートされていません。

例

最初の例では、時間の決定にローカル時間を使用します。2 番目の例では、タイムゾーンを America/Adak に変更します。それはハワイ-アリューシャン標準時です。

ステートメント	結果
<pre>data _null_; t='01Feb2013T12:34:56'dt; put t e8601dx.; run;</pre>	2013-02-01T07:34:56-05:00
<pre>options timezone='America/Adak'; data _null_; t='01Feb2013T12:34:56'dt; put t e8601dx.; run;</pre>	2013-02-01T02:34:56-10:00

関連項目:

[“Working with Dates and Times By Using the ISO 8601 Basic and Extended Notations” \(SAS Viya Formats and Informats: Reference\)](#)

E8601LX 出力形式

ISO 8601 規格の拡張表記 `yyyy-mm-ddThh:mm:ss+|-hh:mm` を使用して、ローカル時間と UTC 間のタイムゾーンオフセット差を追加することにより、日時値をローカル時間として書き出します。**カテゴリ:** 日付と時間

ISO 8601

配置: 右**サポート:** ISO 8601 Elements 5.3.3 and 5.3.4.2

構文

E8601LX w

構文の説明

w

出力幅を指定します。

デフォルト 26

範囲 20-35

詳細

E8601LX 出力形式は、ISO 8601 規格の基本日時表記を使用し、時間値を調整せず
に書き出してローカル SAS セッションに対応した UTC タイムゾーンオフセット
を追加します。

- $yyyy-mm-ddThh:mm:ss+hh:mm$

$yyyy$

4 桁の年です。

mm

01 から 12 までの 2 桁の月です(ゼロ埋め込み)。

dd

01 から 31 までの 2 桁の日です(ゼロ埋め込み)。

hh

00 から 23 までの 2 桁の時間です(ゼロ埋め込み)。

mm

00 から 59 までの 2 桁の分です(ゼロ埋め込み)。

ss

00 から 59 までの 2 桁の秒です(ゼロ埋め込み)。

$+|-hhmm$

基準子午線時間からの時間と分の符号付きオフセットです。オフセットは
 $+|-hh:mm$ (つまり、+または-を伴う 5 文字)である必要があります。

+は基準子午線の東部タイムゾーンに、-は基準子午線の西部タイムゾーンに
それぞれ使用します。たとえば、+02:00 は基準子午線の東部での 2 時間差を
表し、-06:00 は基準子午線の西部での 6 時間差を表します。

制約: 短い形式 $+|-hh$ はサポートされていません。

例

この PUT ステートメントは東部タイムゾーンに対応した時間を書き出します。

```
blx='01Feb2013T12:34:56'dt;
put blx e8601lx.;
```

blx の値	結果
1675341296	2013-02-01T12:34:56-05:00

関連項目:

[“Working with Dates and Times By Using the ISO 8601 Basic and Extended Notations” \(SAS Viya Formats and Informats: Reference\)](#)

E8601TX 出力形式

協定世界時(UTC)値をユーザーのローカル時間に調整します。その後、ローカルの時間は、ISO 8601 規格の日時とタイムゾーンの拡張表記 *hh:mm:ss+|-hh:mm* を使用して読み込まれます。

- カテゴリ:** 日付と時間
ISO 8601
- 配置:** 右
- サポート:** ISO 8601 Elements 5.3.3 and 5.3.4

構文

E8601TX*w*.

構文の説明

w
出力幅を指定します。

デフォルト 14

範囲 9-20

詳細

UTC 値は、イギリスのグリニッジの基準子午線に基づく時間を示します。SAS は、この出力形式を使用して、時間値を UTC 値に変換したり、TIMEZONE=システムオプションの値を使用することで、ユーザーのローカル時間を決定したりします。TIMEZONE=オプションが設定されていない場合、ユーザーのローカル時間は、ローカル時間に基づきます。E8601TX 出力形式は、SAS 日時値を次の ISO 8601 規格の基本時間表記を使用して書き出します。

- *hh:mm:ss+|-hh:mm*

hh
00 から 23 までの 2 桁の時間です(ゼロ埋め込み)。

mm
00 から 59 までの 2 桁の分です(ゼロ埋め込み)。

ss
00 から 59 までの 2 桁の秒です(ゼロ埋め込み)。

+|-hhmm

基準子午線時間からの時間と分の符号付きオフセットです。オフセットは +|-hhmm (つまり、+または-を伴う 4 文字) である必要があります。

+は基準子午線の東部タイムゾーンに、-は基準子午線の西部タイムゾーンにそれぞれ使用します。たとえば、+0200 は基準子午線の東部での 2 時間差を表し、-0600 は基準子午線の西部での 6 時間差を表します。

制約: 短い形式+|-hh はサポートされていません。

SAS では UTC 時間を E8601TZ 入力形式を使用して読み込む際に、オフセット調整後の時間が 24 時間を超えていたり 00 時間未満である場合、オフセット調整後の時間が 000000 と 240000 の間になるように値を調整します。E8601TX 出力形式をこの時間範囲外の時間に適用する場合、時間はアスタリスクを使用してフォーマットされ、その値が範囲外であることが示されます。

例

最初の例では、時間とタイムゾーンオフセットの決定にローカル時間を使用します。2 番目の例では、タイムゾーンを America/Adak に変更します。それはハワイ-アリューシャン標準時です。

ステートメント	結果
<pre>data _null_; t='12:34:56't; put t e8601tx.; run;</pre>	07:34:56-05:00
<pre>options timezone='America/Adak'; data _null_; t='12:34:56't; put t e8601tx.; run;</pre>	02:34:56-10:00

関連項目:

[“Working with Dates and Times By Using the ISO 8601 Basic and Extended Notations” \(SAS Viya Formats and Informats: Reference\)](#)

EURO 出力形式

先頭にユーロ(E)記号を付け、3 桁ごとにカンマを入れ、小数部分との区切りにはピリオドを使用して数値を出力します。

カテゴリ: 数値

CAS

配置: 右

構文

EURO*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 1-32

ヒント 出力にユーロ記号を含める場合、適切な幅を選択していることを確認してください。

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-31

要件 *w* より小さい数にします。

比較

- EURO*w.d* 出力形式は EUROX*w.d* 出力形式に似ていますが、EUROX*w.d* 出力形式は小数点とカンマの役割が逆になっています。この規則はヨーロッパの国で共通です。
- EURO*w.d* 出力形式は DOLLAR*w.d* 出力形式に似ていますが、DOLLAR*w.d* 出力形式は先頭にユーロ記号のかわりにドル記号を付けます。

例

この例では、金額の値に 1254.71 を使用します。

ステートメント	結果
	----+----1----+----2----+----3
put amount euro10.2;	E1,254.71
put amount euro5.;	E1255
put amount euro9.2;	E1,254.71
put amount euro15.3;	E1,254.710

```

data _null_;
input x;
put x euro10.2;
put x euro5.;;
put x euro9.2;
put x euro15.3;
datalines;
1254.71
;

```

```

run;
SAS Log:
E1,254.71
E1255
E1,254.71
  E1,254.710

  /* This code determines the default length. */
  data _null_;
    input x;
    put x euro.;
    datalines;
1
22
333
4444
55555
666666
7777777
88888888
999999999
1234561234
;run;
SAS Log:
  datalines;
  E1
  E22
  E333
  E4,444
  55,555
  666666
  7.78E6
  8.89E7
  1E9
  1.23E9
  NOTE: At least one W.D format was too small for the number to be printed.
  The decimal may be shifted by the "BEST" format.

/* This code determines the range. */
data _null_;
input x;
put x euro5.;
put x euro6.;
put x euro7.;
put x euro8.;
put x euro9.;
put x euro9.2;
put x euro10.;
put x euro10.2;
put x euro10.4;
put x euro11.;
put x euro11.3;
put x euro12.;
put x euro12.2;
put x euro13.;
put x euro13.2;
datalines;

```

333
 4444
 55555
 666666
 7777777
 88888888
 999999999
 1234561234
 ;run;

関連項目:

出力形式:

- [“EUROX 出力形式” \(98 ページ\)](#)

入力形式:

- [“EURO 入力形式” \(416 ページ\)](#)
- [“EUROX 入力形式” \(418 ページ\)](#)

EUROX 出力形式

先頭にユーロ(E)記号を付け、3桁ごとにピリオドを入れ、小数部分との区切りにはカンマを使用して数値を出力します。

カテゴリ: 数値
 CAS
配置: 右

構文

EUROX $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 1-32

ヒント

出力にユーロ記号を含める場合、適切な幅を選択していることを確認してください。

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-31

要件 w より小さい数にします。

比較

- EUROX $w.d$ 出力形式は EUROW d 出力形式に似ていますが、EUROW d 出力形式では、カンマと小数点の役割が逆になっています。この規則は英語圏の国で共通です。
- EUROX $w.d$ 出力形式は DOLLARX $w.d$ 出力形式に似ていますが、DOLLARX $w.d$ 出力形式は先頭にユーロ記号のかわりにドル記号を付けます。

例

この例では、金額の値に 1254.71 を使用します。

ステートメント	結果
	----+----1----+----2----+----3
put amount eurox10.2;	E1.254,71
put amount eurox5.;	E1255
put amount eurox9.2;	E1.254,71
put amount eurox15.3;	E1.254,710

```
data _null_;
  input x;
  put x eurox10.2;
  put x eurox5.;;
  put x eurox9.2;
  put x eurox15.3;
  datalines;
1254.71
; run;
SAS Log:
E1.254,71
E1255
E1.254,71
  E1.254,710

/* This code determines the default length. */
data _null_;
  input x;
  put x eurox.;
  datalines;
1
22
333
4444
55555
666666
7777777
```

```

88888888
999999999
1234561234
;run;
SAS Log:
  E1
  E22
  E333
E4,444
55.555
666666
7.78E6
8.89E7
  1E9
1.23E9

```

注: 1つ以上の W.D 出力形式において、数値を出力するには幅の長さが十分ではありませんでした。"BEST"出力形式が使用されて、10進数のシフトが行われた可能性があります。

関連項目:

出力形式:

- ["EURO 出力形式" \(95 ページ\)](#)

入力形式:

- ["EURO 入力形式" \(416 ページ\)](#)
- ["EUROX 入力形式" \(418 ページ\)](#)

MINGUO 出力形式

SAS 日付値を台湾の日付として *yyyymmdd* 形式で書き出します。

カテゴリ: 日付と時間

配置: 左

構文

MINGUO*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 1-10

詳細

MINGUOw.出力形式では、SAS 日付値を *yyyymmdd* 形式で書き出します。

yyyy

年を表す整数です。

mm

月を表す整数です。

dd

月の日を表す整数です。

台湾のカレンダーでは、1912 年を基準年(1912 年 1 月 1 日が 01/01/01)として使用します。1912 年より前の日付は、一連のアスタリスクで表示されます。年の値は 100 年周期で繰り返されることはありません。年の値は、継続して増加していきます。

例

例の表では、次の入力値を使用します。

- 12054 は、1993 年 1 月 1 日に対応する SAS 日付値です。
- 18993 は、2012 年 1 月 1 日に対応する SAS 日付値です。
- -20088 は、1905 年 1 月 1 日に対応する SAS 日付値です。

ステートメント	結果
<code>x=put(12054,minguo7.); put x=;</code>	x=820101
<code>x=put(12054,minguo9.); put x=;</code>	x=82/01/01
<code>x=put(12054,minguo10.); put x=;</code>	x=0082/01/01
<code>x=put(18993,minguo7.); put x=;</code>	x=1000101
<code>x=put(18993,minguo9.); put x=;</code>	x=100/01/01
<code>x=put(18993,minguo10.); put x=;</code>	x=0101/01/01
<code>x=put(-20088,minguo7.); put x=;</code>	*****
<code>x=put(-20088,minguo9.); put x=;</code>	*****
<code>x=put(-20088,minguo10.); put x=;</code>	*****

関連項目:**入力形式:**

- [“MINGUO 入力形式” \(419 ページ\)](#)

NENGO 出力形式

SAS 日付値を日本の日付として *e.yymmdd* 形式で書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NENGO*w*.

説明

w 出力フィールドの幅を指定します。

デフォルト 10

範囲 2-10

詳細

NENGO*w*.出力形式では、SAS 日付値を *e.yymmdd* 形式で書き出します

e 元号(明治、大正、昭和、平成)の先頭の文字(M、T、S、H)です。

yy 年を表す整数です。

mm 月を表す整数です。

dd 月の日を表す整数です。

幅が小さすぎる場合は、ピリオドが省略されます。

例

例の表では、15342 を入力値として使用します。この値は 2002 年 1 月 2 日に相当する SAS 日付値です。

```
data _null_;
  date=15342;
  put date nengo3;
  put date nengo6;
  put date nengo8;
```

```
put date nengo9.;
put date nengo10.;
run
```

ステートメント	結果
	----+----1
put date nengo3.;	H14
put date nengo6.;	H14/01
put date nengo8.;	H.140102
put date nengo9.;	H14/01/02
put date nengo10.;	H.14/01/02

関連項目:

入力形式:

- [“NENGO 入力形式” \(421 ページ\)](#)

NLBEST 出力形式

ロケールに基づいて最適な数値表記を書き出します。

カテゴリ: 数値

CAS

配置: 右

構文

NLBEST w .

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 1-32

ヒント 0 から.01 の数字(端点を除く)を出力する場合、過度に丸めないように 7 以上のフィールド幅を使用します。0 から-.01 の数字(端点を除く)を出力する場合は、8 以上のフィールド幅を使用します。

詳細

NLBEST 出力形式は、ロケールの小数点と記号の位置に基づいて、最も適切な数値を書き出します。NLBEST は BEST 出力形式に似ています。詳細については、*SAS Viya 出力形式と入力形式: リファレンス*の BEST 出力形式を参照してください。

例

次に、コードのロケールに基づいた結果を示します。

```
x=-1257000
put x nlbest6;
put x nlbest3;
put "=====";
x=-0.1
put x nlbest6;
put x nlbest3;
put "=====";
x=0.1
put x nlbest6;
put x nlbest3;
put "=====";
x=1257000
put x nlbest6;
put x nlbest3;
```

ロケール	結果
locale=English_UnitedStates	-126E4

	=====
	-0.1
	-.1
	=====
	0.1
	0.1
	=====
	1.26E6
	1E6

ロケール	結果
locale=German_Germany	-126E4 *** ===== -0,1 -,1 ===== 0,1 0,1 ===== 1,26E6 1E6
locale=ar_BH	126E4- *** ===== 0.1- .1- ===== 0,1 0,1 ===== 1.26E6 1E6

NLDATE 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から日付を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

注: CAS サーバーでは WORDDATE 出力形式のかわりに NLDATE 出力形式を使用します。

構文

NLDATE*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

注: 指定の幅に基づいて、LONG、MEDIUM、SHORT などの出力形式パターンが使用されます。

デフォルト 20

範囲 10-200

比較

NLDATEw.は DATEw.および WORDDATEw.と似ていますが、NLDATEw.がロケール固有である点が違います。

例

これらの例では、15760 を入力値として使用します。この値は、2003年2月24日に相当する SAS 日付値です。

ステートメント	結果
	----+----1----+----2
options locale=English_UnitedStates; put day nldate.;	February 24, 2003
options locale=German_Germany; put day nldate.;	24. Februar 2003

関連項目:

出力形式:

- [“NLDATEMN 出力形式” \(112 ページ\)](#)
- [“NLDATEW 出力形式” \(114 ページ\)](#)
- [“NLDATEWN 出力形式” \(115 ページ\)](#)

NLDATEL 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEL*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 18

範囲 2-220

詳細

NLDATEL では、長い形式で月と週の名前を含むが出力されます。

例

この例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
<code>data _null;</code>	<code>+--- NLDATEL min=2 default=18 max=200 ---+</code>
<code>dt = datetime();</code>	November 19, 2012
<code>dy = date();</code>	11/19/2012
<code>put "+--- NLDATEL min=2 default=18</code>	Nov 19, 2012
<code>max=200 ---+";</code>	November 19, 2012
<code>put dy nldatel;</code>	November 19, 2012
<code>put dy nldatel10;</code>	
<code>put dy nldatel12;</code>	
<code>put dy nldatel18;</code>	
<code>put dy nldatel200;</code>	
<code>run;</code>	

NLDATEM 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEM*w*

説明

W

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 14

範囲 2-200

詳細

NLDATEM では、Nov 19, 2012 など、簡略形式で日付が出力されます。

例

この例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATEM min=2 default=14 max=200 ---+"; put dy nldatem.; put dy nldatem8.; put dy nldatem14.; put dy nldatem200.; run;	+--- NLDATEM min=2 default=14 max=200 ---+ Nov 19, 2012 11/19/12 Nov 19, 2012 Nov 19, 2012

NLDATEMD 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その値から月の名前と月の日を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEMD*w*.

説明

W

出力フィールドの幅を指定します。

デフォルト 16

範囲 6-200

例

この例では、en_US ロケールオプションを使用します。

ステートメント	結果
put 1 nldatmd;	January 02

関連項目:

出力形式:

- [“NLDATMDL 出力形式” \(116 ページ\)](#)

NLDATMDL 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMDL*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 12

範囲 5-200

詳細

NLDATMDL では、November 19 など、長い形式で月と日を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
<pre>data _null_; dt = datetime(); dy = date(); put "+--- NLDATMDL min=5 default=12 max=200 ---+"; put dy nldatmdl; put dy nldatmdl5; put dy nldatmdl9; put dy nldatmdl12; put dy nldatmdl200; run;</pre>	<pre>+--- NLDATMDL min=5 default=12 max=200 ---+ November 19 11/19 Nov 19 November 19 November 19 November 19 November 19 November 19</pre>

NLDATMDM 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMDM*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 9

範囲 5-200

詳細

NLDATMDM では、Nov 19 など、短縮形を使用した簡略形式で月と日を含む日付が出力されます。月名が数字に短縮される場合は、区切り文字が使用されません。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATEMDM min=5 default=9 max=200 ---+
dt = datetime();	Nov 19
dy = date();	11/19
put "+--- NLDATEMDM min=5 default=9	Nov 19
max=200 ---+";	Nov 19
put dy nldatmdm.;	
put dy nldatmdm5.;	
put dy nldatmdm9.;	
put dy nldatmdm200.;	
run;	

NLDATEMDS 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、月と日を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEMDS*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 5

範囲 5-200

詳細

NLDATEMDS では、MM/dd など、数字と区切り文字を使用した短い形式で月と日を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATEMDS min=5 default=5 max=200 ---+
dt = datetime();	11/19
dy = date();	11/19
put "+--- NLDATEMDS min=5 default=5	11/19
max=200 ---+";	11/19
put dy nldatemds.;	
put dy nldatemds5.;	
put dy nldatemds5.;	
put dy nldatemds200.;	
run;	

NLDATEMN 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から月の名前を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEMN*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように月名が短縮されます。

デフォルト 9

範囲 4-200

比較

NLDATEMN*w*.は MONNAME*w*.と似ていますが、NLDATEMN*w*.がロケール固有である点が違います。

例

これらの例では、15760 を入力値として使用します。この値は、2003 年 2 月 24 日に相当する SAS 日付値です。

ステートメント	結果
	---+----1

ステートメント	結果
options locale=English_UnitedStates; put month nldatemn.;	February
options locale=German_Germany; put month nldatemn.;	Februar

関連項目:

出力形式:

- [“NLDATE 出力形式” \(105 ページ\)](#)
- [“NLDATEW 出力形式” \(114 ページ\)](#)
- [“NLDATEWN 出力形式” \(115 ページ\)](#)

NLDATES 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、日、月、年を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATES*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 10

範囲 2-200

詳細

NLDATES では、mm/dd/yyyy など、数字と区切り文字のみ使用した短い形式で日付が出力されます。

例

この例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATES min=2 default=10 max=200 ---+
dt = datetime();	11/19/2012
dy = date();	11/19/12
put "+--- NLDATES min=2 default=10	11/19/2012
max=200 ---+";	11/19/2012
put dy nldates.;	
put dy nldates8.;	
put dy nldates10.;	
put dy nldates200.;	
run;	

NLDATEW 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から日付と曜日を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

注: CAS サーバーでは WEEKDATE 出力形式のかわりに NLDATEW 出力形式を使用します。

構文

NLDATEW*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付と曜日が短縮されます。

デフォルト 29

範囲 10-200

詳細

エンコーディングとロケールの組み合わせによっては日時名が長くなるため、デフォルトの幅を使用した NLDATEW 出力形式では正確でない地域化された出力が生成される場合があります。ロケールとエンコーディングの組み合わせに対する推奨される幅については、[日付と時間のデフォルト幅の例外 \(60 ページ\)](#) を参照してください。推奨される幅を使用するようにします。

比較

NLDATEW*w* は WEEKDATE*w* に似ていますが、NLDATEW*w* がロケール固有である点が違います。

例

これらの例では、15760 を入力値として使用します。この値は、2003 年 2 月 24 日に相当する SAS 日付値です。

ステートメント	結果
	----+----1-----2
options locale=English_UnitedStates;	Monday, February 24, 2003
date=15760;	Mon, Feb 24, 2003
x=put(date,nldatew.);	Monday, February 24, 2003
y=put(date,nldatew20.);	
z=put(date,nldatew200.);	
run;	
options locale=German_Germany;	Mo., 24. Feb 2003
date=15760;	
x=put(date,nldatew.);	Mo., 24. Feb 2003
y=put(date,nldatew20.);	
z=put(date,nldatew200.);	Montag, 24. Februar 2003
run;	

関連項目:

出力形式:

- [“NLDATE 出力形式” \(105 ページ\)](#)
- [“NLDATEMN 出力形式” \(112 ページ\)](#)
- [“NLDATEWN 出力形式” \(115 ページ\)](#)

NLDATEWN 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その日付値から曜日を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATEWN*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように曜日が短縮されます。

デフォルト 9

範囲 4-200

比較

NLDATEWN_wは DOWNAME_wと似ていますが、NLDATEWN_wがロケール固有である点が違います。

例

これらの例では、15760 を入力値として使用します。この値は、2003 年 2 月 24 日に相当する SAS 日付値です。

ステートメント	結果
	----+----1
options locale=English_UnitedStates; put date nldatewn.;	Monday
options locale=German_Germany; put date nldatewn.;	Montag

関連項目:

出力形式:

- [“NLDATE 出力形式” \(105 ページ\)](#)
- [“NLDATEMN 出力形式” \(112 ページ\)](#)
- [“NLDATEW 出力形式” \(114 ページ\)](#)

NLDATEYM 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と月の名前を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEYM_w.

説明

w

出力フィールドの幅を指定します。

デフォルト 16

範囲 6-200

詳細

データが指定した幅の 6 より大きい場合、出力にはアスタリスクが含まれます。**d=*******. アスタリスクを削除するには PROC LOCALEDATA を使用します。次の例では PROC LOCALEDATA を使用してアスタリスクなしで日付を出力しています。

```
PROC LOCALEDATA;
  LOAD SASLOCALE;
  MODIFY key=DATE_YYMM_SHORT_FORMAT value='%b %y';
  SAVE REGISTRY / _ALL_syntax=SAS;
```

```
data _null_;
  format d nldateym6.;
  d = '17OCT14'd;
  put d=;
  run;
```

```
d=Oct 14
```

例

この例では、ロケールオプションとして spanish_Spain、日付として 2010 年 8 月を使用します。

ステートメント	結果
options locale=spanish_Spain;	
data _null_;	
dy=today();	agosto de 2010
x=put(dy, nldateym.);	ago de 10
y=put(dy, nldateym12.);	agosto de 2010
z=put(dy, nldateym200.);	
run;	

関連項目:

出力形式:

- [“NLDATEMD 出力形式” \(108 ページ\)](#)

NLDATEYML 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEYML*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 14

範囲 5-200

詳細

NLDATEYML では、November 2012 など、長い形式で年と月を含む日付が出力されます。出力形式の幅に合わせて短縮形が使用されることもあります。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATEYML min=5 default=14 max=200 ---+"; put dy nldateyml; put dy nldateyml7; put dy nldateyml11; put dy nldateyml14; put dy nldateyml200; run;	+--- NLDATEYML min=5 default=14 max=200 ---+ November 2012 11/2012 Nov 2012 November 2012 November 2012

NLDATEYMM 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して月と年を日付値として書き出します。

カテゴリ: 日付と時間
CAS
配置: 左

構文

NLDATEYMM*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 11

範囲 5-200

詳細

NLDATEYMM では、Nov 2012 など、短縮形を使用した簡略形式で月と年を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATEYMM min=5 default=11 max=200 ---+"; put dy nldateymm; put dy nldateymm7; put dy nldateymm11; put dy nldateymm200; run;	+--- NLDATEYMM min=5 default=11 max=200 ---+ Nov 2012 11/2012 Nov 2012 Nov 2012

NLDATEYMS 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。

カテゴリ: 日付と時間
CAS
配置: 左

構文

NLDATEYMS*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 7

範囲 5-200

詳細

NLDATEYMS では、mm/yyyy など、数字と区切り文字を使用した短い形式で年と月を含む日付が出力されます。

例

この例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null;	+--- NLDATEYMS min=5 default=7 max=200 ---+
dt = datetime();	11/2012
dy = date();	11/12
put "+--- NLDATEYMS min=5 default=7	11/2012
max=200 ---+";	11/2012
put dy nldateyms.;	
put dy nldateyms5.;	
put dy nldateyms7.;	
put dy nldateyms200.;	
run;	

NLDATEYQ 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と四半期を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEYQ*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト	16
範囲	4-200

詳細

エンコーディングとロケールの組み合わせによっては日時名が長くなるため、デフォルトの幅を使用した NLDATYQL 出力形式では正確でない地域化された出力が生成される場合があります。ロケールとエンコーディングの組み合わせに対する推奨される幅については、[日付と時間のデフォルト幅の例外 \(60 ページ\)](#) を参照してください。推奨される幅を使用するようにします。

例

この例では、fr_FR ロケールオプションを使用します。

ステートメント	結果
options locale=fr_FR;	
data _null_;	+--- NLDATYQL min=4 default=16
dy=today();	max=200 ---+
dt=datetime();	16 T3 08
put "+--- NLDATYQL min=4 default=16	4 ****
max=200 ---+";	14 T3 08
put ' 16' +5 dy nldateyq.;	32 3e trimestre 2008
put ' 4' +5 dy nldateyq4.;	200
put ' 14' +5 dy nldateyq14.;	3e trimestre 2008
put ' 32' +5 dy nldateyq32.;	
put ' 200' +5 dy nldateyq200.;	
run;	

NLDATYQL 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して年と四半期値(Q1-Q4)を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATYQLw.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 18

範囲 4-200

詳細

NLDATEYQL では、4th quarter 2012 など、長い形式で年と四半期を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATEYQL min=4 default=18 max=200 ---+"; put dy nldateyql; put dy nldateyql6.; put dy nldateyql7.; put dy nldateyql18.; put dy nldateyql200.; run;	+--- NLDATEYQL min=4 default=18 max=200 ---+ 4th quarter 2013 2013/4 Q4 2013 4th quarter 2013 4th quarter 2013

NLDATEYQM 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、短縮形を使用して年と四半期値(Q1-Q4)を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATEYQM*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 7
 範囲 4-200

詳細

NLDATEYQM では、Q4 2012 など、短縮形を使用した簡略形式で年と四半期を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATEYQM min=4 default=7 max=200 ---+"; put dy nldateyqm.; put dy nldateyqm6.; put dy nldateyqm7.; put dy nldateyqm200.; run;	+--- NLDATEYQM min=4 default=7 max=200 ---+ Q4 2012 2012/4 Q4 2012 Q4 2012

NLDATEYQS 出力形式

SAS 日付値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して年と四半期値(1-4)を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATEYQS*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 6
 範囲 4-200

詳細

NLDATEYQS では、2012/4 など、数字と区切り文字を使用した短い形式で年と四半期を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null;	+--- NLDATEYQS min=4 default=6 max=200 ---+
dt = datetime();	2012/4
dy = date();	12/4
put "+--- NLDATEYQS min=4 default=6	2012/4
max=200 ---+";	2012/4
put dy nldateyqs;	
put dy nldateyqs4;	
put dy nldateyqs6;	
put dy nldateyqs200;	
run;	

NLDATEYR 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その値から年を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATEYR*w*.

説明

w 出力フィールドの幅を指定します。

デフォルト 16

範囲 2-200

例

この例では、fr_FR ロケールオプションを使用します。

ステートメント	結果
options locale=fr_FR;	
data _null_;	+--- NLDATYEW min=2 default=16
dy=today();	max=200 ---+
dt=datetime();	2008
put "+--- NLDATYEW min=2 default=16	08
max=200 ---+";	2008
put dy nldateyr.;	2008
put dy nldateyr2.;	
put dy nldateyr8.;	
put dy nldateyr200.;	
run;	

NLDATYEW 出力形式

SAS 日付値を、指定されたロケールの日付値に変換し、その値から年と週を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATYEW*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 16

範囲 5-200

例

この例では、fr_FR ロケールオプションを使用します。

ステートメント	結果
options locale=fr_FR;	
data _null_;	16 Week 33 2008
dy=today();	5 *****
dt=datetime();	8 W33 08
put "+--- NLDATM min=5 default=16 max=200 ---+";	32 Week 33 2008
put ' 16' +5 dy nldatm;	200
put ' 5' +5 dy nldatm5;	Week 33 2008
put ' 8' +5 dy nldatm8;	
put ' 32' +5 dy nldatm32;	
put ' 200' +5 dy nldatm200;	
run;	

NLDATM 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その日時値から日時を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATM_w.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日時値が短縮されます。

デフォルト 30

範囲 10-200

比較

The NLDATM_w.出力形式は DATETIME_w.出力形式と似ていますが、NLDATM_w.出力形式がロケール固有である点が違います。

例

これらの例では、1361709583 を入力値として使用します。この値は、2003 年 2 月 24 日の午後 12:39:43 に相当する SAS 日時値です。

ステートメント	結果
	----+----1-----2----+----3
options locale=English_UnitedStates; put day nldatm.;	24Feb2003:12:39:43
options locale=German_Germany; put day nldatm.;	24. Februar 2003 12.39 Uhr

関連項目:

出力形式:

- [“NLDATMAP 出力形式” \(127 ページ\)](#)
- [“NLDATMTM 出力形式” \(136 ページ\)](#)
- [“NLDATMW 出力形式” \(138 ページ\)](#)

NLDATMAP 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その日時値から午前または午後付きの日時を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMAP_w.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日時値が短縮されます。

デフォルト 32

範囲 16-200

比較

NLDATMAP_w.出力形式は DATEAMP_w.と似ていますが、NLDATMAP_w.出力形式がロケール固有である点が違います。

例

これらの例では、1361709583 を入力値として使用します。この値は、2003 年 2 月 24 日の午後 12:39:43 に相当する SAS 日時値です。

ステートメント	結果
	----t----1-----2----t----3
options locale=English_UnitedStates; put event nldatmap.;	February 24, 2003 12:39:43 PM
options locale=Spanish_Mexico; put event nldatmap.;	24/02/2003 12:39:43 PM

関連項目:

出力形式:

- [“NLDATM 出力形式” \(126 ページ\)](#)
- [“NLDATMTM 出力形式” \(136 ページ\)](#)
- [“NLDATMW 出力形式” \(138 ページ\)](#)

NLDATMDT 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前、月の日および年を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMDT*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 18

範囲 10-200

例

この例では、en_US ロケールオプションを使用します。

ステートメント	結果
<pre>options locale=en_US; data _null_; x=86400; put x nldatmdt.; run; options locale=en_US; data _null_; x=86400; put x dtdate.; run;</pre>	<pre>January 02, 1960 02JAN60</pre>

関連項目:

出力形式:

- [“NLDATMMD 出力形式” \(130 ページ\)](#)

NLDATML 出力形式

SAS 日時値を、指定されたロケールの日時文字列に変換し、月、日、年、時間を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATML*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 30

範囲 9-200

詳細

NLDATML では、November 19, 2012 02:57:44 PM など、長い形式で月、日、年、時間を含む日付が出力されます。

例

この例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
<pre>data _null_; dt = datetime(); dy = date(); put "+--- NLDATML min=9 default=30 max=200 ---+"; put dt nldatml.; put dt nldatml19.; put dt nldatml24.; put dt nldatml30.; put dt nldatml200.; run;</pre>	<pre>+--- NLDATML min=9 default=30 max=200 ---+ November 19, 2012 02:57:44 PM 11/19/2012 14:57:44 Nov 19, 2012 02:57:44 PM November 19, 2012 02:57:44 PM November 19, 2012 02:57:44 PM</pre>

NLDATMM 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、月名と時間の短縮形を使用して月、日、年、時間を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMM*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 24

範囲 9-200

詳細

NLDATMM では、Nov 19, 2012 02:51:40 PM など、短縮形を使用した簡略形式で月、日、年、時間を含む日時が出力されます。

NLDATMMD 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前と月の日を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMMD w .

説明

w

出力フィールドの幅を指定します。

デフォルト 16

範囲 6-200

例

この例では、en_US ロケールオプションを使用します。

ステートメント	結果
<pre>options locale=en_US; data _null_ x=put(86400,nldatmmd.); put x=; run;</pre>	January 02

関連項目:

出力形式:

- [“NLDATMYM 出力形式” \(141 ページ\)](#)

NLDATMMDL 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、長い形式で月と日を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMMDL w .

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 12

範囲 5-200

詳細

NLDATMMDL では、November 19 など、長い形式で月と日を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATMMDL min=5 default=12 max=200 ---+
dt = datetime();	November 19
dy = date();	11/19
put "+--- NLDATMMDL min=5 default=12	Nov 19
max=200 ---+";	November 19
put dt nldatmdl;	November 19
put dt nldatmdl5;	
put dt nldatmdl9;	
put dt nldatmdl12;	
put dt nldatmdl200;	
run;	

NLDATMMDM 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、短縮形を使用して月と日を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMMDM*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 9

範囲 5-200

詳細

NLDATMMDM では、Nov 19 など、短縮形を使用した簡略形式で月と日を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null;	+--- NLDATMMDM min=5 default=9 max=200 ---+
dt = datetime();	Nov 19
dy = date();	11/19
put "+--- NLDATMMDM min=5 default=9	Nov 19
max=200 ---+";	Nov 19
put dt nldatmmdm;	
put dt nldatmmdm5;	
put dt nldatmmdm9;	
put dt nldatmmdm200.;	
run;	

NLDATMMDS 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して月と日を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMMDS*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 5

範囲 5-200

詳細

NLDATMMDS では、11/19 など、数字と区切り文字を使用した短い形式で月と日を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATMMDS min=5 default=5 max=200 ---+
dt = datetime();	11/19
dy = date();	11/19
put "+--- NLDATMMDS min=5 default=5	11/19
max=200 ---+";	11/19
put dt nldatmmds.;	
put dt nldatmmds5.;	
put dt nldatmmds5.;	
put dt nldatmmds200.;	
run;	

NLDATMMN 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から月の名前を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMMN*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 9

範囲 4-200

例

この例では、en_US ロケールオプションを使用します。

ステートメント	結果
data _null_;	
dt = datetime();	+--- NLDATMMN min=4 default=10
dy = date();	max=200 ---+
put "+--- NLDATEMN min=4 default=10	October
max=200 ---+";	Oct
put dt nldatmmn.;	October
put dt nldatmmn4.;	October
put dt nldatmmn10.;	
put dt nldatmmn200.;	
run;	

NLDATMS 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、MM/dd/yyyy の形式で書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMS*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 19

範囲 9-200

詳細

NLDATMS では、MM/dd/yyyy hh:mm:ss など、数字と区切り文字を使用した短い形式で月、日、年、時間を含む日付が出力されます。

例

この例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATMS min=9 default=19 max=200 ---+
dt = datetime();	11/19/2012 14:38:52
dy = date();	111912 14
put "+--- NLDATMS min=9 default=19	11/19/2012 14:38:52
max=200 ---+";	11/19/2012 14:38:52
put dt nldatms.;	
put dt nldatms10.;	
put dt nldatms19.;	
put dt nldatms200.;	
run;	

NLDATMTM 出力形式

SAS 日時値の時間部分を、指定されたロケールの時間値に変換し、その時間値から時刻を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMTM*w*.

説明

w 出力フィールドの幅を指定します。

デフォルト 16

範囲 16-200

比較

NLDATMTM*w*.出力形式は TOD*w*.出力形式と似ていますが、NLDATMTM*w*.出力形式がロケール固有である点が違います。

例

これらの例では、1361709583 を入力値として使用します。この値は、2003年2月24日の午後12:39:43に相当する SAS 日時値です。

ステートメント	結果
	----t----1

ステートメント	結果
options locale=English_UnitedStates; put event nldatmtm.;	12:39:43
options locale=German_Germany; put event nldatmtm.;	12.39 Uhr

関連項目:

出力形式:

- [“NLDATM 出力形式” \(126 ページ\)](#)
- [“NLDATMAP 出力形式” \(127 ページ\)](#)
- [“NLDATMW 出力形式” \(138 ページ\)](#)

NLDATMTZ 出力形式

ロケールの SAS 日時値の時間部分を、時刻とタイムゾーンに変換します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMTZ*w*.

説明

w
出力フィールドの幅を指定します。

デフォルト 32

範囲 16-200

例

この例は現在の日時値を使用します。

ステートメント	結果
options locale=fr_FR; data test; x=datetime(); put x=nldatmtz.;; run;	x=10 h 40 -0400

NLDATMW 出力形式

SAS 日時値を、曜日および日時形式の、ロケール依存の日時文字列に変換します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMW*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように曜日と日時が短縮されます。

デフォルト 41

範囲 16-200

詳細

エンコーディングとロケールの組み合わせによっては日時名が長くなるため、デフォルトの幅を使用した NLDATMW 出力形式では正確でない地域化された出力が生成される場合があります。ロケールとエンコーディングの組み合わせに対する推奨される幅については、[日付と時間のデフォルト幅の例外 \(60 ページ\)](#) を参照してください。NLDATMW 出力形式の推奨される幅を使用するようにします。

比較

NLDATMW*w*.出力形式は TWMDY*w*.出力形式と似ていますが、NLDATMW*w*.出力形式がロケール固有である点が違います。

例

これらの例では、1361709583 を入力値として使用します。この値は、2003 年 2 月 24 日の午後 12:39:43 に相当する SAS 日時値です。

ステートメント	結果
	----+----1----+----2----+----3

ステートメント	結果
options locale=English_UnitedStates;	Mon, Feb 24, 2003 12:39:43 PM
data _null_;	Mon, Feb 24, 2003 12:39:43 PM
x=put(1361709583,nldatmw.);	Monday, February 24, 2003 12:39:43 PM
y=put(1361709583,nldatmw30.);	
z=put(1361709583,nldatmw200.);	
run;	
options locale=german_germany;	Mo, 24. Feb 2003 12.39 Uhr
data _null_;	Mo, 24. Feb 2003 12.39 Uhr
x=put(1361709583,nldatmw.);	Montag, 24. Februar 2003 12.39 Uhr
y=put(1361709583,nldatmw30.);	
z=put(1361709583,nldatmw200.);	
run;	

関連項目:

出力形式:

- [“NLDATM 出力形式” \(126 ページ\)](#)
- [“NLDATMAP 出力形式” \(127 ページ\)](#)
- [“NLDATMTM 出力形式” \(136 ページ\)](#)

NLDATMWN 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から曜日を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMWN*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 9

範囲 4-200

例

SAS 日時値から曜日を書き出します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATMWN min=4 default=9 max=200 ---+"; put dt nldatmwn; put dt nldatmwn4; put dt nldatmwn9; put dt nldatmwn200; run;	+--- NLDATMWN min=4 default=9 max=200 ---+ Monday Mon Monday Monday

NLDATMWZ 出力形式

指定されたロケールの SAS 日時値を、曜日、日時、およびタイムゾーン値に変換します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMWZ*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように曜日と日時が短縮されます。

デフォルト 40

範囲 16-200

詳細

エンコーディングとロケールの組み合わせによっては日時名が長くなるため、デフォルトの幅を使用した NLDATMWZ 出力形式では正確でない地域化された出力が生成される場合があります。ロケールとエンコーディングの組み合わせに対する推奨される幅については、[日付と時間のデフォルト幅の例外 \(60 ページ\)](#)を参照してください。推奨される幅を使用するようにします。

例

この例は現在の日時値を使用します。

ステートメント	結果
<pre>options locale=fr_FR; data test; x=datetime(); put x=nldatmwz.; run;</pre>	<pre>x=vendredi 18 mars 2011 10 h 40 -0400</pre>

NLDATMYM 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と月の名前を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMYM*w*.

説明

w
出力フィールドの幅を指定します。

デフォルト 16

範囲 6-200

例

この例では、en_US ロケールオプションを使用します。

ステートメント	結果
<pre>options locale=en_US; data _null_; x=put(86400,nldatmym.); y=put(86400,nldatmym12.); put x=; put y=; run;</pre>	<pre>January 1960 January 1960</pre>

関連項目:

出力形式:

- [“NLDATMMD 出力形式” \(130 ページ\)](#)

NLDATMYML 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMYML*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 14

範囲 5-200

詳細

NLDATMYML では、November 2012 など、長い形式で月と年を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATMYML min=5 default=14 max=200 ---+"; put dt nldatmyml; put dt nldatmyml7; put dt nldatmyml11; put dt nldatmyml14; put dt nldatmyml200.;	+--- NLDATMYML min=5 default=14 max=200 ---+ November 2012 11/2012 Nov 2012 November 2012 November 2012
run;	

NLDATMYMM 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、月と年を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMYMM*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 11

範囲 5-200

詳細

NLDATMYMM では、Nov 2012 など、簡略形式で月と年を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATMYMM min=5 default=11 max=200 ---+"; put dt nldatmymm.; put dt nldatmymm7.; put dt nldatmymm11.; put dt nldatmymm200.; run;	+--- NLDATMYMM min=5 default=11 max=200 ---+ Nov 2012 11/2012 Nov 2012 Nov 2012

NLDATMYMS 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、数字と区切り文字を使用して月と年を書き出します。

カテゴリ: 日付と時間
CAS
配置: 左

構文

NLDATMYMS*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 7

範囲 5-200

詳細

NLDATMYMS では、11/2012 など、数字と区切り文字を使用した短い形式で月と年を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+++ NLDATMYMS min=5 default=7 max=200 ---+"; put dt nldatmyms; put dt nldatmyms5; put dt nldatmyms7; put dt nldatmyms200; run;	+++ NLDATMYMS min=5 default=7 max=200 ---+ 11/2012 11/12 11/2012 11/2012 11/2012 11/2012 11/2012 11/2012

NLDATMYQ 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と四半期を書き出します。

カテゴリ: 日付と時間
CAS
配置: 左

構文

NLDATMYQ w .

説明

w

出力フィールドの幅を指定します。

デフォルト 16

範囲 4-200

詳細

エンコーディングとロケールの組み合わせによっては日時名が長くなるため、デフォルトの幅を使用した NLDATMYQ 出力形式では正確でない地域化された出力が生成される場合があります。ロケールとエンコーディングの組み合わせに対する推奨される幅については、[日付と時間のデフォルト幅の例外 \(60 ページ\)](#)を参照してください。推奨される幅を使用するようにします。

例

この例では、fr_FR ロケールオプションを使用します。

ステートメント	結果
options locale=fr_FR;	
data _null_;	+--- NLDATMYQ min=4 default=16
dy=today();	max=200 ---+
dt=datetime();	16 T3 08
put "+--- NLDATMYQ min=4 default=16	4 *****
max=200 ---+";	14 T3 08
put ' 16' +5 dt nldatmyq.;	32 3e trimestre 2008
put ' 4' +5 dt nldatmyq4.;	200 3e trimestre 2008
put ' 14' +5 dt nldatmyq14.;	
put ' 32' +5 dt nldatmyq32.;	
put ' 200' +5 dt nldatmyq200.;	
run;	

NLDATMYQL 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、四半期(1-4)値、年の順に書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMYQL*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 18

範囲 4-200

詳細

NLDATMYQL では、4th quarter 2012 など、長い形式で四半期、次に年を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null; dt = datetime(); dy = date(); put "+--- NLDATMYQL min=4 default=18 max=200 ---+"; put dt nldatmyql; put dt nldatmyql6; put dt nldatmyql7; put dt nldatmyql18.; put dt nldatmyql200.; run;	+--- NLDATMYQL min=4 default=18 max=200 ---+ 4th quarter 2012 2012/4 Q4 2012 4th quarter 2012 4th quarter 2012

NLDATMYQM 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、四半期(1-4)、年の順に書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMYQM*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 7

範囲 4-200

詳細

NLDATMYQM では、Q4 2012 など、簡略形式で四半期、次に年を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATMYQM min=4 default=7 max=200 ---+
dt = datetime();	Q4 2012
dy = date();	2012/4
put "+--- NLDATMYQM min=4 default=7	Q4 2012
max=200 ---+";	Q4 2012
put dt nldatmyqm;	
put dt nldatmyqm6.;	
put dt nldatmyqm7.;	
put dt nldatmyqm200.;	
run;	

NLDATMYQS 出力形式

SAS 日時値を、指定されたロケールの日付文字列に変換し、年と四半期(1-4)を数字と区切り文字を使用して書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMYQS*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト	6
範囲	4-200

詳細

NLDATMYQS では、2012/4 など、数字と区切り文字を使用した短い形式で年と四半期を含む日付が出力されます。

例

次の例では、日付として 2012 年 11 月 19 日を使用します。

ステートメント	結果
data _null_;	+--- NLDATMYQS min=4 default=6 max=200 ---+
dt = datetime();	2012/4
dy = date();	12/4
put "+--- NLDATMYQS min=4 default=6	2012/4
max=200 ---+";	2012/4
put dt nldatmyqs.;	
put dt nldatmyqs4.;	
put dt nldatmyqs6.;	
put dt nldatmyqs200.;	
run;	

NLDATMYR 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から年を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMYR*w*.

説明

w
出力フィールドの幅を指定します。

デフォルト	16
範囲	2-200

例

この例では、en_US ロケールオプションを使用します。

ステートメント	結果
options locale=fr_FR;	
data _null_;	+--- NLDATMYR min=2 default=16
dy=today();	max=200 ---+
dt=datetime();	2008
put "+--- NLDATMYR min=2 default=16	08
max=200 ---+";	2008
put dt nldatmyr.;	2008
put dt nldatmyr2.;	
put dt nldatmyr32.;	
put dt nldatmyr200.;	
run;	

NLDATMYW 出力形式

SAS 日時値を、指定されたロケールの日時値に変換し、その値から年と週の名前を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLDATMYW*w*.

説明

w
出力フィールドの幅を指定します。

デフォルト 16

範囲 5-200

例

この例では、fr_FR ロケールオプションを使用します。

ステートメント	結果
options locale=fr_FR;	
data _null_;	+--- NLDATMYW min=5 default=16
dy=today();	max=200 ---+
dt=datetime();	16 Week 33 2008
put "+--- NLDATMYW min=5 default=16	5 *****
max=200 ---+";	8 W33 08
put ' 16' +5 dt nldatmyw.;	32 Week 33 2008
put ' 5' +5 dt nldatmyw5.;	200
put ' 8' +5 dt nldatmyw8.;	Week 33 2008
put ' 32' +5 dt nldatmyw32.;	
put ' 200' +5 dt nldatmyw200.;	
run;	

NLDATMZ 出力形式

SAS 日時値を、タイムゾーンおよび日時形式の、ロケール依存の日時文字列に変換します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

NLDATMZ*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 40

範囲 16-200

例

この例は現在の日時値を使用します。

ステートメント	結果
<pre>options locale=fr_FR; data test; x=datetime(); put x=nldatmz.; run;</pre>	<pre>x=18 mars 2011 10 h 40 -0400</pre>

NLMNIAED 出力形式

アラブ首長国連邦の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIAED*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmmiaed32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(AED1,234.57)
put y=;	\$-1,234.57

関連項目:**出力形式:**

- [“NLMNLAED 出力形式” \(188 ページ\)](#)

NLMNIAUD 出力形式

オーストラリアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIAUD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlomniaud32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント

結果

----+----1----+

put x=;

(AUD1,234.57)

put y=;

\$-1,234.57

関連項目:**出力形式:**

- [“NLMNLAUD 出力形式” \(189 ページ\)](#)

NLMNIBGN 出力形式

ブルガリアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIBGN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmbnbn32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(BGN1,234.57)
put y=;	\$-1,234.57

関連項目:**出力形式:**

- [“NLMNLBGN 出力形式” \(190 ページ\)](#)

NLMNIBRL 出力形式

ブラジルの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIBRL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmbri32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(BRL1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLBRL 出力形式” \(191 ページ\)](#)

NLMNICAD 出力形式

カナダの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNICAD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmcad32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(CAD1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLCAD 出力形式” \(192 ページ\)](#)

NLMNICHF 出力形式

リヒテンシュタインとスイスの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNICHF*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmmichf32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(CHF1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLCHF 出力形式” \(193 ページ\)](#)

NLMNICNY 出力形式

中国の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNICNY $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 02

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmcny32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(CNY1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLCNY 出力形式” \(194 ページ\)](#)

NLMNICZK 出力形式

チェコ共和国の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNICZK $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmiczk32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(CZK1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLCZK 出力形式” \(195 ページ\)](#)

NLMNIDKK 出力形式

デンマーク、フェロー諸島およびグリーンランドの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIDKK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmidkk32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(DKK1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLDKK 出力形式” \(196 ページ\)](#)

NLMNIEEK 出力形式

エストニアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIEEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlennieek32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(EEK1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLEEK 出力形式” \(197 ページ\)](#)

NLMNIEGP 出力形式

エジプトの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIEGP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmiiegp32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(EGP1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLEGP 出力形式” \(198 ページ\)](#)

NLMNIEUR 出力形式

オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIEUR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが **Locale=German_Germany** に設定されています。

```
x=put(-1234.56789,nlmmieur32.2);
```

```
y=put(-1234.56789,nlmmieur32.2);
```

ステートメント	結果
	----+----1----+
put x=;	x=-1.234,57 EUR
put y=;	y=-1.234,57 €

関連項目:

出力形式:

- [“NLMNLEUR 出力形式” \(199 ページ\)](#)

NLMNIGBP 出力形式

英国の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIGBP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmgbp32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(GBP1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLGBP 出力形式” \(200 ページ\)](#)

NLMNIHKD 出力形式

香港の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIHKD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmiHKD32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(HKD1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLHKD 出力形式” \(201 ページ\)](#)

NLMNIHRK 出力形式

クロアチアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIHRK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmihrk32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(HRK1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLHRK 出力形式” \(202 ページ\)](#)

NLMNIHUF 出力形式

ハンガリーの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIHUF*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmiuhuf32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(HUF1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLHUF 出力形式” \(203 ページ\)](#)

NLMNIIDR 出力形式

インドネシアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIIDR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmiidr32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(IDR1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLIDR 出力形式” \(204 ページ\)](#)

NLMNIILS 出力形式

イスラエルの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIILS*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmoiils32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(ILS1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLILS 出力形式” \(205 ページ\)](#)

NLMNIINR 出力形式

インドの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIINR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmiinr32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(INR1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLINR 出力形式” \(206 ページ\)](#)

NLMNIJPY 出力形式

日本の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIJPY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnijpy32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1-----+
put x=;	(JPY1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIJPY 出力形式” \(207 ページ\)](#)

NLMNIKRW 出力形式

韓国の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIKRW*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlkrw32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(KRW1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLKRW 出力形式” \(208 ページ\)](#)

NLMNLTTL 出力形式

リトアニアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLTTL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlt32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(LTL1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLLTL 出力形式” \(209 ページ\)](#)

NLMNILVL 出力形式

ラトビアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNILVL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlvl(32.2));
```

```
y=put(-1234.56789,dollar(32.2));
```

ステートメント	結果
	----+----1----+
put x=;	(LVL1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLLVL 出力形式” \(210 ページ\)](#)

NLMNIMOP 出力形式

マカオの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIMOP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmmiop32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(MOP1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLMOP 出力形式” \(211 ページ\)](#)

NLMNIMXN 出力形式

メキシコの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIMXN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmmimxn32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(MXN1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLMXN 出力形式” \(212 ページ\)](#)

NLMNIMYR 出力形式

マレーシアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIMYR $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmmimyr32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(MYR1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLMYR 出力形式” \(213 ページ\)](#)

NLMNINOK 出力形式

ノルウェーの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNINOK $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlminok32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(NOK1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNINOK 出力形式” \(214 ページ\)](#)

NLMNINZD 出力形式

数値

カテゴリ: 数値

配置: 左

構文

NLMNINZD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlminz32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(NZD1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNINZD 出力形式” \(215 ページ\)](#)

NLMNIPLN 出力形式

ポーランドの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIPLN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmoni32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(PLN1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLPLN 出力形式” \(216 ページ\)](#)

NLMNIRUB 出力形式

ロシアの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIRUB*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmdirub32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(RUB1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLRUB 出力形式” \(217 ページ\)](#)

NLMNISEK 出力形式

スウェーデンの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNISEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmmisek32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(SEK1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLSEK 出力形式” \(218 ページ\)](#)

NLMNISGD 出力形式

シンガポールの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNISGD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmsgd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(SGD1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLSGD 出力形式” \(219 ページ\)](#)

NLMNITHB 出力形式

タイの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNITHB*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnithb32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(THB1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLTHB 出力形式” \(220 ページ\)](#)

NLMNITRY 出力形式

トルコの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNITRY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnitry32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	---+---1---+
put x=;	(TRY1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLTRY 出力形式” \(221 ページ\)](#)

NLMNITWD 出力形式

台湾の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNITWD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmitwd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(TWD1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLTWD 出力形式” \(222 ページ\)](#)

NLMNIUSD 出力形式

プエルトリコと米国の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIUSD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmiusd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(USD1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLUSD 出力形式” \(223 ページ\)](#)

NLMNIZAR 出力形式

南アフリカの国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNIZAR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnizar32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(ZAR1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLZAR 出力形式” \(224 ページ\)](#)

NLMNLAED 出力形式

アラブ首長国連邦のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLAED $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlaed32.2);
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(AED1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIAED 出力形式” \(151 ページ\)](#)

NLMNLAUD 出力形式

オーストラリアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLAUD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlaud32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(AU\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLAUD 出力形式” \(152 ページ\)](#)

NLMNLBGN 出力形式

ブルガリアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLBGN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlbgn32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(BGN1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIBGN 出力形式” \(153 ページ\)](#)

NLMNLBRL 出力形式

ブラジルのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLBRL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlbrl32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(R\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIBRL 出力形式” \(154 ページ\)](#)

NLMNLCAD 出力形式

カナダのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLCAD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmcad32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(CA\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNICAD 出力形式” \(155 ページ\)](#)

NLMNLCHF 出力形式

リヒテンシュタインとスイスのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLCHF*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlchf32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	SFr.1,234.57
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNICHF 出力形式” \(156 ページ\)](#)

NLMNLCNY 出力形式

中国のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLCNY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlncy32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(RMB1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNICNY 出力形式” \(157 ページ\)](#)

NLMNLCZK 出力形式

チェコ共和国のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLCZK $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlnczk32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(CZK1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNICZK 出力形式” \(158 ページ\)](#)

NLMNLDKK 出力形式

デンマーク、フェロー諸島およびグリーンランドのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLDKK $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmdlkk32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1-----+
put x=;	(kr1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIDKK 出力形式” \(159 ページ\)](#)

NLMNLEEK 出力形式

エストニアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLEEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnleek32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(Kr1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLEEK 出力形式” \(160 ページ\)](#)

NLMNLEGP 出力形式

エジプトのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLEGP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlegp32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(EGP1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIEGP 出力形式” \(161 ページ\)](#)

NLMNLEUR 出力形式

オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLEUR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが German_Germany に設定されています。

```
x=put(-1234.56789,nlmmieur32.2);
```

```
y=put(-1234.56789,nlmmieur32.2);
```

ステートメント	結果
	----+----1----+----2----+
put x=;	x=-1.234,57 EUR
put y=;	y=-1.234,57 €

関連項目:

出力形式:

- [“NLMNLEUR 出力形式” \(162 ページ\)](#)

NLMNLGBP 出力形式

英国のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLGBP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlnlgbp32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1-----+
put x=;	(£1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIGBP 出力形式” \(163 ページ\)](#)

NLMNLHKD 出力形式

香港のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLHKD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlhkd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(HK\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIHKD 出力形式” \(164 ページ\)](#)

NLMNLHRK 出力形式

クロアチアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLHRK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlhrk32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(Kn1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIHRK 出力形式” \(165 ページ\)](#)

NLMNLHUF 出力形式

ハンガリーのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLHUF*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlhuf32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(Ft1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIHUF 出力形式” \(166 ページ\)](#)

NLMNLIDR 出力形式

インドネシアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLIDR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlidr32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(Rp1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIIDR 出力形式” \(167 ページ\)](#)

NLMNLILS 出力形式

イスラエルのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLILS*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlils32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(ILS1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIIIS 出力形式” \(168 ページ\)](#)

NLMNLINR 出力形式

インドのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLINR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlinr32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(INR1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIINR 出力形式” \(169 ページ\)](#)

NLMNLJPY 出力形式

日本の国際表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLJPY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmljpy32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(JPY1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLJPY 出力形式” \(170 ページ\)](#)

NLMNLKRW 出力形式

韓国のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLKRW*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlkrw32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(KRW1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLKRW 出力形式” \(171 ページ\)](#)

NLMNLLTL 出力形式

リトアニアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLLTL $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlt32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(LT1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNILTL 出力形式” \(172 ページ\)](#)

NLMNLLVL 出力形式

ラトビアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLLVL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlvl(32.2));
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(Ls1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNILVL 出力形式” \(173 ページ\)](#)

NLMNLMOP 出力形式

マカオのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLMOP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlmop32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(P1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIMOP 出力形式” \(174 ページ\)](#)

NLMNLMXN 出力形式

メキシコのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLMXN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlmxn32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(MX\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIMXN 出力形式” \(175 ページ\)](#)

NLMNLMYR 出力形式

マレーシアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLMYR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlmyr32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(R1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIMYR 出力形式” \(176 ページ\)](#)

NLMNLNOK 出力形式

ノルウェーのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLNOK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlnok32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(kr1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNINOK 出力形式” \(177 ページ\)](#)

NLMNLNZD 出力形式

ニュージーランドのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLNZD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmnlznd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(NZ\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNINZD 出力形式” \(178 ページ\)](#)

NLMNLPLN 出力形式

ポーランドのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLPLN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlpln32.2);
y=put(-1234.56789,dollar32.2)
```

ステートメント	結果
	----+----1----+
put x=;	(PLN1,234.57
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIPLN 出力形式” \(179 ページ\)](#)

NLMNLRUB 出力形式

ロシアのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLRUB*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlrub32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(RUB1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLRUB 出力形式” \(180 ページ\)](#)

NLMNLSEK 出力形式

スウェーデンのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLSEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlnsek32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(kr1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNLSEK 出力形式” \(181 ページ\)](#)

NLMNLSGD 出力形式

シンガポールのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLSGD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmsgd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(SG\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNISGD 出力形式” \(182 ページ\)](#)

NLMNLTHB 出力形式

タイのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLTHB*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlnthb32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(THB1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNITHB 出力形式” \(183 ページ\)](#)

NLMNLTRY 出力形式

トルコのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLTRY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlntry32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(YTL1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNITRY 出力形式” \(184 ページ\)](#)

NLMNLTWD 出力形式

台湾のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLTWD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmltwd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(NT\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNITWD 出力形式” \(185 ページ\)](#)

NLMNLUSD 出力形式

プエルトリコと米国のローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLUSD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlusd32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(US\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIUSD 出力形式” \(186 ページ\)](#)

NLMNLZAR 出力形式

南アフリカのローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

配置: 左

構文

NLMNLZAR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

数値の小数点以下の桁数を指定します。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmlzar32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(R1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNIZAR 出力形式” \(187 ページ\)](#)

NLMNY 出力形式

指定されたロケールで、ローカル通貨を使用してローカル表示形式で通貨データを書き出します。

カテゴリ: 数値

CAS

配置: 左

構文

NLMNY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 9

範囲 1-32

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-31

詳細

NLMNY*w.d* 入力形式では、2 の補数で表される負の値を含む、バイナリ整数(固定小数点)値を読み込みます。NLMNY*w.d* 出力形式では、そのロケールでの通貨記号、桁区切り文字および小数点区切り文字を使用して数値を書き出します。

注: NLMNY*w.d* 出力形式では、通貨データの変換は行われません。したがって、フォーマットされた数値は、現在のロケールの通貨が使用された値と等しくなります。

比較

NLMNY*w.d* 出力形式と NLMNYI*w.d* 出力形式では、ロケールに依存する桁区切り文字と小数点区切り文字を使用して通貨データを書き出します。ただし、NLMNYI*w.d* 出力形式では USD などの 3 文字の国際通貨コードを使用するのに対し、NLMNY*w.d* 出力形式では \$ などのローカル通貨記号を使用します。

NLMNY*w.d* 出力形式は DOLLAR*w.d* 出力形式と似ていますが、NLMNY*w.d* 出力形式がロケール固有である点が違います。

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmny32.2);
```

```
y=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(\$1,234.57)
put y=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNYI 出力形式” \(226 ページ\)](#)

入力形式:

- [“NLMNY 入力形式” \(501 ページ\)](#)
- [“NLMNYI 入力形式” \(502 ページ\)](#)

NLMNYI 出力形式

指定されたロケールで、通貨データを国際表示形式で書き出します。

カテゴリ: 数値

CAS

配置: 左

構文

NLMNYI*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 9

範囲 1-32

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-31

詳細

NLMNYIw.d 入力形式では、2 の補数で表される負の値を含む、バイナリ整数(固定小数点)値を読み込みます。NLMNYIw.d 出力形式では、国際通貨コード、およびロケールに依存する桁区切り文字と小数点区切り文字を使用して数値を書き出します。国際通貨コードの位置もロケールに応じて決まります。

注: NLMNYIw.d 出力形式では、通貨データの変換は行われません。したがって、フォーマットされた数値は、現在のロケールの通貨が使用された値と等しくなります。

比較

NLMNYw.d 出力形式と NLMNYIw.d 出力形式では、ロケールに依存する桁区切り文字と小数点区切り文字を使用して通貨データを書き出します。ただし、NLMNYIw.d 出力形式では USD などの 3 文字の国際通貨コードを使用するのに対し、NLMNYw.d 出力形式では \$ などのローカル通貨記号を使用します。

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-1234.56789,nlmyi32.2);
y=put(-1234.56789,nlmy32.2);
z=put(-1234.56789,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	(USD1,234.57)
put y=;	(\$1,234.57)
put z=;	\$-1,234.57

関連項目:

出力形式:

- [“NLMNY 出力形式” \(225 ページ\)](#)

入力形式:

- [“NLMNY 入力形式” \(501 ページ\)](#)
- [“NLMNYI 入力形式” \(502 ページ\)](#)

NLNUM 出力形式

指定されたロケールで、数値をローカル表示形式で書き出します。

カテゴリ: 数値
CAS

配置: 左

構文

NLNUM*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 1-32

d10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-31

詳細

NLNUM*w.d* 入力形式では、2 の補数で表される負の値を含む、バイナリ整数(固定小数点)値を読み込みます。NLNUM*w.d* 入力形式では、2 の補数で表される負の値を含む、バイナリ整数(固定小数点)値を読み込みます。

比較

NLNUM*w.d* 出力形式では、ロケールに依存する桁区切り文字と小数点区切り文字を使用して数値を書き出します。NLNUM*w.d* 出力形式では、桁区切り文字としてカンマ(,)を、小数点区切り文字としてピリオド(.)を使用して数値を書き出します。

w 値または *d* 値が、フォーマットされた数値を生成するほど十分に大きくない場合、NLNUM*w.d* 出力形式では、小数精度が若干失われるとしても、可能な限り桁区切り文字を出力するアルゴリズムを使用します。

例

```
x=put(-1234356.7891,nlnum32.2);
```

ステートメント	結果
	----+-----1-----+
options LOCALE=English_UnitedStates; put x=;	-1,234,356.79
options LOCALE=German_Germany; put x=;	-1.234.356,79

関連項目:

出力形式:

- [“NLNUMI 出力形式” \(229 ページ\)](#)

入力形式:

- [“NLNUM 入力形式” \(503 ページ\)](#)
- [“NLNUMI 入力形式” \(505 ページ\)](#)

NLNUMI 出力形式

指定されたロケールで、数値を国際表示形式で書き出します。

カテゴリ: 数値
CAS

配置: 左

構文

NLNUMI $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 1-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 0

範囲 0-31

詳細

NLNUMI $w.d$ 入力形式では、2 の補数で表される負の値を含む、バイナリ整数(固定小数点)値を読み込みます。NLNUMI $w.d$ 出力形式では、すべてのロケールに対し、桁区切り文字としてカンマ(,)を、また小数点区切り文字としてピリオド(.)を使用して数値を書き出します。

比較

NLNUMI $w.d$ 出力形式では、指定されたロケールで、数値データを国際表示形式で書き出します。NLNUMI $w.d$ 出力形式では、桁区切り文字としてカンマ(,)を、小数点区切り文字としてピリオド(.)を使用して数値を書き出します。

w 値または d 値が、フォーマットされた数値を生成するほど十分に大きくない場合、NLNUM $w.d$ 出力形式では、小数精度が若干失われるとしても、可能な限り桁区切り文字を出力するアルゴリズムを使用します。

例

```
x=put(-1234356.7891,nlnumi32.2);
```

ステートメント	結果
	----+----1----+
options LOCALE=English_UnitedStates; put x=;	-1,234,356.79
options LOCALE=German_Germany; put x=;	-1,234,356.79

関連項目:

出力形式:

- [“NLNUM 出力形式” \(227 ページ\)](#)

入力形式:

- [“NLNUM 入力形式” \(503 ページ\)](#)
- [“NLNUMI 入力形式” \(505 ページ\)](#)

NLPCT 出力形式

指定されたロケールで、百分率のデータをローカル表示形式で書き出します。

カテゴリ: 数値

CAS

配置: 左

構文

NLPCT $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 4-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-31

比較

NLPCT*w.d* 出力形式では、指定されたロケールで、百分率のデータをローカル表示形式で書き出します。NLPCT*w.d* 出力形式では、ロケールに依存する桁区切り文字と小数点区切り文字を使用して百分率値を書き出します。NLPCTI*w.d* 出力形式では、桁区切り文字としてカンマ(,)を、小数点区切り文字としてピリオド(.)を使用して百分率値を書き出します。

NLPCT*w.d* 出力形式は PERCENT*w.d* 出力形式と似ていますが、NLPCT*w.d* 出力形式がロケール固有である点が違います。

例

```
x=put(-12.3456789,nlpct32.2);
y=put(-12.3456789,nlpcti32.2);
z=put(-12.3456789,percent32.2);
```

ステートメント	結果
	----+-----1
options LOCALE=English_UnitedStates;	-1,234.57%
put x=;	-1,234.57%
put y=;	(1234.57%)
put z=;	
options LOCALE=German_Germany;	-1.234,57%
put x=;	-1,234.57%
put y=;	(1234.57%)
put z=;	

関連項目:**出力形式:**

- [“NLPCTI 出力形式” \(232 ページ\)](#)

入力形式:

- [“NLPCT 入力形式” \(506 ページ\)](#)
- [“NLPCTI 入力形式” \(507 ページ\)](#)

NLPCTI 出力形式

指定されたロケールで、百分率のデータを国際表示形式で書き出します。

カテゴリ: 数値

CAS

配置: 左

構文

NLPCTI*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 4-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-31

比較

NLPCTI*w.d* 出力形式では、指定されたロケールで、百分率のデータを国際表示形式で書き出します。NLPCT*w.d* 出力形式では、ロケールに依存する桁区切り文字と小数点区切り文字を使用して百分率値を書き出します。NLPCTI*w.d* 出力形式では、桁区切り文字としてカンマ(,)を、小数点区切り文字としてピリオド(.)を使用して百分率値を書き出します。

NLPCT*w.d* 出力形式は PERCENT*w.d* 出力形式と似ていますが、NLPCT*w.d* 出力形式がロケール固有である点が違います。

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put(-12.3456789,nlpcti32.2);
y=put(-12.3456789,percent32.2);
```

ステートメント

結果

----+-----1

ステートメント	結果
put x=;	-1,234.57%
put y=;	(1234.57)

関連項目:

出力形式:

- [“NLPCT 出力形式” \(230 ページ\)](#)

入力形式:

- [“NLPCT 入力形式” \(506 ページ\)](#)
- [“NLPCTI 入力形式” \(507 ページ\)](#)

NLPCTN 出力形式

百分率を書き出します。負の値にはマイナス記号を使用します。

カテゴリ: 数値

CAS

配置: 右

構文

NLPCTN*w,d*

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 4-32

ヒント

出力フィールドの幅数には、数値が負か正かにかかわらず、マイナス記号(-)、パーセント記号(%)、末尾の空白分を含める必要があります。

d

数値の小数点以下の桁数を指定します。この引数はオプションです。

デフォルト 0

範囲 0-31

要件 *w* より小さい数にします。

詳細

NLPCTN*w.d* 出力形式は、100 を乗算した負の値の先頭にマイナス記号を追加し、フォーマットした値の最後にパーセント記号(%)を追加します。

例

x=-0.02;

ステートメント	結果
put x nlpctn6.;	x=-2%
put x percentn6.;	x=-2%

NLPCTP 出力形式

ロケール固有の数値を百分率として書き出します。

カテゴリ: 数値

CAS

配置: 右

構文

NLPCTP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 4-32

ヒント 出力フィールドの幅数には、パーセント記号(%)分を含める必要があります。

d

数値の小数点以下の桁数を指定します。この引数はオプションです。NLPCTP 出力形式の桁区切り文字と小数点の記号は、ロケールに固有です。

デフォルト 0

範囲 0-31

要件 *w* より小さい数にします。

詳細

NLPCTP*w.d* 出力形式は、100 を乗算してフォーマットした値の最後にパーセント記号(%)を追加します。NLPCTP*w.d* 出力形式は PERCENT*w.d* 出力形式に似ていますが、NLPCTP*w.d* 出力形式の桁区切り文字と小数点の記号がロケール固有である点が違います。

例

x=-0.02;

ステートメント	結果
put x nlpctp6.;	-2%
put x percent6.;	(2%)

NLPVALUE 出力形式

指定されたロケールで、p 値をローカル表示形式で書き出します。

カテゴリ: 数値

CAS

配置: 左

構文

NLPVALUE*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 6

範囲 3-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、d 値は無視されます。

デフォルト 4

範囲 1-30

例

この例では、german_Germany ロケールオプションを使用します。

ステートメント:

```
options locale=german_germany;
data _null_;
  put "+--- nlpvalue min=3 default=6 max=32 ---+";
  x=0.1248;
  put x= +5 x pvalue. +5 x nlpvalue.;
  put x= +5 x pvalue3.1 +5 x nlpvalue3.1;
  put x= +5 x pvalue20.2 +5 x nlpvalue20.2;
  put x= +5 x pvalue32.3 +5 x nlpvalue32.3;
run;
```

結果:

```
+--- nlpvalue min=3 default=6 max=32 ---+
x=0.1248 0.1248 0,1248
x=0.1248 0.1 0,1
x=0.1248 0.12 0,12
x=0.1248 0.125 0,125
```

関連項目:

出力形式:

- [“PVALUE Format” \(SAS Viya Formats and Informats: Reference\)](#)

NLSTRMON 出力形式

指定されたロケールの月名を書き出します。

カテゴリ: 数値
CAS

配置: 左

構文

NLSTRMON*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 20

範囲 1-200

d

次の項目を指定します。

- 00000001:短縮形式で書き出します。
- 00000010:大文字形式で書き出します。

デフォルト 0
 範囲 0-3

詳細

NLSTRMONw.d 出力形式では、指定されたロケールの月名として 1 から 12 の SAS 値を書き出します。次の例では、English_UnitedStates ロケールを使用します。

- 1 =第 1 月(January)
- 2 =第 2 月(February)
- 3 =第 3 月(March)
- 4 =第 4 月(April)
- 5 =第 5 月(May)
- 6 =第 6 月(June)
- 7 =第 7 月(July)
- 8 =第 8 月(August)
- 9 =第 9 月(September)
- 10 =第 10 月(October)
- 11 =第 11 月(November)
- 12 =第 12 月(December)

例

この例では、English_UnitedSates セッションエンコーディングを使用します。

ステートメント	結果
Data _null_;	
monnum = 1 ; /* January=1,	January
December=12 */	Jan
put monnum NLSTRMON20.;	JANUARY
put monnum NLSTRMON20.1; /*	JAN
decimal .1 specified use abbreviation.*/	
put monnum NLSTRMON20.2;	
put monnum NLSTRMON20.3;	
run;	

NLSTRQTR 出力形式

指定されたロケールの四半期の形式で数値を書き出します。

カテゴリ: 数値
 CAS

配置: 左

構文

NLSTRQTR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 20

範囲 1-200

d

次の項目を指定します。

- 00000001:短縮形式で書き出します。
- 00000010:大文字形式で書き出します。

デフォルト 0

範囲 0-3

詳細

NLSTRQTR*w.d* 出力形式では、指定されたロケールの四半期名として 1 から 4 の SAS 値を書き出します。次の例では、English_UnitedStates ロケールを使用します。

- 1 = 1st quarter
- 2 = 2nd quarter
- 3 = 3rd quarter
- 4 = 4th quarter

例

この例では、English_UnitedSates セッションエンコーディングを使用します。

ステートメント	結果
Data _null_;	
qtrnum = 1 ; /* January=1, December=12 */	1st quarter Q1
put qtrnum NLSTRQTR20.;	1ST QUARTER
put qtrnum NLSTRQTR20.1; /* decimal . 1 specified use abbreviation.*/	Q1
put qtrnum NLSTRQTR20.2;	
put qtrnum NLSTRQTR20.3; run;	

NLSTRWK 出力形式

指定されたロケールの曜日の形式で数値を書き出します。

カテゴリ: 数値

CAS

配置: 左

構文

NLSTRWK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 20

範囲 1-200

d

次の項目を指定します。

- 00000001:短縮形式で書き出します。
- 00000010:大文字形式で書き出します。

デフォルト 0

範囲 0-3

詳細

NLSTRWK*w.d* 出力形式では、指定されたロケールの曜日名として 1 から 7 までの SAS 値を書き出します。次の例では、English_UnitedStates ロケールを使用します。

- 1=週の 1 日目(Monday)
- 2=週の 2 日目(Tuesday)
- 3=週の 3 日目(Wednesday)
- 4=週の 4 日目(Thursday)
- 5=週の 5 日目(Friday)
- 6=週の 6 日目(Saturday)
- 7=週の 7 日目(Sunday)

例

この例では、English_UnitedSates セッションエンコーディングを使用します。

ステートメント	結果
Data _null_;	
wknum = 1 ; /* Sunday=1, Saturday=7 */	Sunday Sun
put wknum NLSTRWK20.;	SUNDAY
put wknum NLSTRWK20.1; /* decimal .1 specified use abbreviation.*/	SUN
put wknum NLSTRWK20.2;	
put wknum NLSTRWK20.3;	
run;	

NLTIMAP 出力形式

SAS 時間値を、指定されたロケールの時間値に変換し、その時間値から午前または午後付きの時間値を書き出します。NLTIMAP は、SAS 日時値も変換します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLTIMAP*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 10

範囲 4-200

詳細

デフォルトの幅を使用した NLTIMAP 出力形式では、エンコーディングとロケールの組み合わせによっては日時名が長くなるため、正確でない地域化された出力が生成される場合があります。ロケールとエンコーディングの組み合わせに対する推奨される幅については、[日付と時間のデフォルト幅の例外 \(60 ページ\)](#) を参照してください。推奨される幅を使用するようにします。

比較

NLTIMAP*w*.出力形式は TIMEAMP*Mw*.出力形式と似ていますが、NLTIMAP*w*.出力形式がロケール固有である点が違います。

例

これらの例では、59083 を入力値として使用します。この値は、午後 04:24:43 に相当する SAS 日時値です。

ステートメント	結果
	----t----1----+
options locale=English_UnitedStates; put time nltimap,;	4:24:43 PM
options locale=German_Germany; put time nltimap14,;	4:24:43 nachm

関連項目:

出力形式:

- [“NLTIME 出力形式” \(241 ページ\)](#)

NLTIME 出力形式

SAS 時間値を、指定されたロケールの時間値に変換し、その時間値から時刻を書き出します。NLTIME は、SAS 日時値も変換します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

NLTIME*w*.

説明

w 入力幅を指定します。

デフォルト 20

範囲 10-200

比較

NLTIME*w*.出力形式は TIME*w*.出力形式と似ていますが、NLTIME*w*.出力形式がロケール固有である点が違います。

例

これらの例では、59083 を入力値として使用します。この値は、午後 04:24:43 に相当する SAS 日時値です。

ステートメント	結果
	----t----1----+
options locale=English_UnitedStates; put time ntime.;	4:24:43
options locale=German_Germany; put time ntime.;	16.24

関連項目:

出力形式:

- [“NLTIMAP 出力形式” \(240 ページ\)](#)

\$UCS2B 出力形式

現在の SAS セッションのエンコーディングを使用する文字列を処理し、ビッグエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS2B*w*.

説明

w

出力フィールドの幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

詳細

\$UCS2B*w*.出力形式では、ビッグエンディアン、16 ビット、UCS2 (universal character set code in two octets)形式の UNICODE (ユニコード)文字列を書き出します。現在の SAS セッションのエンコーディングを使用する文字列を処理します。

比較

\$UCS2Bw.出力形式では、\$UCS2BEw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
data _null_;	5927
x = '大';	
y=put(x,\$ucs2b2.);	
put y \$hex.;	
run;	

関連項目:

出力形式:

- ["\\$UCS2L 出力形式" \(244 ページ\)](#)
- ["\\$UCS2X 出力形式" \(247 ページ\)](#)
- ["\\$UTF8X 出力形式" \(263 ページ\)](#)
- ["\\$UCS2BE 出力形式" \(243 ページ\)](#)

入力形式:

- ["\\$UCS2B 入力形式" \(510 ページ\)](#)
- ["\\$UCS2BE 入力形式" \(511 ページ\)](#)
- ["\\$UCS2L 入力形式" \(512 ページ\)](#)
- ["\\$UCS2X 入力形式" \(514 ページ\)](#)
- ["\\$UTF8X 入力形式" \(529 ページ\)](#)

\$UCS2BE 出力形式

ビッグエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS2BEw.

説明

w

出力フィールドの幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

詳細

\$UCS2BEw.出力形式では、現在の SAS セッションのエンコーディングで文字列を書き出します。ビッグエンディアン、16 ビット、UCS2 (universal character set code in two octets)形式の UNICODE (ユニコード)文字列を処理します。

比較

\$UCS2BEw.出力形式では、\$UCS2Bw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
x='592700410042'x; put x \$ucs2be.;	大 AB

関連項目:

出力形式:

- ["\\$UCS2B 出力形式" \(242 ページ\)](#)

入力形式:

- ["\\$UCS2B 入力形式" \(510 ページ\)](#)
- ["\\$UCS2BE 入力形式" \(511 ページ\)](#)

\$UCS2L 出力形式

現在の SAS セッションのエンコーディングを使用する文字列を処理し、リトルエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS2Lw.

説明

w

出力フィールドの幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

詳細

\$UCS2Lw.出力形式では、リトルエンディアン、16 ビット、UCS2 (universal character set code in two octets)形式の UNICODE (ユニコード)文字列を書き出します。現在の SAS セッションのエンコーディングを使用する文字列を処理します。

比較

\$UCS2Lw.出力形式では、\$UCS2LEw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
data _null; x = '大'; y=put(x,\$ucs2l2.); put y \$hex; run;	2759

関連項目:

出力形式:

- ["\\$UCS2B 出力形式" \(242 ページ\)](#)
- ["\\$UCS2LE 出力形式" \(246 ページ\)](#)
- ["\\$UCS2X 出力形式" \(247 ページ\)](#)
- ["\\$UTF8X 出力形式" \(263 ページ\)](#)

入力形式:

- ["\\$UCS2B 入力形式" \(510 ページ\)](#)

- “\$UCS2L 入力形式” (512 ページ)
- “\$UCS2LE 入力形式” (513 ページ)
- “\$UCS2X 入力形式” (514 ページ)
- “\$UTF8X 入力形式” (529 ページ)

\$UCS2LE 出力形式

リトルエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS2LE*w*.

説明

w

出力フィールドの幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

詳細

\$UCS2LE*w*.出力形式では、現在の SAS セッションのエンコーディングで文字列を書き出します。リトルエンディアン、16 ビット、UCS2 (universal character set code in two octets)形式の UNICODE (ユニコード)文字列を処理します。

比較

\$UCS2LE*w*.出力形式とは、\$UCS2L*w*.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
<pre>x='275941004200'x; put x \$ucs2le.;</pre>	大AB

関連項目:**出力形式:**

- “\$UCS2L 出力形式” (244 ページ)

入力形式:

- “\$UCS2L 入力形式” (512 ページ)
- “\$UCS2LE 入力形式” (513 ページ)

\$UCS2X 出力形式

現在の SAS セッションのエンコーディングを使用する文字列を処理し、ネイティブエンディアン、16 ビット、UCS2 形式の UNICODE (ユニコード)文字列を書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS2Xw.

説明

w

出力フィールドの幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

詳細

\$UCS2Xw.出力形式では、動作環境のバイトオーダーで、16 ビットの UCS2 (universal character set code in two octets)形式の UNICODE (ユニコード)文字列を書き出します。

比較

\$UCS2Xw.出力形式では、\$UCS2Xew.出力形式とは逆の処理を行います。同一の動作環境内のデータを変換する場合は、\$UCS2Xw.出力形式を使用します。異なる動作環境でデータを変換する場合は、\$UCS2Bw.出力形式または\$UCS2Lw.出力形式を使用します。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
<pre>x = '大'; put x \$ucs2x2.;</pre>	'5927'x (binary) or '2759'x (little endian)

関連項目:

出力形式:

- “\$UCS2B 出力形式” (242 ページ)
- “\$UCS2XE 出力形式” (248 ページ)
- “\$UCS2L 出力形式” (244 ページ)
- “\$UTF8X 出力形式” (263 ページ)

入力形式:

- “\$UCS2B 入力形式” (510 ページ)
- “\$UCS2L 入力形式” (512 ページ)
- “\$UCS2X 入力形式” (514 ページ)
- “\$UCS2XE 入力形式” (516 ページ)
- “\$UTF8X 入力形式” (529 ページ)

\$UCS2XE 出力形式

ネイティブエンディアン、16ビット、UCS2形式のUNICODE(ユニコード)文字列を処理し、その文字列を現在のSASセッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS2XE*w*.

説明

w

出力フィールドの幅を指定します。16ビットサイズのUNICODE文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

詳細

\$UCS2XEw.出力形式では、現在の SAS セッションのエンコーディングで文字列を書き出します。ネイティブエンディアン、16 ビット、UCS2 (universal character set code in two octets)形式の UNICODE (ユニコード)文字列を処理します。

比較

\$UCS2XEw.出力形式では、\$UCS2Xw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
	大
<pre>x='e5a4a7'x; put x \$utf8xe10.;</pre>	

関連項目:

出力形式:

- ["\\$UCS2X 出力形式" \(247 ページ\)](#)

入力形式:

- ["\\$UCS2X 入力形式" \(514 ページ\)](#)
- ["\\$UCS2XE 入力形式" \(516 ページ\)](#)

\$UCS4B 出力形式

現在の SAS セッションのエンコーディングを使用する文字列を処理し、ビッグエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS4Bw.

説明

w

出力フィールドの幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

詳細

\$UCS4Bw.出力形式では、ビッグエンディアン、32ビット、UCS4 (universal character set code in four octets)形式の UNICODE (ユニコード)文字列を書き出します。現在の SAS セッションのエンコーディングを使用する文字列を処理します。

比較

\$UCS4Bw.出力形式では、\$UCS4BEw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
<code>x = '大'; put x \$ucs4b4.;</code>	'00005927'x (binary)

関連項目:

出力形式:

- “\$UCS2L 出力形式” (244 ページ)
- “\$UCS2X 出力形式” (247 ページ)
- “\$UCS4BE 出力形式” (251 ページ)
- “\$UCS4L 出力形式” (252 ページ)
- “\$UCS4X 出力形式” (254 ページ)
- “\$UTF8X 出力形式” (263 ページ)

入力形式:

- “\$UCS2B 入力形式” (510 ページ)
- “\$UCS2L 入力形式” (512 ページ)
- “\$UCS2X 入力形式” (514 ページ)
- “\$UCS4B 入力形式” (517 ページ)
- “\$UCS4L 入力形式” (518 ページ)
- “\$UCS4X 入力形式” (519 ページ)
- “\$UTF8X 入力形式” (529 ページ)

\$UCS4BE 出力形式

ビッグエンディアン、32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS4BEw.

説明

w

出力フィールドの幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

詳細

\$UCS4BEw.出力形式では、現在の SAS セッションのエンコーディングで文字列を書き出します。ビッグエンディアン、32 ビット、UCS4 (universal character set code in four octets)形式の UNICODE (ユニコード)文字列を処理します。

比較

\$UCS4BEw.出力形式では、\$UCS4Bw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
x='000059270000004100000042'x; put x \$ucs4be.;	大 AB

関連項目:

出力形式:

- ["\\$UCS4B 出力形式" \(249 ページ\)](#)

入力形式:

- “\$UCS4B 入力形式” (517 ページ)

\$UCS4L 出力形式

現在の SAS セッションのエンコーディングを使用する文字列を処理し、リトルエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS4L*w*.

説明

w

出力フィールドの幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

詳細

\$UCS4L*w*.出力形式では、リトルエンディアン、32 ビット、UCS4 (universal character set code in four octets)形式の UNICODE (ユニコード)文字列を書き出します。現在の SAS セッションのエンコーディングを使用する文字列を処理します。

比較

\$UCS4L*w*.出力形式では、\$UCS4LE*w*.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
data _null_;	2759
x = '夫';	
y=put(x,\$ucs4l4.);	
put y \$hex.;	
run;	

関連項目:**出力形式:**

- “\$UCS2B 出力形式” (242 ページ)
- “\$UCS2X 出力形式” (247 ページ)
- “\$UCS4B 出力形式” (249 ページ)
- “\$UCS4LE 出力形式” (253 ページ)
- “\$UCS4X 出力形式” (254 ページ)
- “\$UTF8X 出力形式” (263 ページ)

入力形式:

- “\$UCS2B 入力形式” (510 ページ)
- “\$UCS2L 入力形式” (512 ページ)
- “\$UCS2X 入力形式” (514 ページ)
- “\$UCS4B 入力形式” (517 ページ)
- “\$UCS4L 入力形式” (518 ページ)
- “\$UCS4X 入力形式” (519 ページ)
- “\$UTF8X 入力形式” (529 ページ)

\$UCS4LE 出力形式

リトルエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS4LE*w*.

説明

w

出力フィールドの幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

詳細

\$UCS4LE*w*.出力形式では、現在の SAS セッションのエンコーディングで文字列を書き出します。リトルエンディアン、32 ビット、UCS4 (universal character set code in four octets)形式の UNICODE (ユニコード)文字列を処理します。

比較

\$UCS4LEw.出力形式では、\$UCS4LEw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
x ='275900004100000042000000'x; put x \$ucs4le.;	大 AB

関連項目:

出力形式:

- ["\\$UCS4L 出力形式" \(252 ページ\)](#)

入力形式:

- ["\\$UCS4L 入力形式" \(518 ページ\)](#)

\$UCS4X 出力形式

現在の SAS セッションのエンコーディングを使用する文字列を処理し、ネイティブエンディアン、32 ビット、UCS4 形式の UNICODE (ユニコード)文字列を書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS4X*w*.

説明

w

出力フィールドの幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

詳細

\$UCS4Xw.出力形式では、動作環境のバイトオーダーで、16 ビットの UCS2 (universal character set code in two octets)形式の UNICODE (ユニコード)文字列を書き出します。

比較

\$UCS4Xw.出力形式では、\$UCS4XEw.出力形式とは逆の処理を行います。同一の動作環境内のデータを変換する場合は、\$UCS4Xw.出力形式を使用します。異なる動作環境でデータを変換する場合は、\$UCS4Bw.出力形式または\$UCS4Lw.出力形式を使用します。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
<pre>x = '大'; put x \$ucs4x4;</pre>	'00005927'x (binary) or '27590000'x (little endian)

関連項目:

出力形式:

- ["\\$UCS2L 出力形式" \(244 ページ\)](#)
- ["\\$UCS4XE 出力形式" \(256 ページ\)](#)
- ["\\$UCS2X 出力形式" \(247 ページ\)](#)
- ["\\$UCS4B 出力形式" \(249 ページ\)](#)
- ["\\$UCS4L 出力形式" \(252 ページ\)](#)
- ["\\$UTF8X 出力形式" \(263 ページ\)](#)

入力形式:

- ["\\$UCS2B 入力形式" \(510 ページ\)](#)
- ["\\$UCS2L 入力形式" \(512 ページ\)](#)
- ["\\$UCS2X 入力形式" \(514 ページ\)](#)
- ["\\$UCS4B 入力形式" \(517 ページ\)](#)
- ["\\$UCS4B 出力形式" \(249 ページ\)](#)
- ["\\$UCS4L 入力形式" \(518 ページ\)](#)
- ["\\$UCS4X 入力形式" \(519 ページ\)](#)
- ["\\$UTF8X 入力形式" \(529 ページ\)](#)

\$UCS4XE 出力形式

ネイティブエンディアン、32ビット、UCS4形式のUNICODE(ユニコード)で文字列を処理し、その文字列を現在のSASセッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UCS4XEW.

説明

w

出力フィールドの幅を指定します。32ビットサイズのUNICODE文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

詳細

\$UCS4XEW.出力形式では、現在のSASセッションのエンコーディングで文字列を書き出します。ネイティブエンディアン、32ビット、UCS4(universal character set code in four octets)形式のUNICODE(ユニコード)文字列を処理します。

比較

\$UCS4XEW.出力形式では、\$UCS4XW.出力形式とは逆の処理を行います。

例

この例ではUTF-8エンコーディングを使用しています。

ステートメント	結果
	----+----1
x='275900004100000042000000'x; put x \$ucs4be4.;	✱ AB (little endian)

関連項目:

出力形式:

- ["\\$UCS4X 出力形式" \(254 ページ\)](#)

入力形式:

- ["\\$UCS4X 入力形式" \(519 ページ\)](#)

\$UESC 出力形式

現在の SAS セッションでエンコードされる文字列を処理し、その文字列を Unicode エスケープ(UESC)表記で書き出します。

カテゴリ: 文字

配置: 左

構文

\$UESC*w*.

説明

w

入力幅を指定します。

デフォルト 8

範囲 1-32767

詳細

0 から 9、a から z、A から Z など、すべての動作環境で使用可能でない文字は、UESC で表す必要があります。\$UESC*w*. はネストできます。

比較

\$UESC*w*.出力形式では、\$UESCE*w*.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+----2
x='大';	¥u5927
y='u5927'	¥uu5927
z='uu5927';	¥uuu5927
put x = \$uesc10.;	
put y = \$uesc10.;	
put z = \$uesc10.;	

関連項目:**出力形式:**

- ["\\$UESCE 出力形式" \(258 ページ\)](#)

入力形式:

- ["\\$UESC 入力形式" \(521 ページ\)](#)
- ["\\$UESCE 入力形式" \(522 ページ\)](#)

\$UESCE 出力形式

Unicode エスケープ(UESC)表記の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UESCE*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 1-32767

詳細

データが現在の SAS セッションのエンコーディングによってサポートされていない場合、データは UESC のまま残ります。

比較

\$UESCE*w*.出力形式では、\$UESC*w*.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1-----+----2

ステートメント	結果
x=put('¥u5927',\$uesce10.);	x=¥u5927
x=put('¥uu5927',\$uesce10.);	x=¥uu5927
x=put('¥uuu5927',\$uesce10.);	x=¥uuu5927

関連項目:

出力形式:

- ["\\$UESC 出力形式" \(257 ページ\)](#)

入力形式:

- ["\\$UESC 入力形式" \(521 ページ\)](#)
- ["\\$UESCE 入力形式" \(522 ページ\)](#)

\$UNCR 出力形式

現在の SAS セッションでエンコードされる文字列を処理し、その文字列を数値文字表現(NCR)で書き出します。

カテゴリ: 文字

配置: 左

構文

\$UNCR*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 1-32767

比較

\$UNCR*w*.出力形式では、\$UNCRE*w*.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1-----+----2

ステートメント	結果
<pre>x='91E5'x ; /* Japanese '大' in Shift-JIS */ y='abc'; put x \$uncr10.; put y \$uncr10.;</pre>	<pre>&#22823 abc</pre>

関連項目:

出力形式:

- [“\\$UNCRE 出力形式” \(260 ページ\)](#)

入力形式:

- [“\\$UNCR 入力形式” \(523 ページ\)](#)
- [“\\$UNCRE 入力形式” \(524 ページ\)](#)

\$UNCRE 出力形式

数値文字表現(NCR)の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UNCRE*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 1-32767

詳細

各国語の文字は NCR で表記する必要があります。

比較

\$UNCRE*w*.出力形式では、\$UNCR*w*.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
x='大abc'; put x \$uncr10.;	大 abc

関連項目:

出力形式:

- [“\\$UNCR 出力形式” \(259 ページ\)](#)

入力形式:

- [“\\$UNCR 入力形式” \(523 ページ\)](#)
- [“\\$UNCRE 入力形式” \(524 ページ\)](#)

\$UPAREN 出力形式

現在の SAS セッションでエンコードされる文字列を処理し、その文字列を Unicode かつこ(UPAREN)表記で書き出します。

カテゴリ: 文字

配置: 左

構文

\$UPAREN*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 7-32767

詳細

文字列は、かつこおよび Unicode の 16 進表現によってエンコードされます。

比較

\$UPAREN*w*.出力形式では、\$UPAREN*Ew*.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+----2----+----3----+
<code>x='大';</code>	<code><u5927></code>
<code>y='abc3';</code>	<code><u0061> <u0062> <u0063> <u0033></code>
<code>put x \$uparen7.;</code>	
<code>put y \$uparen28.;</code>	

関連項目:

出力形式:

- [“\\$UPARENE 出力形式” \(262 ページ\)](#)

入力形式:

- [“\\$UPAREN 入力形式” \(525 ページ\)](#)
- [“\\$UPARENE 入力形式” \(527 ページ\)](#)

\$UPARENE 出力形式

Unicode カッコ(UPAREN)内の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

`$UPARENEw.`

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 1-32767

比較

`$UPARENEw.`出力形式では、`$UPARENw.`出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+
<pre>x='<u0061><u0062><u0063><u0033>'; put x \$uparene4.;</pre>	abc3

関連項目:

出力形式:

- [“\\$UPAREN 出力形式” \(261 ページ\)](#)

入力形式:

- [“\\$UPAREN 入力形式” \(525 ページ\)](#)
- [“\\$UPARENE 入力形式” \(527 ページ\)](#)

\$UTF8X 出力形式

現在の SAS セッションのエンコーディングを使用する文字列を処理し、UTF-8 (Universal Transformation Format)エンコーディングの文字列を書き出します。

カテゴリ: 文字

配置: 左

構文

\$UTF8X*w*.

説明

w

出力フィールドの幅を指定します。変数が入る十分な文字幅を指定します。文字の幅は、個々の文字のコードポイント値によって異なります。

デフォルト 8

範囲 1-32767

比較

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1
<pre>x='91E5'x; /* Japanese '太' in Shift-JIS */ put x \$utf8x10.;</pre>	x='e5a4a7'x

関連項目:

出力形式:

- “\$UCS2B 出力形式” (242 ページ)
- “\$UCS2L 出力形式” (244 ページ)
- “\$UCS2X 出力形式” (247 ページ)

入力形式:

- “\$UCS2B 入力形式” (510 ページ)
- “\$UCS2L 入力形式” (512 ページ)
- “\$UCS2X 入力形式” (514 ページ)

\$UTF8XE 出力形式

UTF-8 (Universal Transformation Format)の文字列を処理し、その文字列を現在の SAS セッションのエンコーディングで書き出します。

カテゴリ: 文字

配置: 左

構文

\$UTF8XEW.

説明

w

出力フィールドの幅を指定します。変数が入る十分な文字幅を指定します。文字の幅は、個々の文字のコードポイント値によって異なります。

デフォルト 8

範囲 1-32767

比較

\$UTF8XEW.出力形式では、\$UTF8Xw.出力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----t----1
<pre>x = unicode('u5927'); put x \$utf8xe10.;</pre>	大

関連項目:

出力形式:

- ["\\$UTF8X 出力形式" \(263 ページ\)](#)

入力形式:

- ["\\$UTF8X 入力形式" \(529 ページ\)](#)

WEEKU 出力形式

U アルゴリズムを使用して、10 進数の週番号を書き出します。

カテゴリ: 日付と時間

配置: 左

構文

WEEKU*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 11

範囲 2-200

詳細

WEEKU*w*.出力形式では、週番号形式で書き出します。WEEKU*w*.出力形式では、指定された幅に応じてさまざまな形式で書き出します。U アルゴリズムでは、年内の週番号を使用して SAS 日付値を計算します(日曜日を週の最初の日と見なします)。週番号値は、先頭に 0 を付けた 0 から 53 の範囲の 10 進数として表され、最大値は 53 になります。たとえば、年の 5 週目は 05 として表されます。

幅、出力形式および例の詳細については、次の表を参照してください。

幅	出力形式	例
3-4	Www	w01
5-6	yyWww	03W01
7-8	yyWwwdd	03W0101
9-10	yyyyWwwdd	2003W0101
11-200	yyyy-Www-dd	2003-W01-01

比較

WEEKVw.出力形式では、01 から 53 の範囲の 10 進数として週番号を書き出します。各週は月曜日から始まり、年の第 1 週には 1 月 4 日と年の最初の木曜日の両方が含まれます。1 月の最初の月曜日が 2 日、3 日または 4 日の場合、それより前の日は前年の最後の週に組み込まれます。WEEKWw.出力形式では、00 から 53 の範囲の 10 進数として年の週番号を書き出します。第 1 週の最初の日は月曜日になります。WEEKUw.出力形式では、先頭に 0 を付けた 0 から 53 の範囲の 10 進数として年の週番号を書き出します(週の最初の日は日曜日になります)。

例

```
sasdate = '01JAN2003'd;
```

ステートメント	結果
	----+----1----+
v=put(sasdate,weeku3.); w=put(sasdate,weeku5.); x=put(sasdate,weeku7.); y=put(sasdate,weeku9.); z=put(sasdate,weeku11.); put v; put w; put x; put y; put z;	W00 03W00 03W0004 2003W0004 2003-W00-04

関連項目:

出力形式:

- [“WEEKV 出力形式” \(267 ページ\)](#)
- [“WEEKW 出力形式” \(268 ページ\)](#)

WEEKV 出力形式

V アルゴリズムを使用して、10 進数の週番号を書き出します。

カテゴリ: 日付と時間

配置: 左

構文

WEEKV*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 11

範囲 2-200

詳細

WEEKV*w*.出力形式では、指定された幅に応じてさまざまな形式で書き出します。V アルゴリズムでは、SAS 日付値を計算します。週番号値は、先頭に 0 を付けた 01 から 53 の範囲の 10 進数として表され、最大値が 53 になります。各週は月曜日から始まり、年の第 1 週は 1 月 4 日と年の最初の木曜日の両方を含む週です。1 月の最初の月曜日が 2 日、3 日または 4 日の場合、それより前の日は前年の最後の週に組み込まれます。たとえば、年の 5 週目は 06 として表されます。

幅、出力形式および例の詳細については、次の表を参照してください。

幅	出力形式	例
3-4	Www	w01
5-6	yyWww	03W01
7-8	yyWwwdd	03W0101
9-10	yyyyWwwdd	2003W0101
11-200	yyyy-Www-dd	2003-W01-01

比較

WEEKV*w*.出力形式では、01 から 53 の範囲の 10 進数として週番号を書き出します。各週は月曜日から始まり、年の第 1 週には 1 月 4 日と年の最初の木曜日の両方が含まれます。1 月の最初の月曜日が 2 日、3 日または 4 日の場合、それより前の日は前年の最後の週に組み込まれます。WEEKW*w*.出力形式では、00 から 53 の範囲の 10 進数として年の週番号を書き出します。第 1 週の最初の日は月

曜日になります。WEEKUw.出力形式では、先頭に0を付けた0から53の範囲の10進数として年の週番号を書き出します(週の最初の日は日曜日になります)。

例

```
sasdate='01JAN2003'd;
```

ステートメント	結果
	----+----1----+
v=put(sasdate,weekv3.); w=put(sasdate,weekv5.); x=put(sasdate,weekv7.); y=put(sasdate,weekv9.); z=put(sasdate,weekv11.); put v; put w; put x; put y; put z;	W01 03W01 03W0103 2003W0103 2003-W01-03

関連項目:

出力形式:

- [“WEEKU 出力形式” \(265 ページ\)](#)
- [“WEEKW 出力形式” \(268 ページ\)](#)

WEEKW 出力形式

W アルゴリズムを使用して、10進数の週番号を書き出します。

カテゴリ: 日付と時間

配置: 左

構文

WEEKW*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 11

範囲 2-200

詳細

WEEKWw.出力形式では、指定された幅に応じてさまざまな形式で書き出します。W アルゴリズムでは、年内の週番号を使用して SAS 日付値を計算します(月曜日を週の最初の日と見なします)。週番号値は、先頭に 0 を付けた 0 から 53 の範囲の 10 進数として表され、最大値は 53 になります。たとえば、年の 5 週目は 05 として表されます。

幅、出力形式および例の詳細については、次の表を参照してください。

幅	出力形式	例
3-4	Www	w01
5-6	yyWww	03W01
7-8	yyWwwdd	03W0101
9-10	yyyyWwwdd	2003W0101
11-200	yyyy-Www-dd	2003-W01-01

比較

WEEKVw.出力形式では、01 から 53 の範囲の 10 進数として週番号を書き出します。各週は月曜日から始まり、年の第 1 週には 1 月 4 日と年の最初の木曜日の両方が含まれます。1 月の最初の月曜日が 2 日、3 日または 4 日の場合、それより前の日は前年の最後の週に組み込まれます。WEEKWw.出力形式では、00 から 53 の範囲の 10 進数として年の週番号を書き出します。第 1 週の最初の日は月曜日になります。WEEKUw.出力形式では、先頭に 0 を付けた 0 から 53 の範囲の 10 進数として年の週番号を書き出します(週の最初の日は日曜日になります)。

例

```
sasdate = '01JAN2003'd;
```

ステートメント	結果
	----+----1----+
v=put(sasdate,weekw3.);	W03
w=put(sasdate,weekw5.);	03W03
x=put(sasdate,weekw7.);	03W0003
y=put(sasdate,weekw9.);	2003W0003
z=put(sasdate,weekw11.);	2003-W00-03
put v;	
put w;	
put x;	
put y;	
put z;	

関連項目:**出力形式:**

- [“WEEKU 出力形式” \(265 ページ\)](#)
- [“WEEKV 出力形式” \(267 ページ\)](#)

YEN 出力形式

円記号、桁区切り、小数点付きの数値を出力します。

カテゴリ: 数値
CAS

配置: 右

構文

YEN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 1-32

d

数値の小数点以下の桁数を指定します。

デフォルト 0

範囲 0-31

詳細

YEN*w.d* 出力形式は、先頭に円記号を付け、各値を 3 桁ごとにカンマで区切って数値を書き出します。

円記号文字のコードの 16 進表現は、EBCDIC のシステムでは 5B、ASCII のシステムでは 5C です。これらのコードが表す通貨記号は、他の国では異なる場合があります。

例

```
put cost yen10.2;

data _null_;
  value=1254.71;
  put value yen10.2;
run;
```

Cost	結果
	---+---1
1254.71	¥1,254.71

YYWEEKU 出力形式

曜日情報を除外し、U アルゴリズムを使用して、10 進数の週番号を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

YYWEEKU w .

説明

w
出力フィールドの幅を指定します。

デフォルト 7

範囲 2-8

詳細

YYWEEKU w .出力形式では、週番号形式で書き出します。YYWEEKU w .出力形式では、指定された幅に応じてさまざまな形式で書き出します。U アルゴリズムでは、年内の週番号を使用して SAS 日付値を計算します(日曜日を週の最初の日と見なします)。

幅、出力形式および例の詳細については、次の表を参照してください。

幅	出力形式	例
3-4	W w w	W01
5-6	yyW w w	07W01
7	yyyyW w w	2007W01
8	yyyy-W w w	2007-W01
9-above	無効	無効

比較

YYWEEKUw.出力形式は WEEKUw.出力形式と似ていますが、YYWEEKUw.出力形式では曜日情報を示しません。また、YYWEEKUw.出力形式では、8 より大きい幅は受け付けません。

例

```
sasdate = '01JAN2007'd;
```

ステートメント	結果
	----+----1----+
u=put(sasdate,yyweeku3.);	W00
v=put(sasdate,yyweeku4.);	W00
w=put(sasdate,yyweeku5.);	07W00
x=put(sasdate,yyweeku6.);	07W00
y=put(sasdate,yyweeku7.);	2007W00
z=put(sasdate,yyweeku8.);	2007-W00
put u;	
put v;	
put w;	
put x;	
put y;	
put z;	

関連項目:

出力形式:

- [“WEEKU 出力形式” \(265 ページ\)](#)

YYWEEKV 出力形式

曜日情報を除外し、V アルゴリズムを使用して、10 進数の週番号を書き出します。

カテゴリ: 日付と時間

CAS

配置: 左

構文

YYWEEKVw.

説明

w

出力フィールドの幅を指定します。

デフォルト 7

範囲 2-8

詳細

YYWEEKVw.出力形式では、指定された幅に応じてさまざまな形式で書き出します。V アルゴリズムでは、SAS 日付値を計算します。週番号値は、先頭に 0 を付けた 01 から 53 の範囲の 10 進数として表され、最大値が 53 になります。各週は月曜日から始まり、年の第 1 週は 1 月 4 日と年の最初の木曜日の両方を含む週です。1 月の最初の月曜日が 2 日、3 日または 4 日の場合、それより前の日は前年の最後の週に組み込まれます。たとえば、年の 5 週目は 06 として表されません。

幅、出力形式および例の詳細については、次の表を参照してください。

幅	出力形式	例
3-4	Www	w01
5-6	yyWww	07W01
7	yyyyWww	2007W01
8	yyyy-Www	2007-W01
9-above	無効	無効

比較

YYWEEKVw.出力形式は WEEKVw.出力形式と似ていますが、YYWEEKVw.出力形式では曜日情報を示しません。また、YYWEEKVw.出力形式では、8 より大きい幅は受け付けません。

例

```
sasdate = '01JAN2007'd;
```

ステートメント	結果
	----+----1----+
u=put(sasdate,yyweekv3.);	W01
v=put(sasdate,yyweekv4.);	W01
w=put(sasdate,yyweekv5.);	07W01
x=put(sasdate,yyweekv6.);	07W01
y=put(sasdate,yyweekv7.);	2007W01
z=put(sasdate,yyweekv8.);	2007-W01
put u;	
put v;	
put w;	
put x;	
put y;	
put z;	

関連項目:**出力形式:**

- [“WEEKV 出力形式” \(267 ページ\)](#)

YYWEEKW 出力形式

曜日情報を除外し、W アルゴリズムを使用して、10 進数の週番号を書き出します。

カテゴリ: 日付と時間
CAS

配置: 左

構文

YYWEEKW*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 7

範囲 2-8

詳細

YYWEEKW*w*出力形式では、指定された幅に応じてさまざまな形式で書き出します。W アルゴリズムでは、年内の週番号を使用して SAS 日付値を計算します。

幅、出力形式および例の詳細については、次の表を参照してください。

幅	出力形式	例
3-4	Www	W01
5-6	yyWww	07W01
7	yyyyWww	2007W01
8	yyyy-Www	2007-W01
9-above	無効	無効

比較

YYWEEKWw.出力形式は WEEKWw.出力形式と似ていますが、YYWEEKWw.出力形式では曜日情報を示しません。また、YYWEEKWw.出力形式では、8 より大きい幅は受け付けません。

例

```
sasdate = '01JAN2007'd
```

ステートメント	結果
	----+----1----+
u=put(sasdate,yyweekw3.);	W01
v=put(sasdate,yyweekw4.);	W01
w=put(sasdate,yyweekw5.);	07W01
x=put(sasdate,yyweekw6.);	07W01
y=put(sasdate,yyweekw7.);	2007W01
z=put(sasdate,yyweekw8.);	2007-W01
put u;	
put v;	
put w;	
put x;	
put y;	
put z;	

関連項目:

出力形式:

- [“WEEKW 出力形式” \(268 ページ\)](#)

5 部

各国語サポート関連の関数

10 章	
SAS 文字列関数の国際化の互換性	279
11 章	
関数での VARCHAR サポート	297
12 章	
各国語サポート関連の関数のディクショナリ	301

10 章

SAS 文字列関数の国際化の互換性

SAS 文字列関数の国際化の互換性 279

SAS 文字列関数の国際化の互換性

SAS では、文字データを簡単に操作できる文字列関数と CALL ルーチンを提供します。従来の SAS 文字列関数の多くは、1 文字のサイズが常に 1 バイトであることを前提としています。この処理は、1 バイト文字セット(SBCS)のデータでは正常に動作します。ただし、これらの関数と CALL ルーチンの一部を 2 バイト文字セット(DBCS)または UTF-8 のようなマルチバイト文字セット(MBCS)のデータに使用すると、多くの場合データは適切に処理されず、文字列関数から正しい結果が得られません。

この問題を解決するために、DBCS と MBCS データを慎重に処理する文字列操作が可能な、*K 関数*と呼ばれる文字列関数と CALL ルーチンが導入されました。K 関数には文字列のサイズに関して何の前提もありません。表 10.1 (280 ページ)Dan: ユーザーがオンラインで読んでいる場合はどうなりますか? このページ(適切な言葉?)では、各 SAS 文字列関数の I18N 互換性のレベルを説明します。I18N とは、internationalization (国際化)の略称です。互換性とは、特定の文字列関数を使用するプログラムが、プログラムを変更することなく異なる言語とロケールに適合できるかどうかを意味します。

ユーザーが K 関数を適切に使用するには、バイトベースのオフセット長と文字ベースのオフセット長の違いを理解することが重要です。

バイトベースのオフセットでは、文字に対して指定された開始位置が、文字列中の当該文字のバイト位置であると想定します。SBCS データの場合、1 文字は常に 1 バイト長なので、文字列中の 2 番目の文字は 2 バイト目から始まると考えてかまいません。しかし、文字列中のデータがマルチバイトデータであれば、2 バイト目のデータは、データやそのエンコーディングに応じて、次のいずれかになります。

- 文字列中、2 番目の文字
- 2 バイト文字の第 2 バイト
- 文字列中、1 番目の文字の第 1 バイト

バイトベースの長さは、文字列を構成するバイトの数を表します。

文字ベースのオフセットでは、指定された位置が、文字列中の当該文字の位置であると想定します。どのエンコーディングでも、文字ベースの位置 2 にあるのは、文字列中の 2 番目の文字です。文字列中の文字のサイズが分かっていると想定はできません。

文字ベースの長さは、文字列を構成する文字の数を表します。

K 関数は文字ベースのオフセットや長さを使います。文字列中の各文字のバイト位置は考慮されません。SAS Viya で UTF-8 のデータを処理する場合、K 関数の利用を検討してください。

SAS 文字列関数および対応する K 関数の機能を表す例を以下に示します。

UTF8 エンコーディングの場合、STR='E282AC313233'x という文字列は 6 バイトからなり、4 つの文字を表します。

表 10.1 UTF-8 での文字列表現

16 進表現	E282AC	31	32	33
	€	1	2	3

次の例では、この同じ 16 進文字列を SUBSTR 関数に渡していますが、戻り値は UTF-8 の予期しない値です。予期される戻り値は、1 文字目に当たるユーロ記号です。しかし、結果は無効な文字になっています。LENGTH 関数はバイト指向であるため、データの個々のバイトが読み取られます。SUBSTR 関数からは先頭文字の先頭バイトが返されますが、これは不完全な値です。なぜならば先頭文字は 3 バイトに基づいてエンコードされているからです。

表 10.2 正規関数を使用したコード例

ステートメント	結果
<pre>/* SAS program to submit in a UTF-8 SAS session */ data _null_; str="€123"; s=substr(str,1,1); sl=length(s); l=length(str); put str= \$hex16. /s= sl= / s= \$hex. /l=; run;</pre>	<pre>str=E282AC313233 s= sl=1 s=E22020202020 l=6</pre>

表 10.3 (280 ページ)のコードはこれを解決するため、1 バイトデータしか正常に扱えない SUBSTR()関数の代わりに、1 バイトデータでもマルチバイトデータでも動作する、KSUBSTR()関数を使っています。他の文字列関数についても直接的な置き換えが可能で、LENGTH()または INDEX()関数を変更して、KLENGTH()または KINDEX()関数を使用します。2 つの関数はデータの読み込み方が異なるため、マルチバイト環境で実行した場合、異なる結果が得られます。

表 10.3 Code Example Using K Functions

ステートメント	結果
<pre>/* SAS program to submit in a UTF-8 SAS session */ data _null_; str="€123"; s=ksubstr(str,1,1); sl=klength(s); l=klength(str); put str= \$hex16. /s= sl= / s= \$hex. /l=; run;</pre>	<pre>str=E282AC313233 s=€ l=4</pre>

SAS 文字列処理関数をすべて K 関数に置換する際には、あらかじめ SAS プログラムを検査しておいてください。文字列関数が、単一バイト文字のみを含んでいるデータを処理する場合、K 関数は必要ありません。たとえば、XML タグを含んでいる文字列に対しては、K 関数を使用する必要はありません。SAS プログラムに含まれているデータの内容と、それがどのように処理されるかを把握することにより、SAS コードの不要な更新を回避できます。バイナリデータ処理は文字列処理用の K 関数ではサポートされていません。K 関数は、処理対象となる文字列が、現在のセッションのエンコーディングに一致するものと予想します。UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

文字列関数は、関数が MBCS または SBCS のデータを処理できるかどうかに応じて I18N レベルが割り当てられます。レベルの説明を次に示します。

I18N レベル 0

この関数は SBCS データ向けに設計されています。この関数を MBCS データの処理に使用しないでください。

I18N レベル 1

MBCS データを処理する場合、可能な限りこの関数の使用は避けてください。I18N レベル 1 の関数は、特定の環境下では DBCS または MBCS エンコーディングを使用すると正常に動作しない場合があります。

I18N レベル 2

この関数は SBCS、MBCS(UTF-8)データに使用できます。

表 10.4 SAS 文字列関数

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“ANORM420 関数” (p. 307)	EBCDIC420 でエンコードされた入力文字列を基に正規化された文字列を返します。		X	
“ANYALNUM Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から英数字を検索し、最初に検索された文字の位置を返します。		X	
“ANYALPHA Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から英字を検索し、最初に検索された文字の位置を返します。		X	
“ANYCNTRL Function” (SAS Viya Functions and CALL Routines: Reference)	文字列からコントロール文字を検索し、最初に検索された文字の位置を返します。			X
“ANYDIGIT Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から数字を検索し、最初に検索された数字の位置を返します。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“ANYFIRST Function” (SAS Viya Functions and CALL Routines: Reference)	VALIDVARNAME=V7 において、SAS 変数名の開始文字として有効な文字を文字列から検索し、最初に検索された文字の位置を返します。			X
“ANYGRAPH Function” (SAS Viya Functions and CALL Routines: Reference)	文字列からグラフィカル文字を検索し、最初に検索された文字の位置を返します。		X	
“ANYLOWER Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から小文字を検索し、最初に検索された文字の位置を返します。		X	
“ANYNAME Function” (SAS Viya Functions and CALL Routines: Reference)	VALIDVARNAME=V7 において、SAS 変数名として有効な文字を文字列から検索し、最初に検索された文字の位置を返します。			X
“ANYPRINT Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から印刷可能な文字を検索し、最初に検索された文字の位置を返します。		X	
“ANYPUNCT Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から句読文字を検索し、最初に検索された文字の位置を返します。		X	
“ANYSPEACE Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から空白文字(空白、水平タブ、垂直タブ、キャリッジリターン、ラインフィード、フォームフィード)を検索します。最初に検索された文字の位置を返します。		X	
“ANYUPPER Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から大文字を検索し、最初に検索された文字の位置を返します。		X	

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“ANYXDIGIT Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	数字を表す 16 進法の文字を文字列から検索し、最初に検索された文字の位置を返します。			X
“BASECHAR 関数” (p. 312)	文字を基本文字に変換します。			X
“BYTE Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	ASCII 照合順序または EBCDIC 照合順序の 1 文字を返します。	X		
“CAT Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	先頭または末尾の空白を削除せずに、連結文字列を返します。			X
“CATQ Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	項目を区切る区切り文字を使用したり、区切り文字を含んでいる文字列に引用符を追加したりして、文字値または数値を連結します。			X
“CATS Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	先頭と末尾の空白を削除して、連結文字列を返します。			X
“CATT Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	末尾の空白を削除して、連結文字列を返します。			X
“CATX Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	先頭と末尾の空白を削除し、区切り文字を挿入して、文字列を返します。			X
“CHAR Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列内の指定された位置にある 1 文字を返します。	X		
“CHOOSEC Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	引数のリストからの選択結果を表す文字値を返します。			X
“CHOOSEN Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	引数のリストからの選択結果を表す数値を返します。			X
“COALESCEC Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	数値の引数のリストからの最初の非欠損値を返します。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“COLLATE Function” (SAS Viya Functions and CALL Routines: Reference)	ASCII 照合順序または EBCDIC 照合順序の文字列を返します。	X		
“COMPARE Function” (SAS Viya Functions and CALL Routines: Reference)	2 つの文字列を比較し、異なる文字が検出された場合には最も左にある文字の位置を返し、異なる文字が検出されない場合には 0 を返します。	X		
“COMPBL Function” (SAS Viya Functions and CALL Routines: Reference)	文字列内のワード間にある複数の空白を取り除きます。			X
“COMPGED Function” (SAS Viya Functions and CALL Routines: Reference)	2 つの文字列間の一般化編集距離を返します。	X		
“COMPLEV Function” (SAS Viya Functions and CALL Routines: Reference)	2 つの文字列間のレーベンシュタインの編集距離を返します。	X		
“COMPRESS Function” (SAS Viya Functions and CALL Routines: Reference)	元の文字から指定した文字を削除した文字列を返します。	X		
“COUNT Function” (SAS Viya Functions and CALL Routines: Reference)	指定した部分文字列が文字列内に含まれる回数を数えます。	X		
“COUNTC Function” (SAS Viya Functions and CALL Routines: Reference)	文字のリストに表示される(または表示されない)文字列内の文字の個数を数えます。	X		
“COUNTW Function” (SAS Viya Functions and CALL Routines: Reference)	文字列内の単語の数を数えます。	X		
“DEQUOTE Function” (SAS Viya Functions and CALL Routines: Reference)	引用符で始まる文字列から、対となる引用符を削除し、閉じる側の引用符より右側にあるすべての文字を削除します。		X	

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“FIND Function” (SAS Viya Functions and CALL Routines: Reference)	文字列内の特定の部分文字列を検索します。	X		
“FINDC Function” (SAS Viya Functions and CALL Routines: Reference)	文字のリストにある各文字を文字列から検索します。	X		
“FINDW Function” (SAS Viya Functions and CALL Routines: Reference)	単語の、文字列内での文字位置を返すか、単語の、文字列内での番号を返します。		X	
“FIRST Function” (SAS Viya Functions and CALL Routines: Reference)	文字列内の先頭文字を返します。	X		
“GETLOCENV 関数” (p. 315)	現在のロケール/言語環境を返します。			X
“GETPXLANGUAGE 関数” (p. 316)	現在の 2 文字の言語コードを返します。			X
“GETPXLOCALE 関数” (p. 317)	SAS ロケールの POSIX ロケール値を返します。			X
“GETPXREGION 関数” (p. 318)	現在の 2 文字の地域コードを返します。			X
“HTMLDECODE Function” (SAS Viya Functions and CALL Routines: Reference)	HTML 数値文字参照または HTML 文字実体参照を含む文字列をデコードし、デコードされた文字列を返します。			X
“HTMLENCODE Function” (SAS Viya Functions and CALL Routines: Reference)	HTML 文字実体参照を使用して文字列をエンコードし、エンコードされた文字列を返します。			X
“IFC Function” (SAS Viya Functions and CALL Routines: Reference)	式の真、偽、欠損に基づいて文字値を返します。			X
“IFN Function” (SAS Viya Functions and CALL Routines: Reference)	式の真、偽、欠損に基づいて数値を返します。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“INDEX Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字式から文字列を検索し、最初に検索された文字列の最初の文字の位置を返します。	X		
“INDEXC Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字式から指定した文字すべてを検索し、その文字の位置を返します。	X		
“INDEXW Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字式から単語として指定した文字列を検索し、単語の最初の文字の位置を返します。	X		
“KCOMPARE 関数” (p. 319)	文字式の比較結果を返します。			X
“KCOMPRESS 関数” (p. 321)	指定された文字を文字式から削除します。			X
“KCOUNT 関数” (p. 322)	式の 2 バイト文字の数を返します。			X
“KCVT 関数” (p. 322)	データのある種類のエンコーディングデータから別の種類のエンコーディングデータに変換します。		X	
“KINDEX 関数” (p. 324)	文字式から文字列を検索します。			X
“KINDEXB 関数” (p. 325)	文字式から文字列を検索します。			X
“KINDEXCB 関数” (p. 327)	文字式から指定した文字を検索します。			X
“KINDEXC 関数” (p. 326)	文字式から指定した文字を検索します。			X
“KLEFT 関数” (p. 328)	不要な先頭の DBCS の空白とシフトアウト/シフトイン(SO-SI)を削除し、文字式を左詰めにします。			X
“KLENGTH 関数” (p. 330)	引数の長さを返します。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“KLOWCASE 関数” (p. 331)	引数のすべての文字を小文字に変換します。			X
“KPROPCASE 関数” (p. 332)	中国語、日本語、韓国語、台湾語(CJKT)文字を変換します。			X
“KPROPCHAR 関数” (p. 335)	特殊文字を標準文字に変換します。			X
“KPROPDATA 関数” (p. 336)	印刷不可文字を削除または変換します。			X
“KREVERSE 関数” (p. 338)	文字式を逆にします。			X
“KRIGHT 関数” (p. 339)	末尾の DBCS の空白と SO-SI を削除し、文字式を右詰めにします。			X
“KSCAN 関数” (p. 340)	指定された単語を文字式から選択します。			X
“KSTRCAT 関数” (p. 341)	複数の文字式を連結します。			X
“KSTRIP 関数” (p. 342)	文字列から先頭と末尾の空白を削除します。			X
“KSUBSTR 関数” (p. 344)	引数から部分文字列を抽出します。			X
“KSUBSTRN 関数” (p. 345)	部分文字列を返します。長さが 0 の結果も返せます。			X
“KSUBSTRB 関数” (p. 346)	引数内の部分文字列のバイト位置に従って、引数から部分文字列を抽出します。			X
“KTRIM 関数” (p. 349)	末尾の DBCS の空白と SO-SI を文字式から削除します。			X
“KTRUNCATE 関数” (p. 350)	マルチバイト文字を壊すことなく、文字列をバイト単位で、指定した長さに切ります。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“KUPCASE 関数” (p. 351)	引数内のすべての文字を大文字に変換します。			X
“KUPDATE 関数” (p. 352)	文字値の内容を挿入、削除および置換します。			X
“KUPDATEB 関数” (p. 354)	引数内の文字値のバイト位置に従って、文字値の内容を挿入、削除および置換します。			X
“KUPDATES 関数” (p. 356)	文字値の内容を挿入、削除および置換します。			X
“KVERIFYB 関数” (p. 359)	式に固有の最初の文字の位置を返します。			X
“KVERIFY 関数” (p. 358)	式に固有の最初の文字の位置を返します。			X
“LEFT Function” (SAS Viya Functions and CALL Routines: Reference)	文字列を左詰めにします。			X
“LENGTH Function” (SAS Viya Functions and CALL Routines: Reference)	末尾の空白を除いた文字列の長さを返します。文字列が空白の場合には、1 を返します。			X
“LENGTHC Function” (SAS Viya Functions and CALL Routines: Reference)	末尾の空白を含めた文字列の長さを返します。			X
“LENGTHM Function” (SAS Viya Functions and CALL Routines: Reference)	文字列に割り当てられたメモリの量を返します(バイト単位)。			X
“LENGTHN Function” (SAS Viya Functions and CALL Routines: Reference)	末尾の空白を除いた文字列の長さを返します。			X
“LOWCASE Function” (SAS Viya Functions and CALL Routines: Reference)	引数のすべての文字を小文字に変換します。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“MD5 Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	指定された文字列のメッセージダイジェストの結果を返します。			X
“MISSING Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	引数が欠損値を含むかどうかの結果を表す数値を返します。			X
“MVALID Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	SAS メンバ名として使用する文字列の有効性を確認します。			X
“NLDATE 関数” (p. 360)	SAS 日付値を、日付のディスクリプタを使用して指定したロケールの日付値に変換します。			X
“NLDATM 関数” (p. 363)	SAS 日時値を、日時形式のディスクリプタを使用して指定したロケールの時間値に変換します。			X
“NLTIME 関数” (p. 366)	SAS 時間値または SAS 日時値を、NLTIME ディスクリプタを使用して指定したロケールの時間値に変換します。			X
“NLITERAL Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	指定した文字列を SAS 名前リテラルに変換します。			X
“NOTALNUM Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列から英数字ではない文字を検索し、最初に検索された文字の位置を返します。		X	
“NOTALPHA Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列から英字ではない文字を検索し、最初に検索された文字の位置を返します。		X	

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“NOTCNTRL Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列からコントロール文字ではない文字を検索し、最初に検索された文字の位置を返します。			X
“NOTDIGIT Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列から数字ではない文字を検索し、最初に検索された文字の位置を返します。			X
“NOTFIRST Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	VALIDVARNAME=V7 の場合、SAS 変数名として無効な開始文字を文字列から検索し、最初に検索された文字の位置を返します。			X
“NOTGRAPH Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列からグラフィカル文字ではない文字を検索し、最初に検索された文字の位置を返します。		X	
“NOTLOWER Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列から小文字ではない文字を検索し、最初に検索された文字の位置を返します。		X	
“NOTNAME Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	VALIDVARNAME=V7 の場合、SAS 変数名として無効な文字を文字列から検索し、最初に検索された文字の位置を返します。			X
“NOTPRINT Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列から印刷可能文字ではない文字を検索し、最初に検索された文字の位置を返します。	X		
“NOTPUNCT Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列から句読文字ではない文字を検索し、最初に検索された文字の位置を返します。		X	

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“NOTSPACE Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から空白文字(空白、水平タブ、垂直タブ、キャリッジリターン、ラインフィード、フォームフィード)でない文字を検索し、最初に検索された文字の位置を返します。		X	
“NOTUPPER Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から大文字ではない文字を検索し、最初に検索された文字の位置を返します。		X	
“NOTXDIGIT Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から 16 進数字ではない文字を検索し、最初に検索された文字の位置を返します。			X
“INVALID Function” (SAS Viya Functions and CALL Routines: Reference)	SAS 変数名として使用する文字列の有効性を確認します。			X
“PROPCASE Function” (SAS Viya Functions and CALL Routines: Reference)	引数のすべての単語を適切に大文字と小文字に変換します。			X
“PRXCHANGE Function” (SAS Viya Functions and CALL Routines: Reference)	パターンマッチング置換を実行します。	X		
“PRXMATCH Function” (SAS Viya Functions and CALL Routines: Reference)	パターンマッチングを実行し、パターンが見つかった位置を返します。	X		
“PRXPAREN Function” (SAS Viya Functions and CALL Routines: Reference)	パターン内に一致が存在する最後のブラケットマッチを返します。	X		
“PRXPARSE Function” (SAS Viya Functions and CALL Routines: Reference)	文字列値のパターンマッチングに使用できる Perl 正規表現(PRX)をコンパイルします。	X		
“PRXPOSN Function” (SAS Viya Functions and CALL Routines: Reference)	キャプチャバッファの値を含んでいる文字列を返します。	X		

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“PUT Function” (SAS Viya Functions and CALL Routines: Reference)	指定された出力形式を使用して値を返します。			X
“PUTC Function” (SAS Viya Functions and CALL Routines: Reference)	文字列の出力形式を実行時に指定できます。			X
“PUTN Function” (SAS Viya Functions and CALL Routines: Reference)	数値の出力形式を実行時に指定できます。			X
“QUOTE Function” (SAS Viya Functions and CALL Routines: Reference)	文字値に二重引用符を付加します。			X
“RANK Function” (SAS Viya Functions and CALL Routines: Reference)	ASCII 照合順序または EBCDIC 照合順序による文字の位置を返します。	X		
“REPEAT Function” (SAS Viya Functions and CALL Routines: Reference)	最初の引数を $n+1$ 回繰り返した文字値を返します。			X
“REVERSE Function” (SAS Viya Functions and CALL Routines: Reference)	文字列を逆にします。	X		
“RIGHT Function” (SAS Viya Functions and CALL Routines: Reference)	文字式を右詰めにします。			X
“SASMSG 関数” (p. 367)	データセットからメッセージを返します。返されるメッセージは現在のロケールと指定されたキーに基づきます。			X
“SASMSGL 関数” (p. 370)	データセットからメッセージを返します。メッセージは、指定されたロケール値と指定されたキー値に基づきます。			X
“SCAN Function” (SAS Viya Functions and CALL Routines: Reference)	文字列から n 番目の単語を返します。	X		
“SETLOCALE 関数” (p. 375)	現在の SAS ロケールにロケールキーを指定します。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“SHA256 Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	指定された文字列のメッセージダイジェストの結果を返します。			X
“SORTKEY 関数” (p. 372)	言語並べ替えキーを作成します。			X
“SOUNDEX Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列をエンコードして検索しやすくします。	X		
“SPEDIS Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	2つの単語の一致尤度を調べて、2つの単語間のスペルの違いをコストで表します。	X		
“STRIP Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	先頭と末尾の空白を削除して文字列を返します。			X
“SUBPAD Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	必要に応じて空白埋め込みを使用し、指定した長さの部分文字列を返します。	X		
“SUBSTR (left of =) Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	引数から部分文字列を抽出します。	X		
“SUBSTR (right of =) Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	引数から部分文字列を抽出します。	X		
“SUBSTRN Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	部分文字列を返します。長さが0の結果も返せます。	X		
“TRANSLATE Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列に含まれる特定の文字を置き換えます。	X		
“TRANSTRN Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列に含まれる特定の部分文字列をすべて置き換えるか削除します。			X
“TRANTAB 関数” (392 ページ)	指定された変換テーブルを使用してデータをトランスコードします。	X		

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“TRANWRD Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列に含まれる特定の部分文字列をすべて置き換えるか削除します。			X
“TRIM Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字列から末尾の空白を取り除きます。文字列が欠損値の場合は、1つの空白を返します。			X
“TRIMN Function” (SAS <i>Viya Functions and CALL Routines: Reference</i>)	文字式から末尾の空白を取り除きます。文字式が欠損値の場合は、長さが0の文字列を返します。			X
“TZONEDSTNAME 関数” (p. 387)	夏時間名を返します。			X
“TZONEDSTOFF 関数” (p. 388)	指定された夏時間のタイムゾーンオフセット値を返します。			X
“TZONEID 関数” (p. 382)	現在のタイムゾーンIDを返します。			X
“TZONENAME 関数” (p. 384)	現在の標準時間または夏時間のタイムゾーン名を返します。			X
“TZONEOFF 関数” (p. 385)	ユーザーのタイムゾーンオフセットを返します。			X
“TZONES2U 関数” (p. 386)	UTC 日時値を SAS 日時値へ変換します。			X
“TZONESTTNAME 関数” (p. 389)	標準のタイムゾーン名を返します。			X
“TZONESTTOFF 関数” (p. 390)	指定された標準時間のタイムゾーンオフセット値を返します。			X
“TZONEU2S 関数” (p. 391)	UTC 日時値を SAS 日時値へ変換します。			X

関数	説明	I18N レベル 0	I18N レベル 1	I18N レベル 2
“UNICODE 関数” (p. 393)	Unicode 文字を現在の SAS セッションエンコーディングに変換します。			X
“UNICODEC 関数” (p. 394)	現在の SAS セッションエンコーディングの文字を Unicode 文字に変換します。			X
“UNICODELEN 関数” (p. 396)	Unicode データの文字単位の長さを示します。			X
“UNICODEWIDTH 関数” (p. 397)	Unicode データの表示単位の長さを示します。			X
“UPCASE Function” (SAS Viya Functions and CALL Routines: Reference)	引数内のすべての文字を大文字に変換します。			X
“URLDECODE Function” (SAS Viya Functions and CALL Routines: Reference)	URL のエスケープ構文を使用してデコードされた文字列を返します。		X	
“URLENCODE Function” (SAS Viya Functions and CALL Routines: Reference)	URL のエスケープ構文を使用してエンコードされた文字列を返します。		X	
“VERIFY Function” (SAS Viya Functions and CALL Routines: Reference)	他の文字列に存在しない文字の最初の出現位置を返します。			X
“WHICHC Function” (SAS Viya Functions and CALL Routines: Reference)	1 番目の引数と等しい文字列値を検索し、最初に一致した値のインデックスを返します。			X

11 章 関数での VARCHAR サポート

文字列関数の VARCHAR データタイプ 297

文字列関数の VARCHAR データタイプ

SAS Viya には、文字データを扱う VARCHAR というデータタイプがあります。VARCHAR データタイプの変数は、可変幅であり、文字としてのセマンティックスに従います。SAS 本来の文字データタイプと違って、固定幅ではなく、また、バイトとしてのセマンティックスでもありません。VARCHAR 変数を文字列関数(従来からある文字列関数、K 関数を含む)に渡すと、SAS はその長さが文字数を表していると想定します。VARCHAR 変数のオフセットは、文字列中の文字の位置を表します。固定幅文字データタイプは、バイナリデータに向いています。

VARCHAR は CAS テーブルでサポートされ、data_NULL_step で使われます。VARCHAR カラムを扱えるプロシジャもあります。VARCHAR カラムに非対応のプロシジャは、カラムを固定幅文字カラムに変換します。データセットで VARCHAR を扱うことはできません。

SUBSTR 関数 **substr(X, 10, 2)** でこのデータタイプを使っている例を示します。

変数 X が固定幅文字変数として定義されている場合、SUBSTR は文字列中の 10 番目と 11 番目のバイトを返します。X が VARCHAR 変数として定義されている場合、SUBSTR は文字列中の 10 番目と 11 番目の文字を返します。

データタイプが長さ 10 の VARCHAR である変数は、10 個までの文字を保持できます。

固定幅データタイプと VARCHAR データタイプの違いを表に示します。固定幅文字データタイプはバイトとしてのセマンティックス、VARCHAR は文字としてのセマンティックスに従います。

データ型	従来型の文字列関数	K 関数
固定幅の文字	バイト	文字
VARCHAR	文字	文字

SAS Viya セッションのエンコーディングである UTF-8 は、1 バイトのデータもマルチバイトのデータも扱えます。マルチバイトの UTF-8 文字の例としてユーロ文字(€)があり、これは 3 バイトを占めます。文字列の表記を次の表に示します。€123.文字€は 3 バイト、数字はそれぞれ 1 バイトを占めます。

文字	€			1	2	3
16 進値	E2	82	AC	31	32	33

次の例では、固定幅のデータ型と VARCHAR データ型を比較します。表 11.1 (298 ページ)に、データ€123 の固定幅変数から得た結果を示します。表 11.2 (299 ページ)に、VARCHAR を使用した場合の結果を示します。

K 関数は固定幅文字データ(マルチバイト文字が含まれていても可)を対象とします。

1. 例の 1 行目では、固定幅文字カラムに文字列を代入しています。PUT ステートメントは文字を表示し、さらに対応する 16 進値も表示します。
2. 例の 2 行目では、SUBSTR 関数に位置として 2 を渡しています。変数 x は固定幅文字カラムなので、この位置は x の文字列中 2 番目のバイトと解釈されます。すなわち、文字€を構成する 2 番目のバイトです。したがって、xsub に代入される先頭 2 バイトは、UTF-8 データとしては無効です。xsub の先頭 2 バイトとしてここに表示される結果はごみ(あるいは Unicode の代替文字)です。
3. 例の 3 行目では、INDEX 関数を使って文字列中の'1'を検索しています。変数 x は固定幅文字カラムなので、INDEX 関数の戻り値は、文字列中の該当する文字のバイト位置です。結果は **xidx=4** となります。x 中、'1'は 4 番目のバイトにあるからです。

表 11.1 固定幅文字の例

ステートメント	結果
data _null; x='€123'; put x= x= \$hex6;	x=€123 x=E282AC xsub=-123 xsub=82AC31323320 xidx=4
xsub=substr(x,2); put xsub= xsub= \$hex.;	
xidx=index(x,'1'); put xidx=; run;	

1. 例の 1 行目では、VARCHAR カラムに文字列を代入しています。PUT ステートメントは文字を表示し、さらに対応する 16 進値も表示します。
2. 例の 2 行目では、SUBSTR 関数に位置として 2 を渡しています。この例で、変数 v は VARCHAR として定義されているので、この位置は v の文字列中 2 番目の文字と解釈されます。vsub に代入される結果は文字列'123'で、UTF-8 の有効な文字からなります。
3. 例の 3 行目では、INDEX 関数を使って文字列中の「1」を検索しています。変数 v は VARCHAR カラムなので、INDEX 関数の戻り値は、文字列中の該当する文字の位置です。結果は **vidx=2** となります。v 中、'1'は 2 番目の文字だからです。

表 11.2 VARCHAR 文字の例

ステートメント	結果
<pre>data _null; length v VARCHAR(6); v='€123'; put v= v= \$hex.;</pre>	<pre>v=€123 v=E282AC313233 vsub=123 vsub=31 323320202020202020202020202020202020 vidx=2</pre>
<pre>vsub=substr(v,2); put vsub= vsub= \$hex.;</pre>	
<pre>vidx=index(v,'1'); put vidx=; run;</pre>	

12 章

各国語サポート関連の関数のディクショナリ

カテゴリ別の関数	302
ディクショナリ	307
ANORM420 関数	307
BASECHAR 関数	312
ENCODCOMPAT 関数	313
ENCODISVALID 関数	314
GETLOCENV 関数	315
GETPXLANGUAGE 関数	316
GETPXLOCALE 関数	317
GETPXREGION 関数	318
KCOMPARE 関数	319
KCOMPRESS 関数	321
KCOUNT 関数	322
KCVT 関数	322
KINDEX 関数	324
KINDEXB 関数	325
KINDEXC 関数	326
KINDEXCB 関数	327
KLEFT 関数	328
KLENGTH 関数	330
KLOWCASE 関数	331
KPROPCASE 関数	332
KPROPCHAR 関数	335
KPROPDATA 関数	336
KREVERSE 関数	338
KRIGHT 関数	339
KSCAN 関数	340
KSTRCAT 関数	341
KSTRIP 関数	342
KSUBSTR 関数	344
KSUBSTRN 関数	345
KSUBSTRB 関数	346
KTRANSLATE 関数	348
KTRIM 関数	349
KTRUNCATE 関数	350
KUPCASE 関数	351
KUPDATE 関数	352
KUPDATEB 関数	354
KUPDATES 関数	356
KVERIFY 関数	358
KVERIFYB 関数	359

NLDATE 関数	360
NLDATM 関数	363
NLTIME 関数	366
SASMSG 関数	367
SASMSG1 関数	370
SORTKEY 関数	372
SETLOCALE 関数	375
TZONEID 関数	382
TZONENAME 関数	384
TZONEOFF 関数	385
TZONES2U 関数	386
TZONEDSTNAME 関数	387
TZONEDSTOFF 関数	388
TZONESTTNAME 関数	389
TZONESTTOFF 関数	390
TZONEU2S 関数	391
TRANTAB 関数	392
UNICODE 関数	393
UNICODEC 関数	394
UNICODELEN 関数	396
UNICODEWIDTH 関数	397
VARTRANSCODE 関数	397
VTRANSCODE 関数	399
VTRANSCODEX 関数	400

カテゴリ別の関数

各国語サポートに関するカテゴリ:

一部の関数は SAS 内でのみ実行され、一部の関数は SAS 内および CAS エンジン上で実行されます。関数カテゴリとして CAS を指定した場合、関数は SAS 内および CAS サーバー上で実行されます。関数カテゴリとして CAS を指定しない場合、関数は SAS 内でのみ実行されます。たとえば、BASECHAR 関数は SAS 内および CAS サーバー上で実行されるため、カテゴリとして CAS を指定します。KCVT 関数は SAS でしか実行されないため、カテゴリとして CAS を指定しません。

表 12.1 各国語サポート関連の関数のカテゴリ

カテゴリ	説明
CAS	CAS サーバー上で実行される関数。
文字	文字データを処理します。
通貨の換算	1 つの通貨を別の通貨に換算します。
DBCS	2 バイト文字セットを処理します。
日付と時間	日時データを処理します。

ロケール	指定されたロケールに基づいてデータを処理します。
変数情報	変数情報を処理します。

カテゴリ	言語要素	説明
CAS	BASECHAR 関数 (p. 312)	文字を基本文字に変換します。
	KCOMPRESS 関数 (p. 321)	指定された文字を文字式から削除します。
	KCOUNT 関数 (p. 322)	式の 2 バイト文字の数を返します。
	KINDEX 関数 (p. 324)	文字式から文字列を検索します。
	KINDEXC 関数 (p. 326)	文字式から指定した文字を検索します。
	KLEFT 関数 (p. 328)	不要な先頭の DBCS の空白と SO/SI を削除し、文字式を左詰めにします。
	KLENGTH 関数 (p. 330)	引数の長さを返します。
	KLOWCASE 関数 (p. 331)	引数に含まれるシングル幅の英語のアルファベット文字がすべて小文字に変換されます。
	KREVERSE 関数 (p. 338)	文字式を逆にします。
	KRIGHT 関数 (p. 339)	末尾の DBCS の空白と SO/SI を削除し、文字式を右詰めにします。
	KSCAN 関数 (p. 340)	指定された単語を文字式から選択します。
	KSTRCAT 関数 (p. 341)	複数の文字式を連結します。
	KSTRIP 関数 (p. 342)	文字列から先頭と末尾の空白を削除します。
	KSUBSTR 関数 (p. 344)	引数から部分文字列を抽出します。
	KTRANSLATE 関数 (p. 348)	文字式の特定の文字を置換します。
	KTRIM 関数 (p. 349)	末尾の DBCS の空白と SO/SI を文字式から削除します。
	KUPCASE 関数 (p. 351)	引数に含まれるシングル幅の英語のアルファベット文字がすべて大文字に変換されます。
	KUPDATE 関数 (p. 352)	文字値の内容を挿入、削除および置換します。
	KUPDATES 関数 (p. 356)	文字値の内容を挿入、削除および置換します。
	KVERIFY 関数 (p. 358)	式に固有の最初の文字の位置を返します。
	VTRANSCODE 関数 (p. 399)	指定された文字変数でトランスコーディングが有効かどうかを示す値を返します。

カテゴリ	言語要素	説明
	VTRANSCODEX 関数 (p. 400)	指定された引数でトランスコーディングが有効かどうかを示す値を返します。
DBCS	KCOMPARE 関数 (p. 319)	文字式の比較結果を返します。
	KCOMPRESS 関数 (p. 321)	指定された文字を文字式から削除します。
	KCOUNT 関数 (p. 322)	式の 2 バイト文字の数を返します。
	KINDEX 関数 (p. 324)	文字式から文字列を検索します。
	KINDEXB 関数 (p. 325)	文字式から文字列を検索します。
	KINDEXC 関数 (p. 326)	文字式から指定した文字を検索します。
	KINDEXCB 関数 (p. 327)	文字式から指定した文字を検索します。
	KLEFT 関数 (p. 328)	不要な先頭の DBCS の空白と SO/SI を削除し、文字式を左詰めにします。
	KLENGTH 関数 (p. 330)	引数の長さを返します。
	KLOWCASE 関数 (p. 331)	引数に含まれるシングル幅の英語のアルファベット文字がすべて小文字に変換されます。
	KPROPCASE 関数 (p. 332)	中国語、日本語、韓国語、台湾語(CJKT)文字を変換します。
	KPROPCHAR 関数 (p. 335)	特殊文字を標準文字に変換します。
	KPROPDATA 関数 (p. 336)	印刷不可文字を削除または変換します。
	KREVERSE 関数 (p. 338)	文字式を逆にします。
	KRIGHT 関数 (p. 339)	末尾の DBCS の空白と SO/SI を削除し、文字式を右詰めにします。
	KSCAN 関数 (p. 340)	指定された単語を文字式から選択します。
	KSTRCAT 関数 (p. 341)	複数の文字式を連結します。
	KSUBSTR 関数 (p. 344)	引数から部分文字列を抽出します。
	KSUBSTRN 関数 (p. 345)	部分文字列を返します。長さがゼロの結果も返せます。
	KSUBSTRB 関数 (p. 346)	引数内の部分文字列のバイト位置に従って、引数から部分文字列を抽出します。
KTRANSLATE 関数 (p. 348)	文字式の特定の文字を置換します。	
KTRIM 関数 (p. 349)	末尾の DBCS の空白と SO/SI を文字式から削除します。	
KTRUNCATE 関数 (p. 350)	マルチバイト文字を壊すことなく、文字列をバイト単位で、指定した長さに切ります。	

カテゴリ	言語要素	説明
	KUPCASE 関数 (p. 351)	引数に含まれるシングル幅の英語のアルファベット文字がすべて大文字に変換されます。
	KUPDATE 関数 (p. 352)	文字値の内容を挿入、削除および置換します。
	KUPDATEB 関数 (p. 354)	引数内の文字値のバイト位置に従って、文字値の内容を挿入、削除および置換します。
	KUPDATES 関数 (p. 356)	文字値の内容を挿入、削除および置換します。
	KVERIFY 関数 (p. 358)	式に固有の最初の文字の位置を返します。
	KVERIFYB 関数 (p. 359)	式に固有の最初の文字の位置を返します。
エンコーディング	ENCODCOMPAT 関数 (p. 313)	2つのエンコーディング間のトランスコーディング互換性を確認します。
	ENCODISVALID 関数 (p. 314)	有効なエンコーディング名であるかどうかを検証します。
日付と時間	NLDATE 関数 (p. 360)	SAS 日時値を、日付のディスクリプタを使用して指定したロケールの日付値に変換します。
	NLDATM 関数 (p. 363)	SAS 日時値を、日時形式のディスクリプタを使用して指定したロケールの時間値に変換します。
	NLTIME 関数 (p. 366)	SAS 時間値または SAS 日時値を、NLTIME ディスクリプタを使用して指定したロケールの時間値に変換します。
	TZONEID 関数 (p. 382)	現在のタイムゾーン ID を返します。
	TZONENAME 関数 (p. 384)	現在の標準時間または夏時間のタイムゾーン名を返します。
	TZONEOFF 関数 (p. 385)	ユーザーのタイムゾーンオフセットを返します。
	TZONES2U 関数 (p. 386)	SAS 日時値を UTC 日時値へ変換します。
	TZONEDSTNAME 関数 (p. 387)	夏時間名を返します。
	TZONEDSTOFF 関数 (p. 388)	指定された夏時間のタイムゾーンオフセット値を返します。
	TZONESTTNAME 関数 (p. 389)	標準のタイムゾーン名を返します。
	TZONESTTOFF 関数 (p. 390)	指定された標準時間のタイムゾーンオフセット値を返します。
	TZONEU2S 関数 (p. 391)	UTC 日時値を SAS 日時値へ変換します。
変数情報	VARTRANSCODE 関数 (p. 397)	SAS データセット変数のトランスコーディング属性を返します。

カテゴリ	言語要素	説明
	VTRANSCODE 関数 (p. 399)	指定された文字変数でトランスコーディングが有効かどうかを示す値を返します。
	VTRANSCODEX 関数 (p. 400)	指定された引数でトランスコーディングが有効かどうかを示す値を返します。
文字	ANORM420 関数 (p. 307)	EBCDIC420 でエンコードされた入力文字列から正規化された文字列を返します。
	BASECHAR 関数 (p. 312)	文字を基本文字に変換します。
	KCVT 関数 (p. 322)	データのある種類のエンコーディングデータから別の種類のエンコーディングデータに変換します。
	KSTRIP 関数 (p. 342)	文字列から先頭と末尾の空白を削除します。
	TRANTAB 関数 (p. 392)	指定された変換テーブルを使用してデータをトランスコードします。
	UNICODE 関数 (p. 393)	Unicode 文字を現在の SAS セッションエンコーディングに変換します。
	UNICODEC 関数 (p. 394)	現在の SAS セッションエンコーディングの文字を Unicode 文字に変換します。
	UNICODELEN 関数 (p. 396)	Unicode データの文字単位の長さを示します。
	UNICODEWIDTH 関数 (p. 397)	Unicode データの表示単位の長さを示します。
ロケール	GETLOCENV 関数 (p. 315)	現在のロケール/言語環境を返します。
	GETPXLANGUAGE 関数 (p. 316)	現在の 2 文字の言語コードを返します。
	GETPXLOCALE 関数 (p. 317)	SAS ロケールの POSIX ロケール値を返します。
	GETPXREGION 関数 (p. 318)	現在の 2 文字の地域コードを返します。
	SASMSG 関数 (p. 367)	データセットからメッセージを返します。返されるメッセージは現在のロケールと指定されたキーに基づきます。
	SASMSG 関数 (p. 370)	データセットからメッセージを返します。メッセージは、指定されたロケール値と指定されたキー値に基づきます。
	SORTKEY 関数 (p. 372)	言語並べ替えキーを作成します。
	SETLOCALE 関数 (p. 375)	現在の SAS ロケールにロケールキーを指定します。

ディクショナリ

ANORM420 関数

EBCDIC420 でエンコードされた入力文字列から正規化された文字列を返します。

カテゴリ: 文字

制限事項: この関数には I18N レベル 1 のステータスが付与されています。英語以外の言語を使用する場合、可能な限り I18N レベル 1 の関数の使用は避けてください。特定の環境下では、I18N レベル 1 の関数は、2 バイト文字セット(DBCS)または複数バイト文字セット(MBCS)エンコーディングを使用すると正常に動作しない場合があります。詳細については、[国際化の互換性 \(279 ページ\)](#) を参照してください。

構文

ANORM420 (*string*, <*modifiers*>)

必須引数

string

入力文字列が EBCDIC420 でエンコードされるように指定します。

オプション引数

modifiers

1 つ以上の修飾子を指定する文字定数、変数、または式。次の修飾子は大文字でも小文字でも指定できます。

- g 合字を無視します
- i アラビアインド数字を無視します
- p 形状文字のマッピングを無視します
- s 空白の追加を無視します
- t トランスコーティングを無視します
- z ゼロ幅空白文字のマッピングを無視します

詳細

ANORM420 関数は、EBCDIC420 でエンコードされたデータを処理します。

修飾子の変数については、次の表を参照してください。

ANORM420 関数は、修飾子 *g* が指定されていない場合は、次の文字とコードポイントを適切な合字に置き換えます。

表 12.2 修飾子 *g*: アラビア語合字

文字	ユニコード値	開始	終了	説明
ﻯ	(U+FEF5)	0xB2	0xB1 + 0x47	アラビア合字 Lam-Aleph(上部 Mada)独立形
ﻰ	(U+FEF6)	0xB3	0xB1 + 0x47	アラビア合字 Lam-Aleph(上部 Mada)語末形
ﻱ	(U+FEF7)	0xB4	0xB1 + 0x49	アラビア合字 Lam-Aleph(上部 Hamza)独立形
ﻲ	(U+FEF8)	0xB5	0xB1 + 0x49	アラビア合字 Lam-Aleph(上部 Hamza)語末形
ﻻ	(U+FEFC)	0xB8	0xB1 + 0x56	アラビア合字 Lam-Aleph 独立形
ﻼ	(U+FEFD)	0xB9	0xB1 + 0x56	アラビア合字 Lam-Aleph 語末形

修飾子 *i* を指定せず、トランスコーディングが発生しない場合、ANORM420 はアラビアインド数字を数字に再マッピングします。

表 12.3 修飾子 *i*: アラビアインド数字

開始	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
	U+0660	U+0661	U+0662	U+0663	U+0664	U+0665	U+0666	U+0667	U+0668	U+0669
16進数	0xDF	0xEA	0xEB	0xED	0xEE	0xEF	0xFB	0xFC	0xFD	0xFE
終了	0	1	2	3	4	5	6	7	8	9
16進数	0xF0	0xF1	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7	0xF8	0xF9

修飾子 *p* を指定しない場合、ANORM420 関数は形状文字を対応する非形状文字に再マッピングします。

表 12.4 修飾子 *p*: 形状文字のマッピング

開始	終了
0x43	U+FE7D 0x42 U+0651
0x48	FE82 0x47 U+0622
0x51	U+FE84 0x49 U+0623

開始		終了	
0x57	U+FE8E	0x56	U+0627
0x59	U+FE91	0x58	U+0628
0x64	U+FE97	0x63	U+062A
0x66	U+FE9B	0x65	U+062B
0x68	U+FE9F	0x67	U+062C
0x70	U+FEA3	0x69	U+062D
0x72	U+FEA7	0x71	U+062E
0x78	U+FEB3	0x77	U+0633
0x8A	U+FEB7	0x80	U+0634
0x8C	U+FEBB	0x8B	U+0635
0x8E	U+FEBF	0x8D	U+0636
0x9B	U+FECA	0x9A	U+0639
0x9C	U+FECB	0x9A	U+0639
0x9D	U+FECC	0x9A	U+0639
0x9F	U+FECE	0x9E	U+063A
0xA0	U+FECF	0x9E	U+063A
0xAA	U+FED0	0x9E	U+063A
0xAC	U+FED3	0xAB	U+0641
0xAE	U+FED7	0xAD	U+0642
0xB0	U+FEDB	0xAF	U+0643
0xBA	U+FEDF	0xB1	U+0644
0xBC	U+FEE3	0xBB	U+0645
0xBE	U+FEE7	0xBD	U+0646
0xCB	U+FEEB	0xBF	U+0647
0xCD	U+FEEC	0xBF	U+0647

開始		終了	
0xDB	U+FEF0	0xDA	U+061F
0xDD	U+FEF2	0xDC	U+064A
0xDE	U+FEF3	0xDC	U+064A

修飾子 *s* を指定していない場合、ANORM420 関数は次の文字の後に空白を追加します。

表 12.5 修飾子 *s*: 空白の追加を無視

ب	0x58	(U+0628)	アラビア文字 BEH
ة	0x62	(U+0629)	アラビア文字 TEH MARBUTA
ت	0x63	(U+062A)	アラビア文字 TEH
ث	0x65	(U+062B)	アラビア文字 THEH
ج	0x67	(U+062C)	アラビア文字 JEEM
ح	0x69	(U+062D)	アラビア文字 HAH
خ	0x71	(U+062E)	アラビア文字 KHAH
س	0x77	(U+0633)	アラビア文字 SEEN
ش	0x80	(U+0634)	アラビア文字 SHEEN
ص	0x8B	(U+0635)	アラビア文字 SAD
ض	0x8D	(U+0636)	アラビア文字 DAD
ع	0x9A	(U+0639)	アラビア文字 AIN
ع	0x9B	(U+FECA)	アラビア文字 AIN 語未形
غ	0x9E	(U+063A)	アラビア文字 GHAIN
غ	0x9F	(U+FECE)	アラビア文字 GHAIN 語未形
ف	0xAB	(U+0641)	アラビア文字 FEH

ق	0xAD	(U+0642)	アラビア文字 QAF
ك	0xAF	(U+0643)	アラビア文字 KAF
ل	0xB1	(U+0644)	アラビア文字 LAM
م	0xBB	(U+0645)	アラビア文字 MEEM
ن	0xBD	(U+0646)	アラビア文字 NOON
ه	0xBF	(U+0647)	アラビア文字 HEH
ا	0xDA	(U+0649)	アラビア文字 ALEF MAKSURA
ي	0xDC	(U+064A)	アラビア文字 YEH
آ	0xDB	(U+FEF0)	アラビア文字 ALEF MAKSURA 語未形
آ	0xDD	(U+FEF2)	アラビア文字 YEH 語未形

修飾子 *t* を指定していない場合、ANORM420 関数は正規化された文字列をセッションエンコーディングにトランスコーディングします。

修飾子 *z* を指定しない場合は、ANORM420 関数は ZWSP と呼ばれるゼロ幅空白 (U+200B) を再マッピングします。ゼロ幅文字は、コード位置 0x45 に配置されています。

例

次の例は、ANORM420 関数について説明しています。

ステートメント	結果
<pre>data_null_ ; a = '59CD57BC577745'x ; s1 = anorm420(a) ; /* Turn off addition of space and mapping of Arabic-Indic numbers */ s2 = anorm420(a,"si") ; /* Turn off transcoding */ s3 = anorm420(a,'t') ; put s1= \$hex20. / s2= \$hex20. / s3=\$hex20.; run;</pre>	<pre>s1=C8E5C7E3C7D320A02020 s2=C8E5C7E3C7D3A0202020 s3=59CD57BC577740454040</pre>

BASECHAR 関数

文字を基本文字に変換します。

カテゴリ: 文字
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

STR=BASECHAR(*<instr>*(,*<Unicode type>*))

必須引数

str
変換されたデータ文字列。

instr
入力データ文字列。

Unicode type
次のいずれかの Unicode 文字形式を指定した場合、各国語文字が指定した形式で表記されます。

ESC Unicode エスケープ(\u00cd)。

NCR 数値文字表現(Å)。

PAREN Unicode カッコエスケープ(<u00c5>)。

詳細

BASECHAR 関数は、文字を読み込み、基本文字に変換します。基本文字ならびに 1 つ以上のアクセントまたは結合文字から成る文字もあります。BASECHAR 関数は、文字列の文字を読み込み、Unicode 指定に基づいてアクセントなしの基本文字形式に変換します。Unicode タイプを指定した場合、各国語文字が指定した形式で表記されます。たとえば、データ文字列 Mühlenfließ は Mühlenfließに変換されます。

例

次の例で Unicode タイプの使用法を説明します。

ステートメント	結果
<p>注: SAS セッションエンコーディングは UTF-8 です。</p> <pre>data cities;length name\$24 name_utf8\$24; input name; name_utf8 = basechar(name); put name_utf8=; cards; Mühlenfließ LaUña ZielonaGóra run;</pre>	<p>注: BASECHAR 関数は、Unicode 文字分解マッピングを使用するので、シャープ s (ß)が有効です。http://www.unicode.org/reports/tr44/#Character_Decomposition_Mappings</p> <pre>Muhlenfließ LaUna ZielonaGora</pre>
<p>注: SAS セッションエンコーディングは UTF-8 で、ESC オプションを指定します。</p> <pre>data cities;length name\$24 name_utf8\$80; input name; name_utf8 = basechar(name, "ESC"); put name_utf8=; cards; Mühlenfließ LaUña ZielonaGóra run;</pre>	<pre>Muhlenfließ\u00df LaUna ZielonaGora</pre>
<p>注: SAS セッションエンコーディングは UTF-8 で、NCR オプションを指定します。</p> <pre>data cities;length name\$24 name_utf8\$80; input name; name_utf8 = basechar(name, "NCR"); put name_utf8=; cards; Mühlenfließ LaUña ZielonaGóra run;</pre>	<pre>Muhlenfließ&#x00df LaUna ZielonaGora</pre>

ENCODCOMPAT 関数

2つのエンコーディング間のトランスコーディング互換性を確認します。

カテゴリ: エンコーディング

構文

ENCODCOMPAT(*source1*,<*source2*>)

必須引数

source1

エンコーディングを表す文字列です。

source2

エンコーディングを表す文字列です。この引数はオプションです。

詳細

エンコーディングを1つ指定した場合、この関数は現在の SAS セッションのエンコーディングとの互換性を確認します。

エンコーディングを2つ指定した場合、この関数は2つのエンコーディング間の互換性を確認します。

この関数は2つのエンコーディング識別子を比較し、データをトランスコードする必要があるかどうかを評価します。*Source1* はソースのエンコーディングです。*Source2* はターゲットのエンコーディングです。7ビット ASCII を他のタイプの ASCII にトランスコーディングする場合の互換性はありますが、ASCII を7ビット ASCII にトランスコーディングする場合は、互換性がないことがあります。

ENCODCOMPAT 関数は次の値を示します。

- 1 *Source1* は有効なエンコーディング名ではありません。
- 2 *Source2* は有効なエンコーディング名ではありません。
- 0 エンコーディング間に互換性はありません。トランスコーディングが必要です。
- 1 エンコーディング間の互換性があります。トランスコーディングは不要です。
- 2 改行文字が検出されました。

例

次の例で ENCODCOMPAT の機能を説明します。

ステートメント	結果
<pre>/* session encoding is wlatin1 */ isCompat= EncodCompat("xyz"); put isCompat;</pre>	-1
<pre>/* session encoding is wlatin1 */ isCompat= EncodCompat ("cp1252"); put isCompat;</pre>	1
<pre>isCompat= EncodCompat ("ebcdic1149","open_ed-1149"); put isCompat;</pre>	2
<pre>isCompat= EncodCompat ("cp1251","ebcdic1149"); put isCompat;</pre>	0

ENCODISVALID 関数

有効なエンコーディング名であるかどうかを検証します。

カテゴリ: エンコーディング

構文

ENCODISVALID(*source*)

必須引数

source

エンコーディング名を表す文字列です。

詳細

ENCODISVALID 関数は次の値を返します。

- 0 文字列は有効なエンコーディング名ではありません。
- 1 文字列は有効な短いエンコーディング名です。
- 2 文字列は有効な長いエンコーディング名です。
- 3 文字列は有効なエイリアスエンコーディング名です。

例

次の例で ENCODISVALID の機能を説明します。

SAS ステートメント	結果
isValid= EncodIsValid("xyz"); put isValid;	0
isValid= EncodIsValid("wlt2"); put isValid;	1
isValid= EncodIsValid("wlatin2"); put isValid;	2
isValid= EncodIsValid("cp1250"); put isValid;	3

GETLOCENV 関数

現在のロケール/言語環境を返します。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

GETLOCENV()

詳細

GETLOCENV 関数は、有効な SAS ロケールのロケール/言語環境値を返します。次の環境値が表示されます。

SBCS

SAS セッションエンコーディングは SBCS (1 バイト文字セット) です。文字列の操作のため、SASWZSD が読み込まれます。

DBCS

SAS セッションエンコーディングは DBCS (2 バイト文字セット) です。文字列の操作のため、SASWZSD が読み込まれます。

MBCS

SAS セッションエンコーディングは Unicode (UTF8) です。文字列の操作のため、SASWZSU が読み込まれます。

空白値が返された場合は、WZSS サブシステムが使用できません。このアクションは、構成またはインストールでエラーが発生したことを示します。

例

次の例では、LOCALE=システムオプションが French_France に設定されています。

ステートメント	結果
option locale=french_france; environ=getlocenv(); put environ;	SBCS

GETPXLANGUAGE 関数

現在の 2 文字の言語コードを返します。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

GETPXLANGUAGE()

詳細

GETPXLANGUAGE 関数は、LOCALE=SAS システムオプションの現在の値に基づいて、2 文字の言語コードを返します。言語名の長さは 2 文字です。値を受け取る変数のサイズが 2 文字より小さい場合、値は切り捨てられます。

例

最初の例では、LOCALE=システムオプションが French_France に設定されています。2 番目の例では German に設定されています。3 番目の例では English_United States に設定されています。

ステートメント	結果
option locale=french_france; lang=getpxLanguage(); put lang;	fr
option locale=German; lang=getpxLanguage(); put lang;	de
option locale=en_US; lang=getpxLanguage(); put lang;	ja

関連項目:

システムオプション:

- [“LOCALE システムオプション” \(550 ページ\)](#)

関数:

- [“GETPXREGION 関数” \(318 ページ\)](#)
- [“GETPXLOCALE 関数” \(317 ページ\)](#)

GETPXLOCALE 関数

SAS ロケールの POSIX ロケール値を返します。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

GETPXLOCALE(*<source>*)

必須引数

<source>

ロケール名を指定する省略可能な引数です。

詳細

GETPXLOCALE 関数は、有効な SAS ロケール名に対する POSIX ロケール値を返します。無効なロケール名を指定すると、ヌル文字列が返されます。<source>引数に値を指定しない場合、関数は現在の SAS セッションの POSIX 名を返します。POSIX ロケール名の長さは 5 文字です。値を受け取る変数のサイズが 5 文字より小さい場合、値は切り捨てられます。

例

最初の例では、LOCALE=システムオプションが French_France に設定されています。2 番目の例では、<source>引数が German_Germany に設定されています。3 番目の例では、<source>引数が English_United States に設定されています。

ステートメント	結果
option locale=french_france; locale=getpxLocale(); put locale;	fr_FR
locale=getpxLocale("german_germany"); put locale;	de_DE
locale=getpxLocale("english_unitedstates"); put locale;	en_US

関連項目:

システムオプション:

- [“LOCALE システムオプション” \(550 ページ\)](#)

関数:

- [“GETPXLANGUAGE 関数” \(316 ページ\)](#)
- [“GETPXREGION 関数” \(318 ページ\)](#)

GETPXREGION 関数

現在の 2 文字の地域コードを返します。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

GETPXREGION()

詳細

GETPXREGION 関数は、現在の LOCALE=SAS システムオプションに基づいて、2文字の地域コードを返します。地域名の長さは2文字です。値を受け取る変数のサイズが2文字より小さい場合、値は切り捨てられます。

例

最初の例では、LOCALE=システムオプションが French_France に設定されています。2番目の例では German に設定されています。3番目の例では English_United States に設定されています。

ステートメント	結果
<pre>option locale=french_france; region=getpxRegion(); put region;</pre>	FR
<pre>option locale=German; region=getpxRegion(); put region;</pre>	DE
<pre>option locale=en_US; region=getpxRegion(); put region;</pre>	US

関連項目:

システムオプション:

- [“LOCALE システムオプション” \(550 ページ\)](#)

関数:

- [“GETPXLOCALE 関数” \(317 ページ\)](#)
- [“GETPXLANGUAGE 関数” \(316 ページ\)](#)

KCOMPARE 関数

文字式の比較結果を返します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

ヒント: 非 DBCS で相当する関数は、“[COMPARE Function](#)” (*SAS Viya Functions and CALL Routines: Reference*)です。

構文

KCOMPARE(*source*, <*pos*, <*count*,> > *findstr*)

必須引数

source

比較される文字式を指定します。

pos

比較を開始する *source* の開始位置を指定します。 *pos* を省略した場合、*source* 全体が比較されます。 *pos* が 0 よりも小さい場合、*source* は SO/SI 文字を含まない拡張 DBCS データとして判断されます。

count

比較するバイト数を指定します。 *count* を省略した場合、末尾の空白を除き、*source* の後ろのすべての *pos* が比較されます。

findstr

source と比較する文字式を指定します。

詳細

KCOMPARE では、値を次のように返します。

- *source* が *findstr* より小さい場合は負の値
- *source* が *findstr* と等しい場合は 0
- *source* が *findstr* より大きい場合は正の値

例

次の例では、日本語文字を使用します。

ステートメント	結果					
options pageno=1 nodate ls=80 ps=60;	The SAS System					
data test;	Obs	rc1	rc2	rc3	rc4	rc5
rc1 = kcompare('漢字のテスト','漢字');						
rc2 = kcompare('漢字のテスト',7,'テスト');	1	3	0	3	1	4
rc3 = kcompare('漢字のテスト',1,6,'漢字');						
rc4 = kcompare('漢字のテスト','ABC');						
rc5 = kcompare('ABCDEF','ABC');						
run;						
proc print data=test; run;						

KCOMPRESS 関数

指定された文字を文字式から削除します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[COMPRESS](#) で、[SAS Viya 関数と CALL ルーチン: リファレンス](#)を参照してください。

構文

KCOMPRESS(*source*,<*characters-to-remove*>)

必須引数

source

削除する文字を含む文字式を指定します。KCOMPRESS は、*source* のみが指定されると、すべての 1 バイトおよび 2 バイトの空白を削除してこの式を返します。

characters-to-remove

KCOMPRESS が文字式から削除する文字を指定します。characters-to-remove を省略すると、KCOMPRESS はすべての空白を削除します。

ヒント 文字のリテラル文字列を引用符で囲みます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; result=kcompress('漢字のテスト','の'); put result=; run;</pre>	result=漢字テスト

関連項目:

関数:

- ["KLEFT 関数" \(328 ページ\)](#)
- ["KTRIM 関数" \(349 ページ\)](#)

KCOUNT 関数

式の 2 バイト文字の数を返します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

構文

KCOUNT(*source*)

必須引数

source
数える文字式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; text='abcあいう'; result=kcount(text); put result=; run;</pre>	<pre>result=3</pre>

KCVT 関数

データのある種類のエンコーディングデータから別の種類のエンコーディングデータに変換します。

カテゴリ: 文字

制限事項: この関数には I18N レベル 1 のステータスが付与されています。英語以外の言語を使用する場合、可能な限り I18N レベル 1 の関数の使用は避けてください。特定の環境下では、I18N レベル 1 の関数は、2 バイト文字セット(DBCS)または複数バイト文

字セット(MBCS)エンコーディングを使用すると正常に動作しない場合があります。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

KCVT(*text*, *intype*, *outtype*, <*options*,...>)

必須引数

text

変換する文字変数を指定します。

intype

データのエンコーディングを指定します。テキストのエンコーディングは、入力データのエンコーディングと一致する必要があります。有効な値については、“[データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値](#)” (613 ページ)を参照してください。

ASCIIANY および EBCDICANY は、無効なエンコーディング値です。

outtype

文字データに変換するエンコーディングを指定します。有効な値については、“[データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値](#)” (613 ページ)を参照してください。

ASCIIANY および EBCDICANY は、無効なエンコーディング値です。

options

文字データオプションを指定します。次のオプションを使用できます。

NOSOSI NOSHIFT	シフトコードと半角文字は含めません。
INPLACE	文字データを変換によって置き換えます。INPLACE オプションは、異なるホスト間で文字データの長さが異なる場合に、同じ位置を確保するために指定します。たとえば、INPLACE オプションは、データをシフトコードが必要なホストから、シフトコードが不要な別のホストに変換します。シフトコードに関して <i>outtype</i> に変換される文字データの長さが、 <i>intype</i> に指定された長さよりも長い場合には、切り捨てが行われます。
KANA	文字データの列に、半角カタカナ文字を含めます。
UPCASE	2 バイトの英字を大文字に変換します。
LOWCASE	2 バイトの英字を小文字に変換します。
KATA2HIRA	カタカナデータをひらがなに変換します。
HIRA2KATA	ひらがなデータをカタカナに変換します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KCVT 関数は、SBCS、DBCS および MBCS 文字列をエンコーディングデータに変換します。たとえば、KCVT 関数では、ASCII コードデータから UCS2 エンコーディングデータ、ギリシャ語コードデータから UTF-8、日本語 SJIS コードデータから別の日本語コードデータの変換が可能です。指定可能な INTYPE および

OUTTYPE オプションの種類は、UCS2、UCS2L、UCS2B、および UTF8 です。DBCS モードを有効にするには、構成ファイル内またはコマンドラインで、次の DBCS オプションを指定します。

KCVT 関数が事前に長さを割り当てられていない変数に値を返す場合、デフォルトで変数の長さは 200 に割り当てられます。

例

次のコードは、*my-input-file* として指定された外部テキストファイルについて IBM PC コードを DEC コードに変換し、OUTDD に結果を書き出します。

```
data _null_;
  infile 'my-input-file';
  file outdd noprint;
  input @1 text $char80.;
  text = kcvf(text, 'pcibm', 'dec');
  put @1 text $char80.;
run;
```

KINDEX 関数

文字式から文字列を検索します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[INDEX](#) で *SAS Viya 関数* と *CALL ルーチン: リファレンス* を参照してください。

構文

KINDEX(*source*, *excerpt*)

必須引数

source

検索する文字式を指定します。

excerpt

文字式で検索する文字列を指定します。

ヒント 文字のリテラル文字列を引用符で囲みます。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ) を参照してください。

KINDEX 関数は、*source* 内を左から右へ、*excerpt* で指定された文字列が最初に現れる個所を検索し、*source* 内でのその文字列の先頭文字の位置を返します。*source* 内に文字列が見つからない場合、KINDEX は値 0 を返します。文字列が複数回現れる場合、KINDEX は、最初の出現個所の位置のみを返します。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<code>data _null_;</code>	<code>result=7</code>
<code>text='漢字文字列の検索';</code>	<code>result=0</code>
<code>result=kindex(text,'検索');</code>	
<code>put result=;</code>	
<code>text2='探す';</code>	
<code>result=kindex(text,text2);</code>	
<code>put result=;</code>	
<code>run;</code>	

関連項目:

関数:

- [“KINDEXC 関数” \(326 ページ\)](#)

KINDEXB 関数

文字式から文字列を検索します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

KINDEXB(*source*, *excerpt*)

必須引数

source

検索する文字式を指定します。

excerpt

文字式で検索する文字列を指定します。

ヒント 文字のリテラル文字列を引用符で囲みます。

詳細

KINDEXB 関数は、*source* を左から右へ *excerpt* に指定された文字列の最初の出現個所を検索し、その文字列の先頭文字の *source* 内での位置を返します。*source* 内に文字列が見つからない場合、KINDEXB は値 0 を返します。文字列が複数回現れる場合、KINDEXB は、最初の出現個所の位置のみを返します。

比較

KINDEXB は、バイトベースの値を返します。KINDEX は、文字ベースの値を返します。ほとんどのラテン文字は 2 バイト幅で、ほとんどの CJK 文字は 3 バイト幅です。そのため、これらの関数は異なる結果を返します。

KINDEXCB 関数は、文字列中に存在するあらゆる文字の、最初の出現個所を検索します。KINDEXB 関数は、文字列の、1 つのパターンとしての最初の出現個所を検索します。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null; text='漢字文字列の検索'; result=kindexb(text, '検索'); put result=; text2='探す'; result=kindexb(text,text2); put result=; run;</pre>	<pre>Result=19 Result=0</pre>

KINDEXC 関数

文字式から指定した文字を検索します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、“[INDEXC Function](#)” (*SAS Viya Functions and CALL Routines: Reference*)です。

構文

KINDEXC(*source*, *excerpt-1*, ...*excerpt-n*)

必須引数

source

検索する文字式を指定します。

excerpt-1, ...excerpt-n

文字式で検索する文字を指定します。

ヒント 複数の *excerpt* を指定する場合は、カンマで区切ります。

文字のリテラル文字列を引用符で囲みます。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ) を参照してください。

KINDEXC 関数は、*source* を左から右へ、*excerpt* 内に含まれるいずれかの文字が最初に現れる個所を検索し、*source* 内でのその文字の位置を返します。*excerpt-1* から *excerpt-n* までの文字がいずれも *source* 内で見つからなかった場合、KINDEXC は値 0 を返します。

比較

KINDEXC 関数は、文字列に含まれる個々の文字が最初に現れる個所を検索するのに対し、KINDEX 関数は、文字列がパターンとして最初に現れる個所を検索します。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; a='A B C.def (漢字123456)'; result=kindexc(a,'1234567890','感漢'); put result=; run;</pre>	<pre>result=9</pre>

関連項目:

関数:

- “[KINDEX 関数](#)” (324 ページ)

KINDEXCB 関数

文字式から指定した文字を検索します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

KINDEXCB(*source*, *excerpt-1*, ...*excerpt-n*)

必須引数

source

検索する文字式を指定します。

excerpt-1, ...excerpt-n

文字式で検索する文字を指定します。

ヒント 複数の excerpt を指定する場合は、カンマで区切ります。

文字のリテラル文字列を引用符で囲みます。

詳細

TKINDEXCB 関数は、*source* 内のデータを左から右へ、*excerpt* の引数内のあらゆる文字の、最初の出現個所を検索し、その文字の *source* 内での位置を返します。*excerpt-1* から *excerpt-n* までの文字がいずれも *source* 内で見つからなかった場合、KINDEXC は値 0 を返します。

比較

KINDEXCB は、バイトベースの値を返します。KINDEXC は、文字ベースの値を返します。ほとんどのラテン文字は 2 バイト幅で、ほとんどの CJK 文字は 3 バイト幅です。そのため、これらの関数は異なる結果を返します。

KINDEXCB 関数は、文字列中に存在するあらゆる文字の、最初の出現個所を検索するのに対し、KINDEXB 関数は、文字列の、1 つのパターンとしての最初の出現個所を検索します。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; a='A B C.def (漢字123456)'; result=kindexcb(a, '1234567890', '感漢'); put result=; run;</pre>	Result=19

KLEFT 関数

不要な先頭の DBCS の空白と SO/SI を削除し、文字式を左詰めにします。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[LEFT](#) で、[SAS Viya 関数と CALL ルーチン: リファレンス](#)を参照してください。

構文

KLEFT(*argument*)

必須引数

argument

SAS 文字式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KLEFT は、引数を返して先頭の空白を削除します。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; a=' 漢字 の テスト'; b=kleft(a); put '----+----1----+----2----+'; put a \$25.; put b \$25.; run;</pre>	<pre>----+----1----+----2----+ 漢字 の テスト 漢字 の テスト</pre>

関連項目:

関数:

- [“KCOMPRESS 関数” \(321 ページ\)](#)
- [“KRIGHT 関数” \(339 ページ\)](#)
- [“KTRIM 関数” \(349 ページ\)](#)

KLENGTH 関数

引数の長さを返します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[LENGTH](#) で、*SAS Viya 関数と CALL ルーチン: リファレンス*を参照してください。

構文

KLENGTH(*argument*)

必須引数

argument

SAS 式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KLENGTH 関数は、引数内で最も右にある空白以外の文字の位置を表す整数を返します。引数の値がない場合、KLENGTH は値 0 を返します。引数が初期化されていない数値変数の場合、KLENGTH は値 12 を返し、SAS ログに数値が文字値に変換されたという注意を出力します。

SBCS EnglishVARCHAR データを使っている場合、処理性能に問題が生じる可能性があります。性能向上のため、K 関数でない関数を用いた、次のようなコードにしてください。

性能が向上し、その結果は KLENGTH 関数と同じになります。

```
len = LENGTHC(TRIMN(x));
```

性能が向上し、その結果は LENGTH 関数と同じになります。

```
len = LENGTHC(TRIMN(x));  
if NOT len then len = 1;
```

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; text='abcあいう'; result=klength(text); put result=; run;</pre>	<pre>result=6</pre>

KLOWCASE 関数

引数に含まれるシングル幅の英語のアルファベット文字がすべて小文字に変換されます。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[LOWCASE](#) で、[SAS Viya 関数と CALL ルーチン: リファレンス](#)を参照してください。

構文

KLOWCASE(*argument*)

必須引数

argument

SAS 文字式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KLOWCASE 関数により、文字引数がコピーされ、大文字のシングル幅の英語アルファベット文字がすべて小文字に変換され、変更された値が結果として返されます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; result=klowcase('ABCあいうえお'); put result=; run;</pre>	<pre>result=abcあいうえお</pre>

KPROPCASE 関数

中国語、日本語、韓国語、台湾語(CJKT)文字を変換します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

`str=KPROPCASE(<instr> , (<options>))`

必須引数

str

変換された、現在の SAS セッションエンコーディングのデータ文字列。

instr

入力データ文字列。

options

指定されたオプションに基づいて、日本語、中国語、韓国語、台湾語文字を変換します。

HALF-KATAKANA, FULL-KATAKANA

半角カタカナを全角カタカナに変換します。このオプションは、日本語エンコーディングでのみ使用します。

制限事項 このオプションは、FULL-KATAKANA, HALF-KATAKANA オプションと同時に使用できません。

FULL-KATAKANA, HALF-KATAKANA

全角カタカナを半角カタカナに変換します。このオプションは、日本語エンコーディングでのみ使用します。

制限事項 このオプションは、HALF-KATAKANA, FULL-KATAKANA オプションと同時に使用できません。

KATAKANA, ROMAJI

カタカナ文字列をローマ字文字列に変換します。このオプションは、日本語エンコーディングでのみ使用します。

制限事項 このオプションは、ROMAJI, KATAKANA オプションと同時に使用できません。

ROMAJI, KATAKANA

ローマ字文字列をカタカナ文字列に変換します。このオプションは、日本語エンコーディングでのみ使用します。

制限事項 このオプションは、KATAKANA, ROMAJI オプションと同時に使用できません。

FULL-ALPHABET, HALF-ALPHABET

全角アルファベット文字を半角アルファベット文字に変換します。このオプションは、日本語、中国語、韓国語、台湾語エンコーディングでのみ使用します。

制限事項 このオプションは、HALF-ALPHABET, FULL-ALPHABET オプションと同時に使用できません。

HALF-ALPHABET, FULL-ALPHABET

半角アルファベット文字を全角アルファベット文字に変換します。このオプションは、日本語、中国語、韓国語、台湾語エンコーディングでのみ使用します。

制限事項 このオプションは、FULL-ALPHABET, HALF-ALPHABET オプションと同時に使用できません。

LOWERCASE, UPPERCASE

小文字アルファベット文字を大文字アルファベット文字に変換します。

制限事項 このオプションは、UPPERCASE, LOWERCASE オプションと同時に使用できません。

UPPERCASE, LOWERCASE

大文字アルファベット文字を小文字アルファベット文字に変換します。

制限事項 このオプションは、LOWERCASE, UPPERCASE オプションと同時に使用できません。

PROPER

エンコーディングに基づいて、次のデフォルトオプションを指定します。

- 日本語エンコーディング:
- HALF-KATAKANA, FULL-KATAKANA
- FULL-ALPHABET, HALF-ALPHABET
- LOWERCASE, UPPERCASE
- 韓国語エンコーディング:
- FULL-ALPHABET, HALF-ALPHABET
- 中国語エンコーディング:
- FULL-ALPHABET, HALF-ALPHABET
- 台湾語エンコーディング:
- FULL-ALPHABET, HALF-ALPHABET

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ) を参照してください。

指定されたオプションとデフォルトオプションに基づいて、入力文字列を変換します。KPROPCASE 関数は、中国語、日本語、韓国語、台湾語(CJKT)環境をサポートします。

例

次に、KPROPCASE 関数の使用例を示します。

```
length fullkana halfkana upper lower fullalpha $ 200;
length str1 str2 str3 str4 str5 str7 str8 $ 30 str6 $44;
lower = 'do-naxtutsu'; /* Doughnuts in Japanese Roman word. */
upper = 'DO-NAXTUTSU'; /* Doughnuts in Japanese Roman word. */
fullkana = unicode('\u30C9\u30FC\u30CA\u30C3\u30C4');
halfkana = unicode('\uFF84\uFF9E\uFF70\uFF85\uFF6F\uFF82');
fullalpha = unicode('\uFF24\uFF2F\uFF0D\uFF2E\uFF21\uFF38\uFF34\uFF35\uFF3A\uFF33\uFF35');
str1 = kpropcase(fullkana, 'full-katakana, half-katakana');
if (halfkana EQ trim(str1)) then
  put str1= $hex14.;
str2 = kpropcase(halfkana, 'half-katakana, full-katakana');
if (fullkana EQ trim(str2)) then
  put str2= $hex22.;
str3 = kpropcase(fullkana, 'katakana, romaji');
if (trim(str3) EQ upper) then
  put str3= ;
str4 = kpropcase(upper, 'romaji, katakana');
if (trim(str4) EQ fullkana) then
  put str4= $hex22.;
str5 = kpropcase(fullalpha, 'full-alphabet, half-alphabet');
if (trim(upper) EQ str5) then
  put str5=;
str6 = kpropcase(upper, 'half-alphabet, full-alphabet');
if (trim(str6) EQ fullalpha) then
  put str6= $hex46.;
str7 = kpropcase(lower, 'lowercase, uppercase');
if (trim(str7) EQ upper) then
  put str7=;
str8 = kpropcase(upper, 'uppercase, lowercase');
if (trim(str8) EQ lower) then
  put str8=;
RESULTS:
str1=C4DEB0C5AFC220
str2=8368815B83698362836320
str3=DO-NAXTUTSU
str4=8368815B83698362836320
str5=DO-NAXTUTSU
str6=8263826E817C826D826082778273827482738272827420
str7=DO-NAXTUTSU
str8=do-naxtutsu
```

KPROPCHAR 関数

特殊文字を標準文字に変換します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

`str=KPROPCHAR(<instr>)`

必須引数

str

結果文字列。特殊文字を標準文字に変換します。

instr

入力データ文字列。

詳細

特殊文字を標準文字に変換します。KPROPCHAR 関数では、次の範囲の文字を変換します。

- 囲まれた英数字値: \u2460 から\u24FF。 <http://www.unicode.org/charts/PDF/U2460.pdf> を参照してください。
- Dingbats: \u2776 から\u2793。 <http://www.unicode.org/charts/PDF/U2700.pdf> を参照してください。
- 囲まれた CJK 文字と月: \u3200 から\u32FF。 <http://www.unicode.org/charts/PDF/U3200.pdf> を参照してください。

例

次に、KPROPCHAR 関数の使用例を示します。次の例は UTF8 SAS セッションで処理されます。

```
length in1 out1 $30;  
in1=unicode("\u2460\u2473\u277F\u325F");  
out1=KPROPCHAR(in1);  
put out1;  
RESULTS:  
(1)(20)(-10)(35)
```

ステートメント	結果
<pre>data _null_; length in1 out1 \$30; in1=unicode('\u2460\u2473\u277F\u325F'); out1=KPROPCHAR(in1); put out1; run;</pre>	(1)(20)(-10)(35)
<pre>data _null_; length in1 out1 \$30; in1=unicode('\u2776\u2793'); out1=KPROPCHAR(in1); put out1; run;</pre>	(-1)(-10)
<pre>data _null_; length in1 out1 \$30; in1=unicode('\u3200\u32FE'); out1=KPROPCHAR(in1); put out1; run;</pre>	(ー)(ヲ)

KPROPDATA 関数

印刷不可文字を削除または変換します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

str=KPROPDATA(*<instr>* (*<options>*, *<input encode name>*, *<output encode name>*))

必須引数

str

変換された、セッションエンコーディングでのデータ文字列。

instr

入力データ文字列。

options

印刷不可文字の処理に関する指示を指定します。

UESC

Unicode エスケープ文字列(たとえば、\u0000\u1234)を使用して印刷不可文字を変換します。

TRIM

印刷不可文字を削除します。置換文字は使用しません。

BLANK or ''

各印刷不可文字を 1 バイトの空白で置換します。

QUESTION or '?'

印刷不可文字を 1 バイトの '?' で置換します。

HEX

印刷不可文字を 16 進表現(たとえば、0x810x82)で置換します。

TRUNCATE or TRUNC

最初の印刷不可文字が出現したときにデータ文字列を切り捨てます。

REMOVE

いずれかの印刷不可文字が検出された場合にデータ文字列を削除します。

NCR

Unicode にコードが存在する場合は、NCR 表示形式を使用して印刷不可文字をエンコードします。

input encode name

必要に応じて、入力データのエンコーディング名を指定します。入力エンコーディング名を指定しないと、KPROPDATA 関数では、現在の SAS セッションでエンコードされた文字列としてデータを処理します。SAS エンコーディング名の詳細については、“[データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値](#)” (613 ページ)を参照してください。

output encode name

出力データのエンコーディング名を指定します。エンコーディング名を指定しないと、KPROPDATA 関数では、現在の SAS セッションエンコーディングとして出力を認識します。SAS エンコーディング名の詳細については、“[データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値](#)” (613 ページ)を参照してください。

詳細

入力データ文字列を現在の SAS セッションエンコーディングに変換し、オプションに基づいて印刷不可文字を削除または置換します。

例

次に、KPROPDATA 関数の使用例を示します。

```
length instr $12;
length str1 str2 str3 str4 str5 str6 str7 str8 str9 str10$ 50;
instr = "534153"x||"ae"x || " System";
put instr;
str1 = kpropdata(instr);
put str1= +2 str1= $hex26.;
str2 = kpropdata(instr,'UESC');
put str2= +2 str2= $hex26.;;
str3 = kpropdata(instr, 'UESC', 'wlatin1');
put str3= +2 str3= $hex34.;
str4 = kpropdata(instr,'TRIM','wlatin1');
put str4= +2 str4= $hex26.;
str5 = kpropdata(instr,'BLANK', 'wlatin1');
put str5= +2 str5= $hex26.;
str6 = kpropdata(instr,'?', 'wlatin1');
put str6= +2 str6= $hex26.;
str7 = kpropdata(instr,'hex', 'wlatin1');
```

```

put str7= +2 str7= $hex26.;
str8 = kpropdata(instr,'TRUNC', 'wlatin1');
put str8= +2 str8= $hex26.;
str9 = kpropdata(instr,'REMOVE', 'wlatin1');
put str9= +2 str9= $hex26.;
str10 = kpropdata(instr,'NCR', 'wlatin1');
put str10= +2 str10= $hex26.;
RESULTS:
SAS? System
str1=SAS? System str1=534153AE2053797374656D2020
str2=SAS? System str2=534153AE2053797374656D2020
str3=SAS\u0066 System str3=5341535C75666636652053797374656D20
str4=SAS System str4=5341532053797374656D202020
str5=SAS System str5=534153202053797374656D2020
str6=SAS? System str6=5341533F2053797374656D2020
str7=SAS\xAE System str7=5341535C784145205379737465
str8=SAS str8=53415320202020202020202020
str9= str9=20202020202020202020202020
str10=SAS@ System str10=53415326233137343B20537973

```

KREVERSE 関数

文字式を逆にします。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[REVERSE](#) で、*SAS Viya 関数と CALL ルーチン: リファレンス*を参照してください。

構文

KREVERSE(argument)

必須引数

argument

SAS 文字式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; result=kreverse('漢字のテスト'); put result=; run;</pre>	result=トステの字漢

KRIGHT 関数

末尾の DBCS の空白と SO/SI を削除し、文字式を右詰めにします。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: ["RIGHT Function" \(SAS Viya Functions and CALL Routines: Reference\)](#)を参照してください。

構文

KRIGHT(*argument*)

必須引数

argument

SAS 文字式を指定します。

詳細

制限と詳しい情報については、["SAS 文字列関数の国際化の互換性" \(279 ページ\)](#)を参照してください。

KRIGHT 関数は、末尾の空白を値の先頭に移動した引数を返します。引数の長さは変わりません。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; a='漢字のテスト'; b=kright(a); put '----+----1----+----2----+'; put a \$25.; put b \$25.; run;</pre>	<pre>----+----1----+----2----+ 漢字のテスト 漢字のテスト</pre>

関連項目:

関数:

- “KCOMPRESS 関数” (321 ページ)
- “KLEFT 関数” (328 ページ)
- “KTRIM 関数” (349 ページ)

KSCAN 関数

指定された単語を文字式から選択します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[SCAN](#) で、*SAS Viya 関数と CALL ルーチン: リファレンス*を参照してください。

構文

KSCAN(*argument*,*n*<, *delimiters*>)

必須引数

argument

文字式を指定します。

n

KSCAN が文字式内で選択する単語の番号を算出する数式を指定します。

ヒント *n* が負の場合、KSCAN は、文字式内の単語を文字列の最後から検索して選択します。|*n*|が文字式内の単語数よりも大きい場合、KSCAN は空白値を返します。

delimiters

KSCAN が文字式内で単語区切り文字として使用する文字を指定します。

デフォルト ASCII 環境で *delimiters* を省略すると、SAS は *blank(空白)* . < (+ & ! \$ *); ^ - /, % | を使用します。^文字のない ASCII 環境の場合、KSCAN はかわりに ~文字を使用します。

EBCDIC 環境で *delimiters* を省略すると、SAS は *blank(空白)* . < (+ | & ! \$ *); ~ - /, % | ¢ を使用します。

ヒント *delimiters* を定数として表す場合は、*delimiters* を引用符で囲みます。

詳細

制限と詳しい情報については、[“SAS 文字列関数の国際化の互換性” \(279 ページ\)](#) を参照してください。

区切り文字が文字列内で先頭の単語より前にあっても、KSCAN に影響はありません。複数の区切り文字が連続する場合、KSCAN はそれらをまとめて 1 つとして扱います。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; length x \$20 y \$20; text1='これは漢字関数のテストです。'; x='nonblank'; i=1; do until(x=' '); x=kscan(text1,i,'はの'); put x; i=i+1; end; y='nonblank'; i=-1; do until(y=' '); y=kscan(text1,i,'はの'); put y; i=i-1; end; run;</pre>	<pre>x=これ x=漢字関数 x=テストです。 x= y=テストです。 y=漢字関数 y=これ y=</pre>

KSTRCAT 関数

複数の文字式を連結します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[CAT](#) で、[SAS Vija 関数](#)と [CALL ルーチン: リファレンス](#)を参照してください。

構文

KSTRCAT(*argument-1*, *argument-2*<, ... *argument-n*>)

必須引数

argument

1 バイトまたは 2 バイトの文字式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KSTRCAT 関数が事前に長さを割り当てられていない変数に値を返す場合、デフォルトで変数の長さは 200 に割り当てられます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; text1='漢字'; text2='文字列'; text3='連結'; result=kstrcat(text1,text2,'の',text3); put result=; run;</pre>	<pre>result=漢字文字列の連結</pre>

KSTRIP 関数

文字列から先頭と末尾の空白を削除します。

カテゴリ: 文字
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#) を参照してください。

注: この関数は varchar 型をサポートしていません。

構文

KSTRIP(*string*)

必須引数

string

文字定数、変数または式を指定します。

詳細

KSTRIP 関数は、先頭と末尾の SBCS の空白をすべて削除した引数を返します。引数が空白の場合、KSTRIP は長さがゼロの文字列を返します。

KSTRIP の結果を変数に割り当てる場合、受け取る変数の長さには影響しません。調整された値が受け取る変数の長さより短い場合、値の末尾に新しく空白が埋め込まれます。

DATA ステップでは、KSTRIP 関数が事前に長さを割り当てられていない変数に値を返す場合、その変数に引数の長さが割り当てられます。

比較

空白の文字列に対して、KSTRIP 関数と STRIP 関数は長さがゼロの文字列を返し、TRIM 関数は単一の空白を返します。

KSTRIP 関数と STRIP 関数は、受け取る変数の長さが入力文字列より短い場合、出力を切り捨てます。STRIP 関数は SBCS 文字列の切り捨てのみ処理します。KSTRIP 関数は SBCS データと DBCS データを処理します。

KSTRIP 関数と STRIP 関数はデータを正規化するために使用されます。KSTRIP 関数は DBCS 環境で使用する必要があります。KSTRIP 関数と STRIP 関数では、類似した処理が実行されます。

例

次の例では、KSTRIP 関数による先頭と末尾の空白の削除方法、および DBCS 文字の切り捨て方法について示します。この例では、日本語(Shift_JIS)エンコーディングを使用します。

ステートメント	結果
<pre>data _null_; length a \$5; v = unicode('\u0020\u0020\u8404\u8404\u8404\u0020'); put v = hex; a = kstrip(v); put a = hex; run;</pre>	93B893B820

関連項目:

関数

- [“STRIP Function” \(SAS Viya Functions and CALL Routines: Reference\)](#)

KSUBSTR 関数

引数から部分文字列を抽出します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: [“SUBSTR \(left of =\) Function” \(SAS Viya Functions and CALL Routines: Reference\)](#)を参照してください。

構文

KSUBSTR(*argument*, *position*<,*n*>)

必須引数

argument

SAS 文字式を指定します。

position

文字の開始位置となる数値式を指定します。

n

抽出する部分文字列の長さを示す数値式を指定します。

操作 n が *argument* 内で *position* より後に残った式の長さよりも長い場合、式の残りの部分が抽出されます。

ヒント n を省略すると、式の残りの部分が抽出されます。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ) を参照してください。

KSUBSTR 関数は、式内の *argument* で指定された部分を返します。この部分は、*position* で指定された文字から始まり、 n 文字の長さです。

KSUBSTR で作成される変数の長さは、*argument* の長さから求めることができます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; text='漢字文字列の抽出'; result=ksubstr(text,1,2); put result=; result=ksubstr(text,3,4); put result=; kstart=7; klen=2; result=ksubstr(text,kstart,klen); put result=; run;</pre>	<pre>Result=漢字文 Result=字文</pre>

関連項目:

関数:

- [“KSUBSTRB 関数” \(346 ページ\)](#)
- [“KUPDATE 関数” \(352 ページ\)](#)
- [“KUPDATEB 関数” \(354 ページ\)](#)
- [“KUPDATES 関数” \(356 ページ\)](#)

KSUBSTRN 関数

部分文字列を返します。長さがゼロの結果も返せます。

カテゴリ: DBCS

- 制限事項:** この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。
- 注:** この関数は varchar 型をサポートしていません。
- 参照項目:** 詳細については、[“SUBSTRN Function” \(SAS Viya Functions and CALL Routines: Reference\)](#) および [“SAS 文字列関数の国際化の互換性” \(279 ページ\)](#)を参照してください。

構文

KSUBSTRN(*string*, *position*<,*length*>)

必須引数

string

文字または数値の定数、変数または式を指定します。

string が数値の場合、BEST32.形式の文字値に変換されます。先頭と末尾の空白は削除され、メッセージは SAS ログに送信されません。

position

部分文字列の最初の文字の位置を指定する整数です。

オプション引数

length

部分文字列の長さを指定する整数です。*length* の指定がなければ、KSUBSTRN 関数は指定された位置から末尾までの部分文字列を返します。

詳細

KSUBSTRN の機能は SUBSTRN に似ています。違いは、第 1 引数が数値であるとき、SUBSTRN の出力長が 8 であるのに対して、KSUBSTRN の出力長は 200 である点です。

例

KSUBSTR と KSUBSTRN の違いが分かる例を示します。

ステートメント	結果
<pre>data _null_; ksubstr_result = "*" ksubstr(1234.5678,2,6) "*"; put ksubstr_result=; ksubstrn_result = "*" ksubstrn(1234.5678,2,6) "*"; put ksubstrn_result=; run;</pre>	<pre>ksubstr_result=* 1234* ksubstrn_result=*234.56*</pre>

KSUBSTRB 関数

引数内の部分文字列のバイト位置に従って、引数から部分文字列を抽出します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

KSUBSTRB(*argument*, *position*<*n*>)

必須引数

argument

SAS 文字式を指定します。

position

開始文字位置をバイト単位で指定します。

n

抽出する部分文字列の長さをバイト単位で指定します。

操作 *n* が *argument* 内で *position* より後に残った式の長さ(バイト単位)よりも長い場合、式の残りの部分が抽出されます。

ヒント *n* を省略すると、式の残りの部分が抽出されます。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KSUBSTR 関数は、式内の *argument* で指定された部分を返します。この部分は、*position* で指定されたバイト単位で始まり、*n* バイトの長さです。

KSUBSTRB で作成される変数の長さは、*argument* の長さから求めることができます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; text='漢字文字列の抽出'; result=ksubstrb(text,1,9); put result=; result=ksubstrb(text,3,7); put result=; run;</pre>	<pre>result=漢字文 result=字文</pre>

関連項目:**関数:**

- [“KSUBSTR 関数” \(344 ページ\)](#)

KTRANSLATE 関数

文字式の特定の文字を置換します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

z/OS 固有: この関数は z/OS 上でのみ実行できます。
EBCDIC コードシーケンスを使用します。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、[TRANSLATE](#) で、*SAS Viya 関数と CALL ルーチン: リファレンス*を参照してください。

構文

KTRANSLATE(*source,to-1,from-1<,...to-n,from-n>*)

必須引数**source**

元の文字値を含む SAS 式を指定します。

to

KTRANSLATE で代替文字として使用する文字を指定します。

from

KTRANSLATE で置換する文字を指定します。

操作 *to* と *from* に指定された値は、1 文字ずつ対応します。KTRANSLATE は、*from* の 1 文字を *to* の 1 文字に変換するという具合に処理を続けます。*to* の文字が *from* の文字より少ない場合、KTRANSLATE は *from* の余った文字を空白に変更します。*to* の文字が *from* の文字より多い場合、KTRANSLATE は *to* の余った文字を無視します。

注 一部の動作環境では、*to* と *from* の引数ペアが必要です。照合順序のセグメントが、NULL である *from* の引数に置き換わる動作環境もあります。

詳細

制限と詳しい情報については、[“SAS 文字列関数の国際化の互換性” \(279 ページ\)](#)を参照してください。

KTRANSLATE を使用して、1 バイト文字式を 2 バイト文字式に変換したり、2 バイト文字式を 1 バイト文字式に変換したりできます。

KTRANSLATE が受け入れる *to* および *from* の引数ペアの最大数は、SAS を実行する動作環境によって異なります。短い引数のペアを数個使用する場合と、長い引数のペアを少数使用する場合とでは、機能的な違いはありません。

KTRANSLATE 関数が事前に長さを割り当てられていない変数に値を返す場合、デフォルトで変数の長さは 200 に割り当てられます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; result=ktranslate('正課','回答','二課'); put result=; result=ktranslate('abc','アイウ','abc'); put result=; run;</pre>	<pre>result=正答 result=アイウ</pre>

KTRIM 関数

末尾の DBCS の空白と SO/SI を文字式から削除します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: 非 DBCS で相当する関数は、“[TRIM Function \(SAS Viya Functions and CALL Routines: Reference\)](#)”です。

構文

KTRIM(*argument*)

必須引数

argument

SAS 文字式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KTRIM では、文字の引数をコピーし、すべての末尾の空白を削除して、その結果調整された引数を返します。連結では末尾の空白は削除されないため、KTRIM は連結する場合に便利です。

注: 引数が空白の場合、char タイプについては KTRIM から 1 つの空白が返されます。引数が空白の場合、varchar タイプについては KTRIM から空の空白または 0 という値が返されます。varchar タイプは文字、char タイプはバイトとしてのセマンティックスに従います。

KTRIM の結果を変数に割り当てる場合、受け取る変数の長さには影響しません。調整された値が受け取る変数の長さより短い場合、その変数への割り当て時に空白で値が埋め込まれます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; part1='漢字の'; part2='テスト'; hasblank=part1 part2; noblank=ktrim(part1) part2; put hasblank; put noblank; run;</pre>	<pre>漢字の テスト 漢字のテスト</pre>

関連項目:

関数:

- [“KCOMPRESS 関数” \(321 ページ\)](#)
- [“KLEFT 関数” \(328 ページ\)](#)
- [“KRIGHT 関数” \(339 ページ\)](#)

KTRUNCATE 関数

マルチバイト文字を壊すことなく、文字列をバイト単位で、指定した長さに切ります。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

KTRUNCATE(*argument, number, length*)

必須引数

argument

SAS 文字式を指定します。

number

数値です。

length

整数です。

詳細

制限と詳しい情報については、「[SAS 文字列関数の国際化の互換性](#)」(279 ページ)を参照してください。

KTRUNCATE 関数では、完全な長さの *number*(倍精度浮動小数点型として保存)を *length* で指定したより小さいバイト数に切り捨て、切り捨てられたバイトを 0 で埋め込みます。切り捨てとその後の展開は、最初に完全な長さより短い数を保存した結果を複製し、次にそれを読み込みます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; x1=ktruncate('漢字のテスト',6); x2=ktruncate('漢字のテスト',7); x3=ktruncate('漢字のテスト',9); y1=ktruncate('漢字のテスト',5,9); y2=ktruncate('漢字のテスト',6,6); y3=ktruncate('漢字のテスト',7,6); put x1= / x2= / x3= / y1= / y2= / y3=; run;</pre>	<pre>x1=漢字 x2=漢字 x3=漢字の y1=字のテ y2=のテ y3=のテス</pre>

KUPCASE 関数

引数に含まれるシングル幅の英語のアルファベット文字がすべて大文字に変換されます。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

ヒント: “UPCASE Function” ([SAS Viya Functions and CALL Routines: Reference](#))を参照してください。

構文

KUPCASE(*argument*)

必須引数

argument

SAS 文字式を指定します。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ)を参照してください。

KUPCASE 関数により、文字引数がコピーされ、小文字のシングル幅の英語アルファベット文字がすべて大文字に変換され、変更された値が結果として返されません。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; result=kupcase('abcあいうえお'); put result=; run;</pre>	<pre>result=ABCあいうえお</pre>

KUPDATE 関数

文字値の内容を挿入、削除および置換します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性](#) (279 ページ)を参照してください。

注: この関数は varchar 型をサポートしていません。

構文

KUPDATE(*argument, position, n*<, *characters-to-replace*>)

KUPDATE(*argument, position*<, *n*> , *characters-to-replace*)

必須引数

argument

文字変数を指定します。

position

文字の開始位置となる数値式を指定します。

n

置き換える部分文字列の長さを示す数値式を指定します。

制限事項 *n* は、*argument* 内で、*position* より後に残った式の長さ以下である必要があります。

n の指定はオプションですが、この関数で *n* と *characters-to-replace* を両方とも省略することはできません。

ヒント *n* を省略すると、*characters-to-replace* のすべての文字を使用して *argument* の値を置き換えます。

characters-to-replace

argument の内容を置き換える文字式を指定します。

制限事項 *characters-to-replace* の指定はオプションですが、この関数で *characters-to-replace* と *n* を両方とも省略することはできません。

ヒント 文字のリテラル文字列を引用符で囲みます。

詳細

制限と詳しい情報については、[“SAS 文字列関数の国際化の互換性” \(279 ページ\)](#) を参照してください。

KUPDATE 関数は、*argument* の値を、*position* で指定された開始文字以降で、*characters-to-replace* の式で置き換えます。

KUPDATE 関数が事前に長さを割り当てられていない変数に値を返す場合、デフォルトで変数の長さは 200 に割り当てられます。

例

例 1

次の例は、KUPDATE と KUPDATES の違いを示します。

ステートメント	結果
<pre>data _null_; str='123456'; x1=str; substr(x1,2,3)=''; y1=kupdate(str,2,3); z1=kupdates(str,2,3); put x1= / y1= / z1=; x2=str; substr(x2,2,3)="abcd"; y2=kupdate(str,2,3,'abcd'); z2=kupdates(str,2,3,'abcd'); put x2= / y2= / z2=; x3=str; substr(x3,2,3)="ab"; y3=kupdate(str,2,3,'ab'); z3=kupdates(str,2,3,'ab'); put x3= / y3= / z3=; run;</pre>	<pre>x1=1 56 y1=156 z1=1 56 x2=1abc56 y2=1abcd56 z2=1abc56 x3=1ab 56 y3=1ab56 z3=1ab 56</pre>

例 2

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; x1=kupdate('感じのテスト',1,2,'漢字'); x2=kupdate(x1,1,2,'kanji'); x3=kupdate(x1,1,3); x4=kupdate(x1,3,'かんじ'); put x1= / x2= / x3= / x4=; run;</pre>	<pre>x1=漢字のテスト x2=kaのテスト x3=テスト x4=漢字かんじ</pre>

関連項目:

関数:

- [“KUPDATES 関数” \(356 ページ\)](#)
- [“KUPDATEB 関数” \(354 ページ\)](#)

KUPDATEB 関数

引数内の文字値のバイト位置に従って、文字値の内容を挿入、削除および置換します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

KUPDATEB(*argument*, *position*, *n*<, *characters-to-replace*>)

KUPDATEB(*argument*, *position* <, *n*> , *characters-to-replace*)

必須引数

argument

文字変数を指定します。

position

開始文字位置をバイト単位で指定します。

n

バイト単位に置き換えられる部分文字列の長さを指定します。

制限事項 *n* は、*argument* 内で、*position* より後に残った式の長さ(バイト)以下である必要があります。

n の指定はオプションですが、この関数で *n* と *characters-to-replace* を両方とも省略することはできません。

ヒント *n* を省略すると、*characters-to-replace* のすべての文字を使用して *argument* の値を置き換えます。

characters-to-replace

argument の内容を置き換える文字式を指定します。

制限事項 *characters-to-replace* の指定はオプションですが、この関数で *characters-to-replace* と *n* を両方とも省略することはできません。

ヒント 文字のリテラル文字列を引用符で囲みます。

詳細

制限と詳しい情報については、[“SAS 文字列関数の国際化の互換性” \(279 ページ\)](#) を参照してください。

KUPDATEB 関数は、*argument* の値を *characters-to-replace* の式で置き換えます。KUPDATEB は、*n* で指定したバイト単位で開始する *position* バイト単位を置き換えます。

KUPDATEB 関数が事前に長さを割り当てられていない変数に値を返す場合、デフォルトで変数の長さは 200 に割り当てられます。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; x1=kupdateb('漢字のテスト',1,9,'漢字'); x2=kupdateb(x1,1,9,'kanji'); x3=kupdateb(x1,1,9); x4=kupdateb(x1,7,'かんじ'); put x1= / x2= / x3= / x4=; run;</pre>	<pre>x1=漢字 テスト x2=kanji テスト x3=テスト x4=漢字 かんじ</pre>

関連項目:

関数:

- [“KUPDATE 関数” \(352 ページ\)](#)

KUPDATES 関数

文字値の内容を挿入、削除および置換します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用するよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

注: この関数は varchar 型をサポートしていません。

構文

KUPDATES(*argument*,*position*,*n*<, *characters-to-replace*>)

KUPDATES(*argument*,*position*<,*n*> , *characters-to-replace*)

必須引数

argument

文字変数を指定します。

position

文字の開始位置となる数値式を指定します。

オプション引数

n

置き換える部分文字列の長さを示す数値式を指定します。

制限事項 n は、*argument* 内で、*position* より後に残った式の長さ以下である必要があります。

n の指定はオプションですが、この関数で n と *characters-to-replace* を両方とも省略することはできません。

ヒント n を省略すると、*characters-to-replace* のすべての文字を使用して *argument* の値を置き換えます。

characters-to-replace

argument の内容を置き換える文字式を指定します。

制限事項 *characters-to-replace* の指定はオプションですが、この関数で *characters-to-replace* と n を両方とも省略することはできません。

ヒント 文字のリテラル文字列を引用符で囲みます。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ) を参照してください。

The KUPDATES function replaces the value of *argument* の値を *characters-to-replace* の式で置き換えます。KUPDATES は、*position* で指定された開始文字から n 文字分を置き換えます。

KUPDATES 関数が事前に長さを割り当てられていない変数に値を返す場合、デフォルトで変数の長さは 200 に割り当てられます。

例

例 1

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data null_; x1=kupdates('感じのテスト',1,2,'漢字'); x2=kupdates(x1,1,2,'kanji'); x3=kupdates(x1,1,3); x4=kupdates(x1,3,'かんじ'); put x1= / x2= / x3= / x4=; run;</pre>	<pre>x1=漢字のテスト x2=kaのテスト x3=テスト x4=漢字かんじ</pre>

例 2

次の例は、KUPDATE と KUPDATES の違いを示します。

ステートメント	結果
data _null_;	x1=1 56
str='123456';	y1=156
x1=str; substr(x1,2,3)='';	z1=1 56
y1=kupdate(str,2,3);	x2=1abc56
z1=kupdates(str,2,3);	y2=1abcd56
put x1= / y1= / z1=;	z2=1abc56
x2=str; substr(x2,2,3)="abcd";	x3=1ab 56
y2=kupdate(str,2,3,'abcd');	y3=1ab56
z2=kupdates(str,2,3,'abcd');	z3=1ab 56
put x2= / y2= / z2=;	
x3=str; substr(x3,2,3)="ab";	
y3=kupdate(str,2,3,'ab');	
z3=kupdates(str,2,3,'ab');	
put x3= / y3= / z3=;	
run;	

関連項目:

関数:

- [“KUPDATE 関数” \(352 ページ\)](#)

KVERIFY 関数

式に固有の最初の文字の位置を返します。

カテゴリ: DBCS
CAS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

ヒント: [“VERIFY Function” \(SAS Viya Functions and CALL Routines: Reference\)](#) を参照してください。

構文

KVERIFY(*source*, *excerpt-1*, ...*excerpt-n*)

必須引数

source

SAS 文字式を指定します。

excerpt-1, ...*excerpt-n*

SAS 文字式を指定します。

ヒント 文字のリテラル文字列を引用符で囲みます。

複数の *excerpt* を指定する場合は、カンマで区切ります。

詳細

制限と詳しい情報については、“[SAS 文字列関数の国際化の互換性](#)” (279 ページ) を参照してください。

KVERIFY 関数は、いずれの *excerpt* 内にも存在しない文字のうち、最初の文字の *source* 内での位置を返します。KVERIFY は、*source* 内のすべての文字が少なくとも 1 つの *excerpt* 内にあることを確認すると、0 を返します。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; input grade :\$2. @@; check='良否'; x=kverify(grade,check); if x gt 0 then put 'Invalid grade value: ' grade; cards; 良否良否良否可 ;</pre>	invalid grade value: 可

KVERIFYB 関数

式に固有の最初の文字の位置を返します。

カテゴリ: DBCS

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

KVERIFYB(*source*, *excerpt-1*,<*excerpt-n*>)

必須引数

source

SAS 文字式を指定します。

excerpt

SAS 文字式を指定します。

ヒント 複数の *excerpt* を指定する場合は、カンマで区切ります。

詳細

KVERIFYB 関数は、いずれの excerpt 内にも存在しない文字のうち、最初の文字の source 内での位置を返します。KVERIFYB は、source 内のすべての文字が、少なくとも 1 つの excerpt 内にあることを確認すると、0 を返します。

比較

KVERIFYB は、バイトベースの値を返します。KVERIFY は、文字ベースの値を返します。SBCS (たとえば、wlati1 エンコーディング)文字列を処理した場合、これらは、同一の結果を返します。一方、DBCS セッション(たとえば、EUC-JP または SHIFT-JIS エンコーディング)では、ほとんどの CJK 文字は 2 バイト幅です。このため、これらの関数は異なる結果を返します。

例

次の例では、日本語文字を使用します。

ステートメント	結果
<pre>data _null_; check='漢字'; text='漢字の検索'; x1=kverify(text,check); put x1= x2=kverifyb(text,check); put x2=; run;</pre>	<pre>x1=3 x2=7</pre>

NLDATE 関数

SAS 日付値を、日付のディスクリプタを使用して指定したロケールの日付値に変換します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

NLDATE(*date,descriptor*)

必須引数

date

SAS 日付値を指定します。

descriptor

出力での日付と時間のフォーマット方法を指定する、変数または式です。次のディスクリプタでは、大文字と小文字が区別されます。

#

先頭の 0 を出力結果から削除します。

%%

%文字を指定します。

%a

省略形の曜日のディスクリプタを指定します。省略形の曜日のディスクリプタの範囲は、Mon から Sun です。

%A

完全な曜日のディスクリプタを指定します。完全な曜日のディスクリプタの範囲は、Monday から Sunday です。

%b

省略形の月のディスクリプタを指定します。省略形の月のディスクリプタの範囲は、Jan から Dec です。

%B

完全な月のディスクリプタを指定します。完全な月のディスクリプタの範囲は、January から December です。

%C

完全な月のディスクリプタを指定し、空白埋め込みを使用します。完全な月のディスクリプタの範囲は、January から December です。

%d

日付のディスクリプタを指定し、0 埋め込みを使用します。日付修飾子の範囲は、01 から 31 です。

%e

日付のディスクリプタを指定し、空白埋め込みを使用します。日付のディスクリプタの範囲は、01 から 31 です。

%F

完全な曜日のディスクリプタを指定し、空白埋め込みを使用します。完全な曜日のディスクリプタの範囲は、Monday から Sunday です。

%j

通算日のディスクリプタを 10 進数で指定し、先頭に 0 を使用します。通算日のディスクリプタの範囲は、1 から 366 です。

%m

月のディスクリプタを指定し、0 埋め込みを使用します。月のディスクリプタの範囲は、01 から 12 です。

%o

月のディスクリプタを指定します。月のディスクリプタの範囲は 1 から 12 で、空白埋め込みを使用します。

%u

月曜日から日曜日を表す 1 から 7 の範囲の番号で、曜日のディスクリプタを指定します。

%U

年内の週数を使用してディスクリプタ値を SAS 日付値として計算することで、通算週のディスクリプタを指定します(日曜日を週の開始日と見なします)。週数の値は、0 から 53 の範囲の 10 進数で表し、先頭に 0 を使用します。最大値は 53 です。

%V

ディスクリプタ値を SAS 日付値として計算することで、通算週のディスクリプタを指定します。週数の値は、01 から 53 の範囲の 10 進数で表し、先頭に 0 を使用します。最大値は 53 です。各週は月曜日から始まり、年の第 1 週は 1 月 4 日と年の最初の木曜日の両方を含む週です。1 月の最初の月曜日が 2 日、3 日または 4 日の場合、それより前の日は前年の最後の週に組み込まれます。

%w

日曜日から土曜日を表す 0 から 6 の範囲の番号で、曜日のディスクリプタを指定します。

%W

年内の週数を使用してディスクリプタ値を SAS 日付値として計算することで、通算週のディスクリプタを指定します(月曜日を週の開始日と見なします)。週数の値は、0 から 53 の範囲の 10 進数で表し、先頭に 0 を使用します。最大値は 53 です。

%y

年(2 桁)の修飾子を指定します。年のディスクリプタの範囲は、00 から 99 です。

%Y

年(4 桁)のディスクリプタを指定します。年のディスクリプタの範囲は、1970 から 2069 です。

詳細

NLDATE 関数では、SAS 日付値を、日付のディスクリプタを使用して指定したロケールの日付値に変換します。

例

SAS 日付値から作成されたログファイル名の例を次に示します。

ステートメント	結果
<pre>options locale=English_Unitedstates; logfile=nldate('24Feb2003'd,'%B-%d.log'); put logfile;</pre>	February-24.log
<pre>options locale=German_Germany; logfile=nldate('24Feb2003'd,'%B-%d.log'); put logfile;</pre>	Februar-24.log

SAS 日付値から作成された曜日名の例を次に示します。

ステートメント	結果
	----+----1----+

ステートメント	結果
<pre>options locale=English_unitedstates; weekname=nldate('24Feb2003'd,'%A'); put weekname;</pre>	Monday
<pre>options locale=German_Germany; weekname=nldate('24Feb2003'd,'%A'); put weekname;</pre>	Montag

関連項目:

出力形式:

- [“NLDATE 出力形式” \(105 ページ\)](#)

NLDATM 関数

SAS 日時値を、日時形式のディスクリプタを使用して指定したロケールの時間値に変換します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

NLDATM(*datetime*,*descriptor*)

必須引数

datetime

SAS 日時値を指定します。

descriptor

出力での日付と時間のフォーマット方法を指定する、変数または式です。次のディスクリプタでは、大文字と小文字が区別されます。

#

先頭の 0 を出力結果から削除します。

%%

%文字を指定します。

%a

省略形の曜日のディスクリプタを指定します。省略形の曜日のディスクリプタの範囲は、Mon から Sun です。

%A

完全な曜日のディスクリプタを指定します。完全な曜日のディスクリプタの範囲は、Monday から Sunday です。

%b

省略形の月のディスクリプタを指定します。省略形の月のディスクリプタの範囲は、Jan から Dec です。

%B

完全な月のディスクリプタを指定します。完全な月のディスクリプタの範囲は、January から December です。

%c

完全な月のディスクリプタを指定し、空白埋め込みを使用します。完全な月のディスクリプタの範囲は、January から December です。

%d

日付のディスクリプタを指定し、0 埋め込みを使用します。日付のディスクリプタの範囲は、01 から 31 です。

%e

日付のディスクリプタを指定し、空白埋め込みを使用します。日付のディスクリプタの範囲は、01 から 31 です。

%F

完全な曜日のディスクリプタを指定し、空白埋め込みを使用します。完全な曜日のディスクリプタの範囲は、Monday から Sunday です。

%H

24 時間表示の時間のディスクリプタを指定します。時間のディスクリプタの範囲は、00 から 23 です。

%I

12 時間表示の時間のディスクリプタを指定します。時間のディスクリプタの範囲は、01 から 12 です。

%j

通算日のディスクリプタを 10 進数で指定し、先頭に 0 を使用します。通算日のディスクリプタの範囲は、1 から 366 です。

%m

月のディスクリプタを指定し、0 埋め込みを使用します。月のディスクリプタの範囲は、01 から 12 です。

%M

分のディスクリプタを指定します。分のディスクリプタの範囲は、00 から 59 です。

%o

月のディスクリプタを指定し、空白埋め込みを使用します。月のディスクリプタの範囲は、1 から 12 です。

%p

午前または午後のディスクリプタを指定します。

%S

秒のディスクリプタを指定します。秒のディスクリプタの範囲は、00 から 59 です。

%u

月曜日から日曜日を表す 1 から 7 の範囲の番号で、曜日のディスクリプタを指定します。

%U

年内の週数を使用してディスクリプタ値を SAS 日付値として計算することで、通算週のディスクリプタを指定します(日曜日を週の開始日と見なし

ます)。週数の値は、0 から 53 の範囲の 10 進数で表します。先頭に 0 を使用し、最大値は 53 です。

%V

ディスクリプタ値を SAS 日付値として計算することで、通算週のディスクリプタを指定します。週数の値は、01 から 53 の範囲の 10 進数で表します。先頭に 0 を使用し、最大値は 53 です。各週は月曜日から始まり、年の第 1 週は 1 月 4 日と年の最初の木曜日の両方を含む週です。1 月の最初の月曜日が 2 日、3 日または 4 日の場合、それより前の日は前年の最後の週に組み込まれます。

%w

日曜日から土曜日を表す 0 から 6 の範囲の番号で、曜日のディスクリプタを指定します。

%W

年内の週数を使用してディスクリプタ値を SAS 日付値として計算することで、通算週のディスクリプタを指定します(月曜日を週の開始日と見なします)。週数の値は、0 から 53 の範囲の 10 進数で表します。先頭に 0 を使用し、最大値は 53 です。

%y

年(2 桁)のディスクリプタを指定します。年のディスクリプタの範囲は、00 から 99 です。

%Y

年(4 桁)のディスクリプタを指定します。年のディスクリプタの範囲は、1970 から 2069 です。

詳細

NLDATM 関数では、SAS 日時値を、日時のディスクリプタを使用して指定したロケールの日時値に変換します。

例

SAS 日時値から作成された時間(午前または午後)の例を次に示します。

ステートメント	結果
	----+----1----+
options locale=English; time_ampm=nldatm('24Feb2003:12:39:43'dt,'%I%p'); put time_ampm;	12PM
options locale=German; time_ampm=nldatm('24Feb2003:12:39:43'dt,'%I%p'); put time_ampm;	12nachm

関連項目:**出力形式:**

- [“NLDATM 出力形式” \(126 ページ\)](#)

NLTIME 関数

SAS 時間値または SAS 日時値を、NLTIME ディスクリプタを使用して指定したロケールの時間値に変換します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

NLTIME(*time* | *datetime*,*descriptor*,*startpos*)

必須引数

time

SAS 時間値を指定します。

datetime

SAS 日時値を指定します。

descriptor

ディスクリプタの値を指定する、変数または式です。大文字または小文字で次のディスクリプタを入力できます。

#

先頭の 0 を出力結果から削除します。

%%

%文字を指定します。

%H

24 時間表示の時間のディスクリプタを指定します。時間のディスクリプタの範囲は、00 から 23 です。

%I

12 時間表示の時間のディスクリプタを指定します。時間のディスクリプタの範囲は、01 から 12 です。

%M

分の修飾子を指定します。分のディスクリプタの範囲は、00 から 59 です。

%P

午前または午後のディスクリプタを指定します。

%S

秒のディスクリプタを指定します。秒のディスクリプタの範囲は、00 から 59 です。

startpos

検索を開始する位置を指定し、検索の方向を指定する整数です。

詳細

NLTIME 関数では、SAS 時間値または SAS 日時値を、時間のディスクリプタを使用して指定したロケールの時間値に変換します。

例

SAS 時間から作成された午前または午後の時間の例を次に示します。

ステートメント	結果
	----+----1----+
options locale=English; time_ampm=nltime('12:39:43't,%i%p); put time_ampm;	12 PM
options locale=German; time_ampm=nltime('12:39:43't,%i%p); put time_ampm;	12 nachm

関連項目:

出力形式:

- [“NLTIME 出力形式” \(241 ページ\)](#)

SASMSG 関数

データセットからメッセージを返します。返されるメッセージは現在のロケールと指定されたキーに基づきます。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

```
SASMSG (BASENAME", "KEY", <<"QUOTE"|"DQUOTE"|"NOQUOTE">
<, "substitution 1", ..., "substitution 7">>)
```

必須引数

BASENAME

メッセージを含むデータセットの名前

KEY

メッセージキー

注: 無効なキー名を指定すると、キー名が返されます。

QUOTE|DQUOTE|NOQUOTE

メッセージテキストと代入文字列に追加する引用符のタイプを指定します。

デフォルト DQUOTE

substitution

文字列代入 文字列代入の最大件数は7です。SAS メッセージデータセットは7ビット ASCII データセットでなければなりません。

詳細

SAS メッセージデータセットは7ビット ASCII データセットでなければなりません。xxxx'は、文字の Unicode 値の10進数表現です。

SASMSG 関数で使用するデータセットは、この関数の使用専用で作成してください。データセットには次の変数を含める必要があります。

#	変数名	タイプ	長さ	説明
1	ロケール	char	5	メッセージの言語
2	key	char	60	メッセージを認識するキー
3	lineno	num	5	メッセージのライン番号(逆順)
4	text	text	1,200	メッセージのテキスト

データセットは次の変数で並べ替えている必要があります。locale, key, and lineno.変数 lineno は降順でなければなりません。locale と key の複合インデックスを定義する必要があります。並べ替えとインデックス付きのデータセットを作成するサンプルプログラムは次のようになります。

```
%let basename=MyProduct;

proc sort data=t.&basename;
  by locale key descending lineno;
run;

proc datasets lib=t
  memtype=data;
  modify &basename;
  index create indx=(LOCALE KEY);
run;
quit;
```

返されるメッセージは LOCALE システムオプションに基づきます。LOCALE オプションは *ll RR* で表されます。ll は2文字の言語コードを表し、PR は2文字の地域コードを表します。一致するものがない場合、関数は言語のみの一致を探します。locale と key のペアが見つからない場合、関数はデフォルトの英語(en)を使用します。英語(en)のキーが存在しない場合、キー名が返されます。

フォーマティングは変更することができます。フォーマットコード %s を使用して、文字列代入を使用できます。代入順序を変更する事ができます。英語以外の言語へのメッセージの翻訳では、代入順序の変更が必要な場合があります。引数番号 #nn を指定することにより、フォーマット文字列内で順序を変えることができます。nn は代入リストの引数番号になります。次の例では順序について説明します。

ステートメント	結果
<pre>msg = sasmsg ("nls.mymsg","IN_CD_LOG", "noquote","cat","dog"); IN_CD_LOGINFO = My %#1s. Your %#2s</pre>	msg= My cat. Your dog.
<pre>IN_CD_LOGINFO = My %#2s. Your %#1s</pre>	msg= My dog. Your cat.

SASMSG 関数は%SYSFUNC マクロ関数と一緒にオープンコードマクロで使用されます。

%SYSFUNC マクロで呼び出される関数に渡す引数は、二重引用符で囲まないとくたさい。%SYSFUNC 外部で SASMSG 関数に渡される引数は、二重引用符で囲む必要があります。

SASMSG 関数が%SYSFUNC マクロ関数と一緒に使われる場合、返される文字列は%NRBQUOTE 関数を使用して囲まれます。

例

例 1

次の例で SASMSG のフォーマティング機能を説明します。

```
%macro demo_sasmsg;
data _null_;
  msg = sasmsg("nls.mymsg","IN_APW_SAVE_OK","noquote");
  put msg=;
run;
%mend demo_sasmsg;
```

SAS ステートメント	結果
<pre>options locale = en_US; %demo_sasmsg ;</pre>	msg=The Access Control key was successfully saved.
<pre>options locale = es_ES; %demo_sasmsg;</pre>	msg=La clave de control de acceso se ha guardado.
<pre>options locale = french_France; %demo_sasmsg;</pre>	msg=La clé de contrôle d'accès a bien été enregistrée.

例 2

次の例でオープンマクロの機能を説明します。

```
%MACRO PRT(loc,tb,key);
  option locale=&loc;
  %PUT %SYSFUNC(SASMSG(&tb,&key) );
%MEND PRT;
```

SAS ステートメント	結果
%PRT(en_US,&TABLEID,IN_EDIT)	"Edit"
%PRT(es_ES,&TABLEID,IN_EDIT)	"Editar"
%PRT(fr_FR,&TABLEID,IN_EDIT)	"Modifier"

SASMSGGL 関数

データセットからメッセージを返します。メッセージは、指定されたロケール値と指定されたキー値に基づきます。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

```
SASMSGGL("BASENAME", "KEY", "LOCALE", (, , , <<"Q"|"D"|"N">
<, "substitution 1", ..., "substitution 6">>)
```

必須引数

BASENAME

メッセージを含むデータセットの名前

KEY

メッセージキー

注: 無効なキー名を指定すると、キー名が返されます。

LOCALE

posix ロケール値 (ll_RR).

QUOTE|DQUOTE|NOQUOTE

メッセージテキストと代入文字列に追加する引用符のタイプを指定します。

デフォルト DQUOTE

substitution

文字列代入 文字列代入の最大件数は 7 です。SAS メッセージデータセットは 6 ビット ASCII データセットでなければなりません。

詳細

SAS メッセージデータセットは 7 ビット ASCII データセットでなければなりません。7 ビット ASCII エンコーディングで表現できない文字は'luxxxx のような Unicode エスケープ形式で表現されます。xxxx は、文字の Unicode 値の 10 進数表現です。

SASMSGGL 関数で使用するデータセットは、この関数の使用専用で作成してください。データセットには次の変数を含める必要があります。

#	変数名	タイプ	長さ	説明
1	ロケール	char	5	メッセージの言語
2	key	char	60	メッセージを認識するキー
3	lineno	num	5	メッセージのライン番号 (逆順)
4	text	text	1200	メッセージのテキスト

データセットは次の変数で並べ替えている必要があります。 *locale*, *key*, and *lineno*. 変数 *lineno* は降順でなければなりません。 *locale* と *key* の複合インデックスを定義する必要があります。並べ替えとインデックス付きのデータセットを作成するサンプルプログラムは次のようになります。

```
%let basename=MyProduct;

proc sort data=t.&basename;
  by locale key descending lineno;
run;

proc datasets lib=t
  memtype=data;
  modify &basename;
  index create indx=(LOCALE KEY);
run;
quit;
```

返されるメッセージは LOCALE システムオプションに基づきます。 LOCALE オプションは *ll_rr* で表されます。 *ll* は 2 文字の言語コードを表し、 *PR* は 2 文字の地域コードを表します。一致するものがない場合、関数は言語のみの一致を探します。 *locale* と *key* のペアが見つからない場合、関数はデフォルトの英語(en)を使用します。英語(en)のキーが存在しない場合、キー名が返されます。

フォーマティングは変更することができます。フォーマットコード *%s* を使用して、文字列代入を使用できます。代入順序を変更する事ができます。英語以外の言語へのメッセージの翻訳では、代入順序の変更が必要な場合があります。引数番号 *#nn* を指定することにより、フォーマット文字列内で順序を変えることができます。 *nn* は代入リストの引数番号になります。次の例では順序の変更について説明します。

ステートメント	結果
<pre>msg = sasmsg1 ("nls.mymsg","IN_CD_LOG","en_US","N", "cat","dog"); IN_CD_LOGINFO = My %#1s. Your %#2s</pre>	<pre>msg= My cat. Your dog.</pre>
<pre>IN_CD_LOGINFO = My %#2s. Your %#1s</pre>	<pre>msg= My dog. Your cat.</pre>

SASMSG L 関数は%SYSFUNC マクロ関数と一緒にオープンコードマクロで使用できます。

%SYSFUNC マクロで呼び出される関数に渡す引数は、二重引用符で囲まないとくたさい。%SYSFUNC 外部で SASMSG L 関数に渡される引数は、二重引用符で囲む必要があります。

SASMSG L 関数が%SYSFUNC マクロ関数と一緒に使われる場合、返される文字列は%NRBQUOTE 関数を使用して囲まれます。

例

例 1

次の例で SASMSG L のフォーマティング機能を説明します。

ステートメント	結果
<code>sasmsgl("nls.mymsg", "IN_APW_SAVE_OK", "en_US", "n");</code>	Access Control key was successfully saved.
<code>sasmsgl("nls.mymsg", "IN_APW_SAVE_OK", "es_ES", "n");</code>	clave de control de acceso se ha guardado.
<code>sasmsgl("nls.mymsg", "IN_APW_SAVE_OK", "fr", "n");</code>	La clé de contrôle d'accès a bien été enregistrée.

例 2

次の例でオープンマクロの機能を説明します。

SAS ステートメント	結果
<code>%PUT %SYSFUNC(SASMSG L(NLS.MYDS, IN_ASD_LABEL("US")));</code>	Edin
<code>%PUT %SYSFUNC(SASMSG L(NLS.MYDS, IN_ASD_LABEL("ES")));</code>	Edin
<code>%PUT %SYSFUNC(SASMSG L(NLS.MYDS, IN_ASD_LABEL("fr")));</code>	Edin

SORTKEY 関数

言語並べ替えキーを作成します。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

`sortKey(string, <locale, strength, case_order, numeric_order,>)`

必須引数**string**

文字式

locale

POSIX 名(ja_JP)形式でロケール名を指定します。ロケール名と POSIX 値のリストについては、[表 17.1 \(599 ページ\)](#)を参照してください。

strength

強度の値は、照合レベルに関連付けられます。5 つの照合レベル値があります。次の表に、5 つのレベルの情報を示します。強度のデフォルト値は、ロケールに関連付けられます。

値	照合の種類	説明
PRIMARY または P	PRIMARY では、基本文字間の差異を指定します("a"<"b"など)。	この差異は最強です。たとえば、辞書は基本文字により異なるセクションに分割されません。
SECONDARY または S	文字のアクセント記号が第 2 差異となります("as"<"às"<"at"など)。	文字間のその他の差異も、言語に応じて第 2 差異と考慮される場合があります。文字列内に第 1 差異がある場合は、第 2 差異は無視されます。
TERTIARY または T	大文字と小文字の差異は、第 3 レベルで区別されます("ao"<"Ao"<"aò"など)。	例として、通常の仮名と小文字の仮名の違いがあります。文字列内に第 1 または第 2 差異がある場合は、第 3 差異は無視されます。
QUATERNARY または Q	レベル 1 から 3 で句読点が無視される場合、句読点の有無で単語を区別するために追加レベルが使用されます("ab"<"a-b"<"aB"など)。	第 1、第 2 または第 3 差異がある場合は、この差異は無視されます。句読点の無視が必須の場合や日本語テキストを処理する場合は、第 4 レベルを使用する必要があります。
IDENTICAL または I	他のすべてのレベルが等しい場合、最終的な決定をするために同一レベルが使用されます。各文字列の正規化形式 D の Unicode コードポイント値がこのレベルで比較され、レベル 1 から 4 で違いがないことが確認されます。	たとえば、ヘブライ語の詠唱マークのみがこのレベルで区別されます。2 つの文字列間でコードポイント値のみが異なることはほとんどないため、このレベルは必要最小限に使用します。

case order

大文字と小文字で並べ替えます。この引数は、TERTIARY、QUATERNARY または IDENTICAL レベルでのみ有効です。次の表に、case order 引数の値と情報を示します。

値	説明

UPPER または U	大文字、小文字の順序で並べ替えます。
LOWER または L	小文字、大文字の順序で並べ替えます。

numeric order

数を表す文字のかわりに数値で番号を並べ替えます。

値	説明
NUMERIC または N	数値で番号(整数)を並べ替えます。たとえば、"8 Main St."は"45 Main St."より前になります。

collation order

collation の値には、Phonebook と Traditional という 2 種類があります。照合値を選択しない場合、ユーザーのロケールのデフォルト照合が選択されません。次の表に、詳細を示します。

値	説明
PHONEBOOK または P	文字の電話帳スタイルの並べ替えを指定します。 PHONEBOOK はドイツ語でのみ選択します。
TRADITIONAL または T	文字の従来スタイルの並べ替えを指定します。 TRADITIONAL はスペイン語でのみ選択します。

詳細

SORTKEY 関数により、データの言語並べ替えキーが作成されます。少なくとも 1 つの引数を入力する必要があります。キーを受け取る変数の長さが十分でない場合、データが切り捨てられて警告が表示されます。

locale

ロケール値には POSIX 名(LL_RR)を使用します。LL は 2 文字の言語コードを表し、RR は 2 文字の地域コードを表します。たとえば、en_US は英語で、米国の POSIX 名です。en は英語を表し、US は米国を表します。ロケール値が指定されていない場合、セッションロケールが使用されます。

strength

strength 引数は、アクセント記号または大文字、小文字が照合または一致テキストに影響するかどうかを決定します。strength に値が指定されていない場合、ロケールにより値が決まります。次の値を strength に指定できます。

PRIMARY

基本文字が含まれます。たとえば、文字 A、a、Å のすべてが同様に処理されます。

SECONDARY

データを PRIMARY と同様に処理し、さらにアクセント記号を処理します。文字 A と a は同等に処理され、Å はアクセント記号付き文字として処理されます。

TERTIARY

データを SECONDARY と同様に処理し、さらに大文字、小文字を処理します。たとえば、A、a、Å はすべて異なる文字として処理されます。

QUATERNARY

データを TERTIARY と同様に処理し、さらに句読点を処理します。

IDENTICAL

データを QUATERNARY と同様に処理し、さらにコードポイントを処理します。

case order

大文字または小文字でデータを並べ替えるように指定します。次の表に、UPPER 値または LOWER 値を指定した例を示します。

UPPER	LOWER
Aztec	aztec
aztec	Aztec
Mars	mars
mars	Mars

collation order

照合順序値 PHONEBOOK は、ロケールがドイツ語でない場合は無視されません。

照合順序値 TRADITIONAL は、ロケールがスペイン語でない場合は無視されます。

他のロケールでは警告メッセージが表示されます。

SETLOCALE 関数

現在の SAS ロケールにロケールキーを指定します。

カテゴリ: ロケール

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用するよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

SAS ロケールの設定

SETLOCALE (*sas_locale*)

単一ロケール要素のカスタマイズ

SETLOCALE (*key,value*)

単一ロケール要素のカスタマイズ

SETLOCALE (*category_name, sas_locale*)

必須引数

sas_locale

SAS 名または Posix 名を使用して SAS ロケール名を指定します。ロケールのエイリアスを指定する事もできます。

key

SAS ロケール要素キーを指定します。詳細セクションの要素キーのリストを参照してください。

value

ロケール要素の値を指定します。

category_name

カテゴリ名を指定します。

- LC_TIME
- LC_MONETARY
- LC_NUMERIC
- LC_ALL

詳細

次のロケール要素を変更することができます。 *key* の値は *max length* の値よりも小さくする必要があります。 *type* に次の値を指定できます。

- 0 文字列
- 1 符号なし整数。二重引用符を使用しなくてはなりません。

ロケール要素キー	最大長	タイプ	カテゴリ
DATESTYLE	3	0	
PAPERSIZE	8	0	
FTITLE	512	0	
FTEXT	512	0	
SIMFONT	512	0	
SORTSEQ	8	0	
MESSAGES	8	0	
FORMATNAME_DATE	512	0	
FORMATNAME_DATETIME	512	0	
FORMATNAME_TIME	512	0	

ロケール要素キー	最大長	タイプ	カテゴリ
FORMATNAME_NU MERIC	512	0	
FORMATNAME_PE RCENT	512	0	
FONT_SERIF	32	0	
FONT_SANSERIF	32	0	
FONT_CURSIVE	32	0	
FONT_FANTASY	32	0	
FONT_MONOSPAC E	32	0	
BRUSH	32	0	
SIMPLEX	32	0	
COMPLEX	32	0	
SWISS	32	0	
ITALIC	32	0	
DATE_FORMAT	512	0	LC_TIME
DATE_SHORT_FOR MAT	512	0	LC_TIME
DATETIME_AMPM_ FORMAT	512	0	LC_TIME
DATETIME_FORMA T	512	0	LC_TIME
DATETIME_SHORT _FORMAT	512	0	LC_TIME
DATETIME_WEEK_ FORMAT	512	0	LC_TIME
DATETIME_WEEK_ SHORT_FORMAT	512	0	LC_TIME
TIME_AMPM_FOR MAT	512	0	LC_TIME
TIME_FORMAT	512	0	LC_TIME

ロケール要素キー	最大長	タイプ	カテゴリ
DATE_WEEK_FORMAT	512	0	LC_TIME
DATE_WEEK_SHORT_FORMAT	512	0	LC_TIME
DATE_YMMM_FORMAT	512	0	LC_TIME
DATE_YMMM_SHORT_FORMAT	512	0	LC_TIME
DATE_MMDD_FORMAT	512	0	LC_TIME
DATE_MMDD_SHORT_FORMAT	512	0	LC_TIME
DATE_YEAR_FORMAT	512	0	LC_TIME
DATE_YEAR_SHORT_FORMAT	512	0	LC_TIME
DATE_YYQQ_FORMAT	512	0	LC_TIME
DATE_YYQQ_SHORT_FORMAT	512	0	LC_TIME
DATE_YYWW_FORMAT	512	0	LC_TIME
DATE_YYWW_SHORT_FORMAT	512	0	LC_TIME
DATE_SEP	8	0	LC_TIME
ABMON01	512	0	LC_TIME
ABMON02	512	0	LC_TIME
ABMON03	512	0	LC_TIME
ABMON04	512	0	LC_TIME
ABMON05	512	0	LC_TIME
ABMON06	512	0	LC_TIME
ABMON07	512	0	LC_TIME

ロケール要素キー	最大長	タイプ	カテゴリ
ABMON08	512	0	LC_TIME
ABMON09	512	0	LC_TIME
ABMON10	512	0	LC_TIME
ABMON11	512	0	LC_TIME
ABMON12	512	0	LC_TIME
MON01	512	0	LC_TIME
MON02	512	0	LC_TIME
MON03	512	0	LC_TIME
MON04	512	0	LC_TIME
MON05	512	0	LC_TIME
MON06	512	0	LC_TIME
MON07	512	0	LC_TIME
MON08	512	0	LC_TIME
MON09	512	0	LC_TIME
MON10	512	0	LC_TIME
MON11	512	0	LC_TIME
MON12	512	0	LC_TIME
ABDAY1	512	0	LC_TIME
ABDAY2	512	0	LC_TIME
ABDAY3	512	0	LC_TIME
ABDAY4	512	0	LC_TIME
ABDAY5	512	0	LC_TIME
ABDAY6	512	0	LC_TIME
ABDAY7	512	0	LC_TIME
DAY1	512	0	LC_TIME

ロケール要素キー	最大長	タイプ	カテゴリ
DAY2	512	0	LC_TIME
DAY3	512	0	LC_TIME
DAY4	512	0	LC_TIME
DAY5	512	0	LC_TIME
DAY6	512	0	LC_TIME
DAY7	512	0	LC_TIME
AM	512	0	LC_TIME
PM	512	0	LC_TIME
ABQTR1	512	0	LC_TIME
ABQTR2	512	0	LC_TIME
ABQTR3	512	0	LC_TIME
ABQTR4	512	0	LC_TIME
QTR1	512	0	LC_TIME
QTR2	512	0	LC_TIME
QTR3	512	0	LC_TIME
QTR4	512	0	LC_TIME
INT_CURRENCY_SY MBOL	3	0	LC_MONETARY
CURRENCY_SYMB OL	32	0	LC_MONETARY
MON_DECIMAL_P OINT	8	0	LC_MONETARY
MON_THOUSAND S_SEP	8	0	LC_MONETARY
MON_GROUPING	3	1	LC_MONETARY
MON_POSITIVE_SI GN	8	0	LC_MONETARY
MON_NEGATIVE_S IGN	8	0	LC_MONETARY

ロケール要素キー	最大長	タイプ	カテゴリ
MON_INT_FRAC_DIGITS	3	1	LC_MONETARY
MON_FRAC_DIGITS	3	1	LC_MONETARY
MON_P_CS_PRECEDES	3	1	LC_MONETARY
MON_P_SEP_BY_SPACE	3	1	LC_MONETARY
MON_P_SIGN_POSN	3	1	LC_MONETARY
MON_N_SIGN_POSN	3	1	LC_MONETARY
DECIMAL_POINT	1	0	LC_NUMERIC
THOUSANDS_SEP	1	0	LC_NUMERIC
GROUPING	3	1	LC_NUMERIC
POSITIVE_SIGN	8	0	LC_NUMERIC
NEGATIVE_SIGN	8	0	LC_NUMERIC
P_CS_PRECEDES	3	1	LC_NUMERIC
P_SEP_BY_SPACE	3	1	LC_NUMERIC
N_CS_PRECEDES	3	1	LC_NUMERIC
P_SEP_BY_SPACE	3	1	LC_NUMERIC
N_CS_PRECEDES	3	1	LC_NUMERIC
N_SEP_BY_SPACE	3	1	LC_NUMERIC
P_SIGN_POSN	3	1	LC_NUMERIC
N_SIGN_POSN	3	1	LC_NUMERIC
HEIGHT	3	1	
WIDTH	3	1	

例

例 1

次の例では、SETLOCALE 関数は日本語(jp_JP)のロケールを指定します。SETLOCALE 関数は以前のロケールを返します。この例では、以前のロケールは English_United States です。

ステートメント	結果
<pre>data _null_; x=setlocale('ja_JP'); put x=; run;</pre>	x=English_UnitedStates

例 2

次の例では、SETLOCALE 関数は要素値が変更されているロケール名を返します。

ステートメント	結果
<pre>data _null_; x = setlocale("LC_MONETARY", 'zh_CN'); put x=; run;</pre>	x=japanese_Japan

例 3

次の例では、SETLOCALE 関数は指定された DATE_YEAR_FORMAT キーの値を変更します。

ステートメント	結果
<pre>data _null_; x=setlocale('DATE_YEAR_FORMAT', '¥%Y'); put x=; run;</pre>	x=%Y¥

TZONEID 関数

現在のタイムゾーン ID を返します。

カテゴリ: 日付と時間

別名: TZID

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

この関数は CAS server では使用できません。

構文

TZONEID<*time-zone-id*>

オプション引数

time-zone-id

SAS で定義された region 値または area 値を指定します。ゾーン ID の指定時、タイムゾーン名と夏時間ルールに基づきタイムゾーンが決定されます。

詳細

TZONEID 関数は、TIMEZONE=オプションが空白の場合か、またはユーザー定義のタイムゾーンが指定されている場合に空白値を返します。

TZONEID 関数は、タイムゾーン ID を検証します。タイムゾーン ID が指定されている場合、その ID が有効であればタイムゾーン ID を返し、無効であれば空白値を返します。

例

最初の例では、TIMEZONE オプションが JST に設定されています。2 番目の例では、TIMEZONE が空白値に設定されています。3 番目の例では、TIMEZONE がユーザー指定のタイムゾーンに設定されています。4 番目の例では、有効なタイムゾーン ID と無効なタイムゾーン ID が表示されます。

ステートメント	結果
<pre>options timezone=jst; data _null_; tzid=tzoneid(); put tzid=; run;</pre>	tzid=ASIA/TOKYO
<pre>options timezone=""; data _null_; tzid=tzoneid(); put tzid=; run;</pre>	tzid=
<pre>options timezone='xxx-12'; /* user defined timezone */ data _null_; tzid=tzoneid(); put tzid=; run;</pre>	tzid=
<pre>data null; name_valid=tzoneid('asia/tokyo'); name_invalid=tzoneid('Milky Way'); put name_valid =; put name_invalid=; run;</pre>	name_valid=ASIA/TOKYO name_invalid=

TZONENAME 関数

現在の標準時間または夏時間のタイムゾーン名を返します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

この関数は CAS server では使用できません。

構文

TZONENAME()

TZONENAME<time-zone-id,datetime>

オプション引数

time-zone-id

SAS で定義された *region* 値または *area* 値を指定します。ゾーン ID の指定時、タイムゾーン名と夏時間ルールに基づきタイムゾーンが決定されます。

参照項目: タイムゾーン ID のリストについては、[付録 1, "タイムゾーン ID とタイムゾーン名" \(627 ページ\)](#)を参照してください。

datetime

SAS 日時値を指定します。

詳細

TIMEZONE=オプションが空白の場合、TZONENAME 関数は空白値を返します。

TZONENAME は、指定されたタイムゾーンと日時に基づいてタイムゾーン名を返します。SAS 日時が指定されていない場合、現在の日時が使用されます。TZID が指定されていない場合、TIMEZONE=オプションで指定されたタイムゾーン ID が使用されます。

例

最初の例では、TIMEZONE オプションが空白値に設定されています。2 番目の例では、TIMEZONE がタイムゾーン名 JST に設定されています。3 番目の例では、TIMEZONE がユーザー指定のタイムゾーンに設定されています。4 番目の例では、TIMEZONE がタイムゾーン ID に設定されています。

ステートメント	結果
<pre>options tz=""; data _null_; tzname=tzonename(); put tzname =; run;</pre>	<pre>tzname=</pre>

ステートメント	結果
<pre>options tz='jst'; data _null_; tzname=tzname(); put tzname =; run;</pre>	tzname=JST
<pre>options tz='xxx-12'; data _null_; tzname=tzname(); put tzname =; run;</pre>	tzname=XXX
<pre>options tz='American/Chicago'; data _null_; tzname=tzname('01SEP2014:01:01:01'dt); put tzname =; run;</pre>	tzname=CDT

TZONEOFF 関数

ユーザーのタイムゾーンオフセットを返します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

この関数は CAS server では使用できません。

構文

TZONEOFF()

TZONEOFF(<time-zone-id, datetime>)

オプション引数

time-zone-id

SAS で定義された *region* 値または *area* 値を指定します。タイムゾーン ID の指定時、タイムゾーン名と夏時間ルールに基づきタイムゾーンが決定されます。

参照項 目: タイムゾーン ID のリストについては、[付録 1, “タイムゾーン ID とタイムゾーン名” \(627 ページ\)](#)を参照してください。

datetime

SAS 日時値を指定します。

詳細

引数が指定されていない場合は、TZONEOFF 関数は、指定された TIMEZONE オプションのタイムゾーンオフセットを返します。引数としてタイムゾーン ID を

指定した TZONEOFF(time-zone-id)関数は、指定したタイムゾーン ID に対するタイムゾーンオフセットを返します。引数としてタイムゾーン名を指定した TZONEOFF 関数は、指定したタイムゾーン名に対するタイムゾーンオフセットを返します。タイムゾーン名が現在のロケールで無効な場合は、エラーを受け取ります。タイムゾーン ID はロケール依存でないため、タイムゾーン ID の使用をお勧めします。

SASDTM が指定されていない場合は、TZONEOFF は現在のタイムゾーンオフセットを返します。SASDTM が指定されている場合は、指定された時間値のローカル時間を取得するためのオフセットを返します。

例

最初の例では、引数がないため、TZONEOFF 関数は現在の SAS セッションに対するオフセットを返します。2 番目の例では、指定したタイムゾーン ID に基づいてオフセットを返します。3 番目の例では、指定したタイムゾーン ID および指定の日付と時間に基づいてオフセットを返します。4 番目の例では、TIMEZONE オプション値と指定の日付に基づいてオフセットを返します。

SAS 日時が指定されていない場合は、TZONEOFF 関数は現在のタイムゾーンオフセットを返します。SAS 日時が指定されている場合は、指定された時間値のローカル時間を提供するためのオフセットを返します。

ステートメント	結果
option TIMEZONE='AUSTRALIA/MELBOURNE'; %PUT %SYSFUNC(TZONEOFF());	39600
option TIMEZONE='AUSTRALIA/MELBOURNE'; %PUT %SYSFUNC(TZONEOFF(EUROPE/ROME));	3600
data _null_ ; dt1='05DEC2012:08:17:52'dt ; dt2='05JUN2012:08:17:52'dt ; offset1= TZONEOFF('EUROPE/MOSCOW', dt1); offset2= TZONEOFF('EUROPE/MOSCOW', dt2); put offset1= / offset2= ; run ;	offset1=10800 offset2=14400
option TIMEZONE='EUROPE/MOSCOW' ; data _null_ ; dt1='05DEC2012:08:17:52'dt ; dt2='05JUN2012:08:17:52'dt ; offset1= TZONEOFF(dt1); offset2= TZONEOFF(dt2); put offset1= / offset2= ; run ;	offset1=10800 offset2=14400

TZONES2U 関数

SAS 日時値を UTC 日時値へ変換します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。
この関数は CAS server では使用できません。

構文

TZONES2U <datetime, time-zone-id>

オプション引数

datetime

SAS 日時値を指定します。

time-zone-id

SAS で定義された region 値または area 値を指定します。タイムゾーン ID の指定時、タイムゾーン名と夏時間ルールに基づきタイムゾーンが決定されます。

参照項目: タイムゾーン ID のリストについては、[付録 1, "タイムゾーン ID とタイムゾーン名" \(627 ページ\)](#)を参照してください。

詳細

TZONES2U()関数は、指定された TIMEZONE の UTC ベースの時間を返します。引数としてタイムゾーン ID を指定した TZONES2U(time-zone-id)関数は、指定したタイムゾーン ID に対する UTC ベースの時間を返します。タイムゾーン名が現在のロケールで無効な場合は、エラーを受け取ります。

例

次の例では、SAS 日時値を UTC 時間値へ変換します。

ステートメント	結果
option locale=ja_JP TZ='JST'; data _null_; dt='05Nov2012:08:17:52'dt; utc1 = tzones2u(dt); utc2 = tzones2u(dt,'ASIA/TOKYO'); utc3 = tzones2u(dt,'JST'); put dt= /utc1= is8601dz. //; put dt= /utc2= is8601dz. //; put dt= /utc3= is8601dz. //; run;	dt=1667722672 utc1=2012-11-04T23:17:52+00:00 dt=1667722672 utc2=2012-11-04T23:17:52+00:00 dt=1667722672 utc3=2012-11-04T23:17:52+00:00

TZONEDSTNAME 関数

夏時間名を返します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。
この関数は CAS server では使用できません。

構文

TZONEDSTNAME()

TZONEDSTNAME<*time-zone-id*>

オプション引数

time-zone-id

SAS で定義された region 値または area 値を指定します。タイムゾーン ID を指定するとき、タイムゾーン名と夏時間ルールに基づいて、SAS で使用されるタイムゾーンが決定されます。

詳細

TZID が指定されていない場合、TIMEZONE=オプションで指定された TIMEZONE ID が使用されます。夏時間が指定されていない場合は、空白名が返されます。

例

次の例では、米国/シカゴのタイムゾーンを指定します。

ステートメント	結果
<pre>options timezone='America/Chicago'; data null; dstname=tzonedstname(); put dstname; run;</pre>	dstname=CDT

TZONEDSTOFF 関数

指定された夏時間のタイムゾーンオフセット値を返します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。
この関数は CAS server では使用できません。

構文

TZONEDSTOFF()

TZONEDSTOFF<*time-zone id*>

オプション引数

time zone id

SAS で定義された region 値または area 値を指定します。タイムゾーン ID を指定するとき、タイムゾーン名と夏時間ルールに基づいて、SAS で使用されるタイムゾーンが決定されます。

詳細

TZID が指定されていない場合、TIMEZONE=オプションで指定された TIMEZONE ID が使用されます。夏時間が指定されていない場合は、空白名が返されます。

例

次の例では、米国/シカゴのタイムゾーンを指定します。

ステートメント	結果
<pre>options timezone='America/Chicago'; data null; dstoff=tzonedstoffs(); put dstoff; run;</pre>	dstoff=18000

TZONESTTNAME 関数

標準のタイムゾーン名を返します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

この関数は CAS server では使用できません。

構文

TZONESTTNAME()

TZONESTTNAME<time-zone id>

オプション引数

time-zone-id

SAS で定義された region 値または area 値を指定します。タイムゾーン ID を指定するとき、タイムゾーン名と夏時間ルールに基づいて、SAS で使用されるタイムゾーンが決定されます。

詳細

TZID が指定されていない場合、TIMEZONE=オプションで指定された TIMEZONE ID が使用されます。

例

次の例では、アジア/大阪のタイムゾーンを指定します。

ステートメント	結果
<pre>options timezone='Asia/Osaka'; data null; name=tzonesttname(); put name; run;</pre>	JST

TZONESTTOFF 関数

指定された標準時間のタイムゾーンオフセット値を返します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

この関数は CAS server では使用できません。

構文

TZONESTTOFF()

TZONESTTOFF<time-zone id>

オプション引数

time-zone-id

SAS で定義された region 値または area 値を指定します。タイムゾーン ID を指定するとき、タイムゾーン名と夏時間ルールに基づいて、SAS で使用されるタイムゾーンが決定されます。

詳細

TZID が指定されていない場合、TIMEZONE=オプションで指定された TIMEZONE ID が使用されます。標準時が指定されていない場合は、空白名が返されます。

例

次の例では、アジア/大阪のタイムゾーンを指定します。

ステートメント	結果
<pre>options timezone='Asia/Osaka'; data null; name=tzoneststoff(); put name; run;</pre>	32400

TZONEU2S 関数

UTC 日時値を SAS 日時値へ変換します。

カテゴリ: 日付と時間

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

この関数は CAS server では使用できません。

構文

TZONEU2S <UTC date time value,time-zone id>

オプション引数

UTC date time value

UTC (Coordinated Universal Time)日時値を指定します。

time-zone id

SAS で定義された region 値または area 値を指定します。ゾーン ID の指定時、タイムゾーン名と夏時間ルールに基づきタイムゾーンが決定されます。

詳細

TZONEU2S(datetime)関数は、指定された TIMEZONE オプションの UTC 時間に対する SAS 日時を返します。引数としてタイムゾーン ID を指定した

TZONEU2S(datetime, time-zone-id)関数は、指定したタイムゾーン ID の UTC 時間に対する SAS 日時を返します。タイムゾーン名が現在のロケールで無効な場合は、エラーを受け取ります。

例

次の例では、1 つの UTC 日時値を 3 つの特定の SAS 日時値へ変換します。

ステートメント	結果
<pre>option locale=fr_FR TZ='AMERICA/DENVER'; data _null_; utc_date = '2012-09-02T02:34:56+00:00'; udt = input(utc_date,is8601dz.); sdt1 = tzoneu2s(udt); sdt2 = tzoneu2s(udt,'EUROPE/AMSTERDAM'); sdt3 = tzoneu2s(udt,'CET'); put sdt1= datetime. / sdt2= datetime. / sdt3= datetime.; run;</pre>	<pre>sdt1=01SEP12:20:34:56 sdt2=02SEP12:03:34:56 sdt3=02SEP12:03:34:56</pre>

TRANTAB 関数

指定された変換テーブルを使用してデータをトランスコードします。

カテゴリ: 文字

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

TRANTAB(*string*, *trantab_name*)

必須引数

string

トランスコードされる入力データ。

trantab_name

変換テーブル。

詳細

TRANTAB 関数では、変換テーブルを使用して 1 つの内部表示形式から別の内部表示形式に文字を再マッピングすることによって、データ文字列をトランスコードします。入力文字列のデータのエンコーディングは、変換テーブルのテーブル 1 のエンコーディングと一致する必要があります。TRANTAB 関数では、テーブル 1 を使用してエンコーディングからデータを再マッピングします。

変換テーブルは各国語の要件をサポートするために SAS 6 で導入されました。SAS 8.2 では、変換テーブルの直接使用より優れた、LOCALE=および ENCODING=システムオプションが導入されました。SAS 9 では、変換テーブルは、外部ファイルのトランスコーディングにのみに使用されます。セッションエンコーディング間で、SAS ファイルに直接トランスコーディングされます。SAS 9.2 では、TRANTAB 関数が下位互換性を維持するためにサポートされています。

これ以降のリリースでは、LOCALE=システムオプションをお勧めします。

注意:

TRANTAB 関数は、経験を積んだ SAS ユーザーのみが使用してください。

例

次の例では、Latin2 でエンコードされたデータを大文字の Latin2 エンコーディングにトランスコードする変換テーブルを使用します。

ステートメント	結果
teststrg=trantab('testing','lat2_ucs'); put teststrg;	TESTING

UNICODE 関数

Unicode 文字を現在の SAS セッションエンコーディングに変換します。

カテゴリ: 文字

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

STR=**UNICODE**(*<instr>* (,*<Unicode type>*))

必須引数

str

現在の SAS セッションエンコーディングに変換されたデータ文字列。

instr

入力データ文字列。

Unicode type

Unicode 文字形式

ESC	Unicode エスケープ(たとえば、\u0042)。ESC はデフォルトの形式です。
NCR	数値文字表現(たとえば、大 や±)
PAREN	Unicode かっこエスケープ(たとえば、<u0061>)
UTF8	UTF8 エンコーディング
UTF16	ビッグエンディアンの UTF16 エンコーディング UCS2 はエイリアスです。
UTF16B	ビッグエンディアンの UTF16 エンコーディング UCS2B はエイリアスです。
UTF16L	リトルエンディアンの UTF16 エンコーディング UCS2L はエイリアスです。
UTF32	ビッグエンディアンの UTF32 エンコーディング UCS4 はエイリアスです。

- UTF32B ビッグエンディアンの UTF32 エンコーディング UCS4B はエイリアスです。
- UTF32L リトルエンディアンの UTF32 エンコーディング UCS4L はエイリアスです。

詳細

Unicode 文字を読み込み、現在の SAS セッションエンコーディングに変換します。

Unicode 関数は Unicode 補助文字をサポートしています。

例

次に、UNICODE 関数の使用例を示します。

```
/* Run this program using any CJK LOCALE */
data _null_;
str1=unicode("\u0041\u0042\u0043");put str1; /* default */
str2=unicode("\u0041\u0042\u0043", "esc");put str2; /* ESC - Unicode Escape */
str3=unicode("&#177;", "ncr");put str3; /* NCR - Numeric Character Representation */
str4=unicode("&#22823;", "ncr");put str4; /* NCR - Numeric Character Representation */
str5=unicode("<u0061><u0062>", 'paren');put str5; /* PAREN - Unicode Parenthesis Escape */
str6=unicode('2759'x, 'ucs2');put str6; /* UCS2 - UCS2 encoding */
str7=unicode('5927'x, 'ucs2b');put str7; /* UCS2B - UCS2 encoding with big endian */
str8=unicode('2759'x, 'ucs2l');put str8; /* UCS2L - UCS2 encoding with little endian */
str9=unicode('27590000'x, 'ucs4');put str9; /* UCS4 - UCS4 encoding */
str10=unicode('00005927'x, 'ucs4b');put str10; /* UCS4B - UCS4 encoding with big endian */
str11=unicode('27590000'x, 'ucs4l');put str11; /* UCS4L - UCS4 encoding with little endian */
str12=unicode('E5A4A7'x, 'utf8');put str12; /* UTF8 - UTF8 encoding */
str13=unicode('2759'x, 'utf16');put str13; /* UTF16 - UTF16 encoding */
str14=unicode('5927'x, 'utf16b');put str14; /* UTF16B - UTF16 encoding with big endian */
str15=unicode('2759'x, 'utf16l');put str15; /* UTF16L - UTF16 encoding with little endian */
str16=unicode("\u00020bb7", "esc");put str16; /* ESC - Unicode Escape for Supplementary Character */
str17=unicode("&#134071;", "ncr");put str17; /* NCR - Numeric Character Representation for
Supplementary Character */
str18=unicode('<u00020BB7>', 'paren');put str18; /* PAREN - Unicode Parenthesis Escape for
Supplementary Character*/
run;
```

UNICODE 関数例の結果を次に示します。

```
str1=ABC str2=ABC str3=± str4=大 str5=ab str6=大 str7=大 str8=大 str9=大 str10=大
str11=大 str12=大 str13=大 str14=大 str15=大
```

UNICODEC 関数

現在の SAS セッションエンコーディングの文字を Unicode 文字に変換します。

カテゴリ: 文字

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

```
STR=UNICODEC(<instr> (<Unicode type> ))
```

必須引数

str

Unicode エンコーディングに変換されたデータ文字列。

instr

入力データ文字列。

Unicode type

Unicode 文字形式

ESC	Unicode エスケープ(たとえば、\u0042)。ESC はデフォルトの形式です。
NCR	数値文字表現(たとえば、大 や±))
PAREN	Unicode っこエスケープ(たとえば、<u0061>)
UTF8	UTF8 エンコーディング
UTF16	ビッグエンディアンの UTF16 エンコーディング UCS2 はエイリアスです。
UTF16B	ビッグエンディアンの UTF16 エンコーディング UCS2B はエイリアスです。
UTF16L	リトルエンディアンの UTF16 エンコーディング UCS2L はエイリアスです。
UTF32	ビッグエンディアンの UTF32 エンコーディング UCS4 はエイリアスです。
UTF32B	ビッグエンディアンの UTF32 エンコーディング UCS4B はエイリアスです。
UTF32L	ビッグエンディアンの UTF32L エンコーディング UCS4L はエイリアスです。

詳細

現在の SAS セッションエンコーディングに含まれる文字を読み込み、Unicode エンコーディングに変換します。

例

次に、UNICODEC 関数の使用例を示します。

ステートメント	結果
data _null_;	str1=414243202020 str2=414243C3A920
str1=unicodec("ABC", 'utf8'); put str1= \$hex12.; str2=unicodec("ABCé", 'utf8'); put str2= \$hex12.; run;	

UNICODELEN 関数

Unicode データの文字単位の長さを示します。

カテゴリ: 文字

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使用できるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

UNICODELEN()

詳細

UNICODELEN 関数では、Unicode データの文字単位の長さを示します。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
len1=unicodelen("abc 太");	len1=4
len2=unicodelen("\u0041\u0042\u0043\u5927", 'esc');	len2=4
len3=unicodelen("大', 'ncr');	len3=1
len4=unicodelen("<u0061><u0062>", 'paren');	len4=2

関連項目:

関数:

- [“UNICODEWIDTH 関数” \(397 ページ\)](#)

UNICODEWIDTH 関数

Unicode データの表示単位の長さを示します。

カテゴリ: 文字

制限事項: この関数には I18N レベル 2 のステータスが付与されており、SBCS、DBCS および MBCS(UTF8)で使えるよう設計されています。詳細については、[国際化の互換性 \(279 ページ\)](#)を参照してください。

構文

UNICODEWIDTH()

詳細

UNICODEWIDTH 関数では、Unicode データの表示単位の長さを示します。表示単位は、文字を固定幅フォントで表示する時の文字の幅を示します。

CJK 記号の文字、CJK 統合漢字、全角アルファベットと句読点、全角通貨記号、CJK 統合漢字拡張 B-D および CJK 互換漢字補助は、表示単位の値が 2 になります。他の文字は表示単位の値が 1 になります。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
<code>len1=unicodewidth("abc 𠮟");</code>	len1=5
<code>len2=unicodewidth("\u0041\u0042\u0043\u5927",'esc');</code>	len2=5
<code>len3=unicodewidth("&#22823; ','ncr');</code>	len3=2
<code>len4=unicodewidth("<u0061><u0062>','paren');</code>	len4=2

関連項目:

関数:

- [“UNICODELEN 関数” \(396 ページ\)](#)

VARTRANSCODE 関数

SAS データセット変数のトランスコーディング属性を返します。

カテゴリ: 変数情報

構文

VARTRANSCODE(*data-set-id*, *var-num*)

必須引数

data-set-id

OPEN 関数が返すデータセット識別子を指定します。

var-num

SAS データセット内での変数の位置を指定します。

ヒント VARNUM 関数はこの値を返します。

詳細

トランスコーディングとは、1つのエンコーディングから別のエンコーディングにデータを変換する処理のことです。VARTRANSCODE 関数は、*var-num* 変数がある値をトランスコードしない場合は 0 を、*var-num* 変数がある値をトランスコードする場合は 1 を返します。

変数のトランスコーディングの詳細については、*SAS Viya 各国語サポート: リファレンスガイドのトランスコーディング (19 ページ)*を参照してください。値のエンコーディングおよびデータのトランスコーディングの詳細については、*SAS Viya 各国語サポート: リファレンスガイドの SAS データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値 (613 ページ)*を参照してください。

例

次の例は、文字変数をトランスコードするかどうかの決定方法を示しています。

```
data a;
  attrib x length=$3. transcode=no;
  attrib y length=$3. transcode=yes;
  x='abc';
  y='xyz';
run;
data _null_;
  dsid=open('work.a','i');
  nobs=attrn(dsid,"nobs");
  nvars=attrn(dsid,"nvars");
  do i=1 to nobs;
    xrc=fetch(dsid,1);
    do j=1 to nvars;
      transcode = vartranscode(dsid,j);
      put transcode=;
    end;
  end;
run;
```

SAS は次の出力をログに書き込みます。

```
transcode=0
transcode=1
```

関連項目:

関数:

- [“ATTRN Function” \(SAS Viya Functions and CALL Routines: Reference\)](#)
- [“OPEN Function” \(SAS Viya Functions and CALL Routines: Reference\)](#)
- [“VARNUM Function” \(SAS Viya Functions and CALL Routines: Reference\)](#)
- [“VTRANSCODE 関数” \(399 ページ\)](#)
- [“VTRANSCODEX 関数” \(400 ページ\)](#)

VTRANSCODE 関数

指定された文字変数でトランスコーディングが有効かどうかを示す値を返します。

カテゴリ: 変数情報
CAS

注: この関数は varchar 型をサポートしていません。

構文

VTRANSCODE (*var*)

必須引数

var

スカラまたは配列参照として表される文字変数を指定します。

制限事項 式は引数として使用できません。

詳細

VTRANSCODE 関数は、トランスコーディングがオフの場合は 0 を、オンの場合は 1 を返します。

デフォルトでは、DATA ステップのすべての文字変数がトランスコードされます。トランスコーディングをオフにするには、ATTRIB ステートメントの TRANSCODE=属性を使用します。

比較

- VTRANSCODE 関数は、指定された変数でトランスコーディングが有効かどうかを示す値を返します。一方、VTRANSCODEX 関数は、変数名を判別するために引数を評価します。その後、VTRANSCODEX 関数はその変数名に関連付けられたトランスコーディングステータス(オンまたはオフ)を返します。
- VTRANSCODE 関数は、引数として式を受け入れません。VTRANSCODEX 関数は式を受け入れますが、指定された式の値で配列参照を示すことはできません。
- 関連する関数が、変数名、種類、出力形式、長さなど他の変数属性の値を返します。変数属性のリストについては、“変数情報”の関数(SAS Viya 関数と CALL ルーチン: リファレンス)を参照してください。

例

ステートメント	結果
	----+----1----+
attrib x transcode = yes; attrib y transcode = no; rc1 = vtranscode(y); put rc1=;	rc1=0

関連項目:

関数:

- [“VTRANSCODEX 関数” \(400 ページ\)](#)

ステートメント:

- [ATTRIB in](#)

VTRANSCODEX 関数

指定された引数でトランスコーディングが有効かどうかを示す値を返します。

カテゴリ: 変数情報
CAS

注: この関数は varchar 型をサポートしていません。

構文

VTRANSCODEX (*var*)

必須引数

var

文字変数名に対して評価する SAS 文字式を指定します。

制限事項 指定された式の値で配列参照を示すことはできません。

詳細

VTRANSCODEX 関数は、トランスコーディングがオフの場合は 0 を、オンの場合は 1 を返します。

デフォルトでは、DATA ステップのすべての文字変数がトランスコードされます。トランスコーディングをオフにするには、ATTRIB ステートメントの TRANSCODE=属性を使用します。

比較

- VTRANSCODE 関数は、指定された変数でトランスコーディングが有効かどうかを示す値を返します。一方、VTRANSCODEX 関数は、変数名を判別するために引数を評価します。その後、VTRANSCODEX 関数はその変数名に関連付けられたトランスコーディングステータス(オンまたはオフ)を返します。
- VTRANSCODE 関数は、引数として式を受け入れません。VTRANSCODEX 関数は式を受け入れますが、指定された式の値で配列参照を示すことはできません。
- 関連する関数が、変数名、種類、出力形式、長さなど他の変数属性の値を返します。リストについては、*SAS Viya 関数と CALL ルーチン: リファレンスの”変数情報”*の関数を参照してください。

例

ステートメント	結果
	----+----1----+
attrib x transcode = yes; attrib y transcode = no; rc1 = vtranscodex('y'); put rc1=;	rc1=0

関連項目:

関数:

- [“VTRANSCODE 関数” \(399 ページ\)](#)

ステートメント:

- [ATTRIB](#)

6 部

各国語サポート関連の入力形式

13 章	
各国語サポート関連の入力形式のディクショナリ	405

13 章

各国語サポート関連の入力形式の ディクショナリ

カテゴリ別の入力形式	407
ディクショナリ	414
\$CPTDW 入力形式	414
\$CPTWD 入力形式	415
EURO 入力形式	416
EUROX 入力形式	418
MINGUO 入力形式	419
NENGO 入力形式	421
NLDATE 入力形式	422
NLDATEW 入力形式	423
NLDATM 入力形式	424
NLDATMAP 入力形式	425
NLDATMW 入力形式	426
NLMNIAED 入力形式	427
NLMNIAUD 入力形式	428
NLMNIBGN 入力形式	429
NLMNIBRL 入力形式	430
NLMNICAD 入力形式	431
NLMNICHF 入力形式	432
NLMNICNY 入力形式	433
NLMNICZK 入力形式	434
NLMNIDKK 入力形式	435
NLMNIEEK 入力形式	436
NLMNIEGP 入力形式	437
NLMNIEUR 入力形式	438
NLMNIGBP 入力形式	439
NLMNIHKD 入力形式	440
NLMNIHRK 入力形式	441
NLMNIHUF 入力形式	442
NLMNIIDR 入力形式	443
NLMNIILS 入力形式	444
NLMNIINR 入力形式	445
NLMNIJPY 入力形式	446
NLMNIKRW 入力形式	447
NLMNILTL 入力形式	448
NLMNILVL 入力形式	449
NLMNIMOP 入力形式	450
NLMNIMXN 入力形式	451
NLMNIMYR 入力形式	452
NLMNINOK 入力形式	453
NLMNINZD 入力形式	454

NLMNIPLN 入力形式	455
NLMNIRUB 入力形式	456
NLMNISEK 入力形式	457
NLMNISGD 入力形式	458
NLMNITHB 入力形式	459
NLMNITRY 入力形式	460
NLMNITWD 入力形式	461
NLMNIUSD 入力形式	462
NLMNIZAR 入力形式	463
NLMNLAED 入力形式	464
NLMNLAUD 入力形式	465
NLMNLBGN 入力形式	466
NLMNLBRL 入力形式	467
NLMNLCAD 入力形式	468
NLMNLCHF 入力形式	469
NLMNLCNY 入力形式	470
NLMNLCZK 入力形式	471
NLMNLDKK 入力形式	472
NLMNLEEK 入力形式	473
NLMNLEGP 入力形式	474
NLMNLEUR 入力形式	475
NLMNLGBP 入力形式	476
NLMNLHKD 入力形式	477
NLMNLHRK 入力形式	478
NLMNLHUF 入力形式	479
NLMNLIDR 入力形式	480
NLMNLILS 入力形式	481
NLMNLINR 入力形式	482
NLMNLJPY 入力形式	483
NLMNLKRW 入力形式	484
NLMNLLTL 入力形式	485
NLMNLLVL 入力形式	486
NLMNLMOP 入力形式	487
NLMNLMXN 入力形式	488
NLMNLMYR 入力形式	489
NLMNLNOK 入力形式	490
NLMNLNzd 入力形式	491
NLMNLPLN 入力形式	492
NLMNLRUB 入力形式	493
NLMNLSEK 入力形式	494
NLMNLSGD 入力形式	495
NLMNLTHB 入力形式	496
NLMNLTRY 入力形式	497
NLMNLTWD 入力形式	498
NLMNLUSD 入力形式	499
NLMNLZAR 入力形式	500
NLMNY 入力形式	501
NLMNYI 入力形式	502
NLNUM 入力形式	503
NLNUMI 入力形式	505
NLPCT 入力形式	506
NLPCTI 入力形式	507
NLTIMAP 入力形式	508
NLTIME 入力形式	509
\$UCS2B 入力形式	510
\$UCS2BE 入力形式	511

\$UCS2L 入力形式	512
\$UCS2LE 入力形式	513
\$UCS2X 入力形式	514
\$UCS2XE 入力形式	516
\$UCS4B 入力形式	517
\$UCS4L 入力形式	518
\$UCS4X 入力形式	519
\$UCS4XE 入力形式	520
\$UESC 入力形式	521
\$UESCE 入力形式	522
\$UNCR 入力形式	523
\$UNCRE 入力形式	524
\$UPAREN 入力形式	525
\$UPARENE 入力形式	527
\$UPARENPN 入力形式	528
\$UTF8X 入力形式	529

カテゴリ別の入力形式

各国語サポート関連の入力形式には、次の 6 つのカテゴリがあります。

カテゴリ	説明
BIDI テキストの操作	データ変数から双方向のデータ値を読み込むように指示します。
文字	文字変数に文字データ値を読み込むように指示します。
DBCS	さまざまなアジア言語を管理するように指示します。
日付と時間	日付、時刻および日時を表す変数にデータ値を読み込むように指示します。
ヘブライ語のテキストの操作	データ変数からヘブライ語のデータを読み込むように指示します。
数値	数値変数に数値データ値を読み込むように指示します。

次の表に、SAS 入力形式の概要を示します。詳細については、それぞれの各国語サポート関連の入力形式に関する情報を参照してください。

カテゴリ	言語要素	説明
数値	EURO 入力形式 (p. 416)	数値を読み込み、ヨーロッパ通貨の埋め込み文字を取り除いて、カンマと小数点を逆にします。
	EUROX 入力形式 (p. 418)	数値を読み込み、ヨーロッパ通貨の埋め込み文字を削除します。

カテゴリ	言語要素	説明
	NLMNIAED 入力形式 (p. 427)	アラブ首長国連邦の国際表示形式で通貨データを読み込みます。
	NLMNIAUD 入力形式 (p. 428)	オーストラリアの国際表示形式で通貨データを読み込みます。
	NLMNIBGN 入力形式 (p. 429)	ブルガリアの国際表示形式で通貨データを読み込みます。
	NLMNIBRL 入力形式 (p. 430)	ブラジルの国際表示形式で通貨データを読み込みます。
	NLMNICAD 入力形式 (p. 431)	カナダの国際表示形式で通貨データを読み込みます。
	NLMNICHF 入力形式 (p. 432)	リヒテンシュタインとスイスの国際表示形式で通貨データを読み込みます。
	NLMNICNY 入力形式 (p. 433)	中国の国際表示形式で通貨データを読み込みます。
	NLMNICZK 入力形式 (p. 434)	チェコ共和国の国際表示形式で通貨データを読み込みます。
	NLMNIDKK 入力形式 (p. 435)	デンマーク、フェロー諸島およびグリーンランドの国際表示形式で通貨データを読み込みます。
	NLMNIEEK 入力形式 (p. 436)	エストニアの国際表示形式で通貨データを読み込みます。
	NLMNIEGP 入力形式 (p. 437)	エジプトの国際表示形式で通貨データを読み込みます。
	NLMNIEUR 入力形式 (p. 438)	オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインの国際表示形式で通貨データを読み込みます。
	NLMNIGBP 入力形式 (p. 439)	英国の国際表示形式で通貨データを読み込みます。
	NLMNIHKD 入力形式 (p. 440)	香港の国際表示形式で通貨データを読み込みます。
	NLMNIHRK 入力形式 (p. 441)	クロアチアの国際表示形式で通貨データを読み込みます。
	NLMNIHUF 入力形式 (p. 442)	ハンガリーの国際表示形式で通貨データを読み込みます。
	NLMNIIDR 入力形式 (p. 443)	インドネシアの国際表示形式で通貨データを読み込みます。
	NLMNIILS 入力形式 (p. 444)	イスラエルの国際表示形式で通貨データを読み込みます。

カテゴリ	言語要素	説明
	NLMNIINR 入力形式 (p. 445)	インドの国際表示形式で通貨データを読み込みます。
	NLMNIJPY 入力形式 (p. 446)	日本の国際表示形式で通貨データを読み込みます。
	NLMNIKRW 入力形式 (p. 447)	韓国の国際表示形式で通貨データを読み込みます。
	NLMNILTL 入力形式 (p. 448)	リトアニアの国際表示形式で通貨データを読み込みます。
	NLMNILVL 入力形式 (p. 449)	ラトビアの国際表示形式で通貨データを読み込みます。
	NLMNIMOP 入力形式 (p. 450)	マカオの国際表示形式で通貨データを読み込みます。
	NLMNIMXN 入力形式 (p. 451)	メキシコの国際表示形式で通貨データを読み込みます。
	NLMNIMYR 入力形式 (p. 452)	マレーシアの国際表示形式で通貨データを読み込みます。
	NLMNINOK 入力形式 (p. 453)	ノルウェーの国際表示形式で通貨データを読み込みます。
	NLMNINZD 入力形式 (p. 454)	ニュージーランドの国際表示形式で通貨データを読み込みます。
	NLMNIPLN 入力形式 (p. 455)	ポーランドの国際表示形式で通貨データを読み込みます。
	NLMNIRUB 入力形式 (p. 456)	ロシアの国際表示形式で通貨データを読み込みます。
	NLMNISEK 入力形式 (p. 457)	スウェーデンの国際表示形式で通貨データを読み込みます。
	NLMNISGD 入力形式 (p. 458)	シンガポールの国際表示形式で通貨データを読み込みます。
	NLMNITHB 入力形式 (p. 459)	タイの国際表示形式で通貨データを読み込みます。
	NLMNITRY 入力形式 (p. 460)	トルコの国際表示形式で通貨データを読み込みます。
	NLMNITWD 入力形式 (p. 461)	台湾の国際表示形式で通貨データを読み込みます。
	NLMNIUSD 入力形式 (p. 462)	プエルトリコと米国の国際表示形式で通貨データを読み込みます。
	NLMNIZAR 入力形式 (p. 463)	南アフリカの国際表示形式で通貨データを読み込みます。
	NLMNLAED 入力形式 (p. 464)	アラブ首長国連邦のローカル表示形式で通貨データを読み込みます。

カテゴリ	言語要素	説明
	NLMNLAUD 入力形式 (p. 465)	オーストラリアのローカル表示形式で通貨データを読み込みます。
	NLMNLBGN 入力形式 (p. 466)	ブルガリアのローカル表示形式で通貨データを読み込みます。
	NLMNLBRL 入力形式 (p. 467)	ブラジルのローカル表示形式で通貨データを読み込みます。
	NLMNLCAD 入力形式 (p. 468)	カナダのローカル表示形式で通貨データを読み込みます。
	NLMNLCHF 入力形式 (p. 469)	リヒテンシュタインとスイスのローカル表示形式で通貨データを読み込みます。
	NLMNLCNY 入力形式 (p. 470)	中国のローカル表示形式で通貨データを読み込みます。
	NLMNLCZK 入力形式 (p. 471)	チェコ共和国のローカル表示形式で通貨データを読み込みます。
	NLMNLDDK 入力形式 (p. 472)	デンマーク、フェロー諸島およびグリーンランドのローカル表示形式で通貨データを読み込みます。
	NLMNLEEK 入力形式 (p. 473)	エストニアのローカル表示形式で通貨データを読み込みます。
	NLMNLEGP 入力形式 (p. 474)	エジプトのローカル表示形式で通貨データを読み込みます。
	NLMNLEUR 入力形式 (p. 475)	オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインのローカル表示形式で通貨データを読み込みます。
	NLMNLGBP 入力形式 (p. 476)	英国のローカル表示形式で通貨データを読み込みます。
	NLMNLHKD 入力形式 (p. 477)	香港のローカル表示形式で通貨データを読み込みます。
	NLMNLHRK 入力形式 (p. 478)	クロアチアのローカル表示形式で通貨データを読み込みます。
	NLMNLHUF 入力形式 (p. 479)	ハンガリーのローカル表示形式で通貨データを読み込みます。
	NLMNLIDR 入力形式 (p. 480)	インドネシアのローカル表示形式で通貨データを読み込みます。
	NLMNLILS 入力形式 (p. 481)	イスラエルのローカル表示形式で通貨データを読み込みます。

カテゴリ	言語要素	説明
	NLMNLINR 入力形式 (p. 482)	インドのローカル表示形式で通貨データを読み込みます。
	NLMNLJPY 入力形式 (p. 483)	日本のローカル表示形式で通貨データを読み込みます。
	NLMNLKRW 入力形式 (p. 484)	韓国のローカル表示形式で通貨データを読み込みます。
	NLMNLLTL 入力形式 (p. 485)	リトアニアのローカル表示形式で通貨データを読み込みます。
	NLMNLLVL 入力形式 (p. 486)	ラトビアのローカル表示形式で通貨データを読み込みます。
	NLMNLMOP 入力形式 (p. 487)	マカオのローカル表示形式で通貨データを読み込みます。
	NLMNLMXN 入力形式 (p. 488)	メキシコのローカル表示形式で通貨データを読み込みます。
	NLMNLMYR 入力形式 (p. 489)	マレーシアのローカル表示形式で通貨データを読み込みます。
	NLMNLNOK 入力形式 (p. 490)	ノルウェーのローカル表示形式で通貨データを読み込みます。
	NLMNLNZD 入力形式 (p. 491)	ニュージーランドのローカル表示形式で通貨データを読み込みます。
	NLMNLPLN 入力形式 (p. 492)	ポーランドのローカル表示形式で通貨データを読み込みます。
	NLMNLRUB 入力形式 (p. 493)	ロシアのローカル表示形式で通貨データを読み込みます。
	NLMNLSEK 入力形式 (p. 494)	スウェーデンのローカル表示形式で通貨データを読み込みます。
	NLMNLSGD 入力形式 (p. 495)	シンガポールのローカル表示形式で通貨データを読み込みます。
	NLMNLTHB 入力形式 (p. 496)	タイのローカル表示形式で通貨データを読み込みます。
	NLMNLTRY 入力形式 (p. 497)	トルコのローカル表示形式で通貨データを読み込みます。
	NLMNLTWD 入力形式 (p. 498)	台湾のローカル表示形式で通貨データを読み込みます。
	NLMNLUSD 入力形式 (p. 499)	プエルトリコと米国のローカル表示形式で通貨データを読み込みます。

カテゴリ	言語要素	説明
	NLMNLZAR 入力形式 (p. 500)	南アフリカのローカル表示形式で通貨データを読み込みます。
	NLMNY 入力形式 (p. 501)	指定されたロケールで、通貨データをローカル表示形式で読み込み、そのデータを数値に変換します。
	NLMNYI 入力形式 (p. 502)	指定ロケールで、通貨データを国際表示形式で読み込み、そのデータを数値に変換します。
	NLNUM 入力形式 (p. 503)	指定されたロケールで、数値データをローカル表示形式で読み込み、そのデータを数値に変換します。
	NLNUMI 入力形式 (p. 505)	指定されたロケールで、数値データを国際表示形式で読み込み、そのデータを数値に変換します。
	NLPCT 入力形式 (p. 506)	指定されたロケールで、百分率データをローカル表示形式で読み込み、そのデータを数値に変換します。
	NLPCTI 入力形式 (p. 507)	指定されたロケールで百分率データを国際表示形式で読み込み、そのデータを数値に変換します。
日付と時間	MINGUO 入力形式 (p. 419)	台湾の形式の日付を読み込みます。
	NENGO 入力形式 (p. 421)	eyymmdd 形式の日本の日付値を読み込みます。
	NLDATE 入力形式 (p. 422)	指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 日付値に変換します。
	NLDATEW 入力形式 (p. 423)	指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 日付および曜日に変換します。
	NLDATM 入力形式 (p. 424)	指定されたロケールの日時値を読み込み、その日時値をローカルの SAS 日時値に変換します。
	NLDATMAP 入力形式 (p. 425)	指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 日付値(午前または午後の指定付き)に変換します。
	NLDATMW 入力形式 (p. 426)	指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 曜日および日時に変換します。
	NLTIMAP 入力形式 (p. 508)	指定されたロケールの時間値を a.m.および p.m.表示を使用して読み込み、その時間値をローカルの SAS 時間値に変換します。
	NLTIME 入力形式 (p. 509)	指定されたロケールの時間値を読み込み、その時間値をローカルの SAS 時間値に変換します。
ヘブライ語のテキストの操作	\$CPTDW 入力形式 (p. 414)	ヘブライ語 DOS (cp862)エンコーディングの文字列を読み込み、その文字列を Windows (cp1255)エンコーディングに変換します。

カテゴリ	言語要素	説明
	\$CPTWD 入力形式 (p. 415)	Windows (cp1255)エンコーディングの文字列を読み込み、その文字列をヘブライ語 DOS (cp862)エンコーディングに変換します。
文字	\$UCS2B 入力形式 (p. 510)	ビッグエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UCS2BE 入力形式 (p. 511)	現在の SAS セッションのエンコーディングの文字列を読み込み、その文字列をビッグエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)に変換します。
	\$UCS2L 入力形式 (p. 512)	リトルエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UCS2LE 入力形式 (p. 513)	現在の SAS セッションのエンコーディングにある文字列を読み込み、その文字列をリトルエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)に変換します。
	\$UCS2X 入力形式 (p. 514)	16 ビット、UCS2 形式の UNICODE(ユニコード)文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UCS2XE 入力形式 (p. 516)	現在の SAS セッションのエンコーディングの文字列を読み込み、その文字列を、16 ビット、UCS2 形式の UNICODE(ユニコード)に変換します。
	\$UCS4B 入力形式 (p. 517)	ビッグエンディアン、32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UCS4L 入力形式 (p. 518)	リトルエンディアン、32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UCS4X 入力形式 (p. 519)	32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UCS4XE 入力形式 (p. 520)	現在の SAS セッションのエンコーディングの文字列を読み込み、その文字列を、32 ビット、UCS4 形式の UNICODE(ユニコード)に変換します。
	\$UESC 入力形式 (p. 521)	UESC 表記でエンコードされた文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UESCE 入力形式 (p. 522)	現在の SAS セッションのエンコーディングを使用する文字列を読み取り、その文字列を UESC 表記に変換します。
	\$UNCR 入力形式 (p. 523)	NCR 文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ	言語要素	説明
	\$UNCRE 入力形式 (p. 524)	現在の SAS セッションのエンコーディングを使用する文字列を読み込み、その文字列を NCR に変換します。
	\$UPAREN 入力形式 (p. 525)	UPAREN 表記でエンコードされた文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。
	\$UPARENE 入力形式 (p. 527)	現在の SAS セッションのエンコーディングを使用する文字列を読み込み、その文字列を UPAREN 表記に変換します。
	\$UPARENP 入力形式 (p. 528)	UPAREN 表記でエンコードされた文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換しますが、各国語文字は UPAREN 表記のエンコーディングのまま残ります。
	\$UTF8X 入力形式 (p. 529)	UTF-8 でエンコードされた文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。

ディクショナリ

\$CPTDW 入力形式

ヘブライ語 DOS (cp862)エンコーディングの文字列を読み込み、その文字列を Windows (cp1255)エンコーディングに変換します。

カテゴリ: ヘブライ語のテキストの操作

構文

\$CPTDW*w*.

説明

w

入力幅を指定します。

デフォルト 200

範囲 1-32767

比較

\$CPTDW*w*.入力形式では、\$CPTWD*w*.入力形式とは逆の処理を行います。

例

次の例では、入力値として 808182 を使用します。

ステートメント	結果
	----+----1----+
x=input('808182',\$cptdw6.); put x;	1108

関連項目:

出力形式:

- “\$CPTDW 出力形式” (88 ページ)
- “\$CPTWD 出力形式” (89 ページ)

入力形式:

- “\$CPTWD 入力形式” (415 ページ)

\$CPTWD 入力形式

Windows (cp1255)エンコーディングの文字列を読み込み、その文字列をヘブライ語 DOS (cp862)エンコーディングに変換します。

カテゴリ: ヘブライ語のテキストの操作

構文

\$CPTWD w .

説明

w

入力幅を指定します。

デフォルト 200

範囲 1-32767

比較

\$CPTWD w .入力形式では、\$CPTDW w .入力形式とは逆の処理を行います。

例

次の例では、入力値として1108を使用します。

ステートメント	結果
	----+----1----+

ステートメント	結果
<code>x=input(' €□, \$cptwd6.); put x;</code>	€□,

関連項目:**出力形式:**

- [“\\$CPTWD 出力形式” \(89 ページ\)](#)
- [“\\$CPTDW 出力形式” \(88 ページ\)](#)

入力形式:

- [“\\$CPTDW 入力形式” \(414 ページ\)](#)

EURO 入力形式

数値を読み込み、ヨーロッパ通貨の埋め込み文字を取り除いて、カンマと小数点を逆にします。

カテゴリ: 数値

構文

EURO*w.d*

説明

w

入力幅を指定します。

デフォルト 6

範囲 1-32

d

値を除算する 10 のべき乗を指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-31

詳細

EURO*w.d* 入力形式では数値を読み込み、埋め込みユーロ記号(E)、カンマ、空白、パーセント記号、ハイフン、閉じかっこを入力データから削除します。小数点は、実数部と小数部を分ける区切り文字と見なされます。EURO*w.d* 入力形式では、フィールドの先頭にある開きかっこがマイナス記号に変換されます。

比較

- EUROW.d 入力形式は EUROXw.d 入力形式に似ていますが、EUROXw.d 入力形式は小数点とカンマの役割が逆になっています。この規則はヨーロッパの国で共通です。
- 入力にカンマもピリオドもない場合、EUROW.d 入力形式と EUROXw.d 入力形式は相互に交換可能です。

例

次の表に、ユーロでの通貨の入力値、適用される SAS ステートメントおよび結果を示します。

```
data _null_;
  input x euro10.;
  put x=;
  datalines;
E1
E1.23
1.23
1,234.56
;
run;
SAS Log:
x=1
x=1.23
x=1.23
x=1234.56
```

値	ステートメント	結果
		----+----1----2
E1	input x euro10.; put x;	1
E1.23	input x euro10.; put x;	1.23
1.23	input x euro10.; put x;	1.23
1,234.56	input x euro10.; put x;	1234.56

関連項目:

出力形式:

- [“EURO 出力形式” \(95 ページ\)](#)
- [“EUROX 出力形式” \(98 ページ\)](#)

入力形式:

- [“EUROX 入力形式” \(418 ページ\)](#)

EUROX 入力形式

数値を読み込み、ヨーロッパ通貨の埋め込み文字を削除します。

カテゴリ: 数値

構文

EUROX $w.d$

説明

w

入力幅を指定します。

デフォルト 6

範囲 1-32

d

値を除算する 10 のべき乗を指定します。データに小数点を表すカンマが含まれる場合、 d 値は無視されます。

デフォルト 0

範囲 0-31

詳細

EUROX $w.d$ 入力形式では数値を読み込み、埋め込みユーロ記号(E)、ピリオド、空白、パーセント記号、ハイフン、閉じかっこを入力データから削除します。カンマは、実数部と小数部を分ける区切り文字と見なされます。EUROX $w.d$ 入力形式では、フィールドの先頭にある開きかっこがマイナス記号に変換されます。

比較

- EUROX $w.d$ 入力形式は EUROW $.d$ 入力形式に似ていますが、EUROW $.d$ 入力形式では、カンマと小数点の役割が逆になっています。この規則は英語圏の国で共通です。
- 入力にカンマもピリオドもない場合、EUROX $w.d$ 入力形式と EUROW $.d$ 入力形式は相互に交換可能です。

例

次の表に、ユーロでの通貨の入力値、適用される SAS ステートメントおよび結果を示します。

```
data _null_;
  input x eurox10.;
  put x=;
```



```

datalines;
E1
E1.23
1.23
1,234.56
; run;
SAS Log:
7  input x eurox10.;
8  put x=;
9  datalines;
x=1
x=123
x=123
x=1.23456

```

値	ステートメント	結果
		----+----1----2
E1	input x eurox10.; put x;	1
E1.23	input x eurox10.; put x;	123
1.23	input x eurox10.; put x;	123
1,234.56	input x eurox10.; put x;	1.23456

関連項目:

出力形式:

- [“EURO 出力形式” \(95 ページ\)](#)
- [“EUROX 出力形式” \(98 ページ\)](#)

入力形式:

- [“EURO 入力形式” \(416 ページ\)](#)

MINGUO 入力形式

台湾の形式の日付を読み込みます。

カテゴリ: 日付と時間

構文

MINGUO*w*.

説明

w

入力幅を指定します。

デフォルト 6

範囲 6-10

詳細

台湾の日付の通常の形式は、*yyyymmdd* です。

yyyy

年を表す整数です。

mm

月を表す 01 から 12 の整数です。

dd

月の日を表す 01 から 31 の整数です。

台湾のカレンダーでは、1912 年を基準年(1912 年 1 月 1 日が 01/01/01)として使用します。1912 年より前の日付は無効です。年の値は 100 年周期で繰り返されることはありません。年の値は、継続して増加していきます。

年、月、日の値は、*YMMDDw* 入力形式で使用可能な、空白、スラッシュ、ハイフンなどの区切り文字で区切ることができます。区切り文字を使用する場合は、すべての値の間に挿入します。区切り文字を省略する場合は、値が 10 未満の日または月の先頭に 0 を使用してください。

例

次の例では、入力値に異なる日付を使用します。

```
input date minguo10;
put date date9;

data _null_;
  input date minguo10;
  put date date9;
datalines;
49/01/01
891215
03-01-01
;
```

値	結果
	----+----1----+
49/01/01	01JAN1960
891215	15DEC2000

値	結果
103-01-01	01JAN2014

関連項目:

出力形式:

- [“MINGUO 出力形式” \(100 ページ\)](#)

入力形式:

- [“YYMMDD Informat” \(SAS Viya Formats and Informats: Reference\)](#)

NENGO 入力形式

`eyymmdd` 形式の日本の日付値を読み込みます。

カテゴリ: 日付と時間

構文

NENGO*w*.

説明

w

入力幅を指定します。

デフォルト 10

範囲 7-32

詳細

日本の日付の通常の形式は、`eyymmdd` です。

e

元号(明治、大正、昭和、平成)の先頭の文字(M、T、S、H)です。

yy

年を表す整数です。

mm

月を表す 01 から 12 の整数です。

dd

月の日を表す 01 から 31 の整数です。

e の値と整数はピリオドで区切ることができます。**e** を省略すると、現在の元号が使用されます。年、月、日の値は、空白や数値以外の文字で区切ることができます。ただし、区切り文字を使用する場合は、すべての値の間に挿入します。区切り文字を省略する場合は、値が 10 未満の日または月の先頭に 0 を使用してください。

例

次の例では、異なる入力値を使用します。

```
input nengo_date nengo8.;
put nengo_date date9.;

data _null_;
  input nengo_date nengo8.;
  put nengo_date date9.;
  put nengo_date= ;
  datalines;
h11108
h.11108
11/10/08
;
```

値	結果
	----t----1----+
h11108	08OCT1999
h.11108	08OCT1999
11/10/08	08OCT1999

関連項目:

出力形式:

- [“NENGO 出力形式” \(102 ページ\)](#)

NLDATE 入力形式

指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 日付値に変換します。

カテゴリ: 日付と時間

別名: NLDATEW

構文

NLDATE*w*.

説明

w

入力幅を指定します。

デフォルト 10

範囲 10-200

例

次の例では、入力値として February 24, 2003 を使用します。

ステートメント	結果
	----+----1----+
<pre>options locale=English_UnitedStates; dy='February 24, 2003'; y=input(dy,nldate200.); put y=;</pre>	15760
<pre>options locale=German_Germany; dy='24. Februar 2003'; y=input(dy,nldate16.); put y=;</pre>	15760

関連項目:

出力形式:

- [“NLDATE 出力形式” \(105 ページ\)](#)

NLDATEW 入力形式

指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 日付および曜日に変換します。

カテゴリ: 日付と時間

配置: 左

構文

NLDATEW w .

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 20

範囲 16-200

例

次の例では、入力値として February 24, 2014 を使用します。

ステートメント	結果
	----+----1----+
options locale=English_UnitedStates; data; dy='Monday, February 24, 2014'; y=input(dy,nldatew200.); put y=; run;	19778
options locale=German_Germany; dy='Mo. 24.Februar 2014'; y=input(dy,nldatew16.); put y=; run;	19778

NLDATM 入力形式

指定されたロケールの日時値を読み込み、その日時値をローカルの SAS 日時値に変換します。

カテゴリ: 日付と時間

別名: NLDATMW、NLDATMAP

構文

NLDATM*w*.

説明

w

入力幅を指定します。

デフォルト 19

範囲 19-200

例

次の例では、入力値として February 24, 2003 12:39:43 を使用します。

ステートメント	結果
	----+----1----+

ステートメント	結果
<pre>options locale=English_UnitedStates; y=input('24.Feb03:12:39:43', nldatm.); put y=;</pre>	1361709583
<pre>options locale=German_Germany; y=input('24.Februar 2003 12.39 Uhr', nldatm.); put y=;</pre>	1330171200

関連項目:

出力形式:

- [“NLDATM 出力形式” \(126 ページ\)](#)

NLDATMAP 入力形式

指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 日付値(午前または午後の指定付き)に変換します。

カテゴリ: 日付と時間

配置: 左

構文

NLDATMAP*w*.

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日時値が短縮されます。

デフォルト 32

範囲 17-200

例

これらの例では、入力値として February 24, 2014 の 12:39:43 p.m.を使用します。

ステートメント	結果
	----+----1----+----2----+----3

ステートメント	結果
<pre>options locale=English_UnitedStates; data; dy='February 24, 2014 12:39:43 PM'; y=input(dy,nldatmap200.); put y=; run;</pre>	1708864783
<pre>options locale=Spanish_Mexico; data; dy='24/02/2003 12:39:43 PM'; y=input(dy,nldatmap200.); put y=; run;</pre>	1708864783

NLDATMW 入力形式

指定されたロケールの日付値を読み込み、その日付値をローカルの SAS 曜日および日時に変換します。

カテゴリ: 日付と時間

配置: 左

構文

NLDATMW_w

説明

w

出力フィールドの幅を指定します。必要に応じて、出力形式の幅に合うように日付が短縮されます。

デフォルト 40

範囲 34-200

例

次の例では、入力値として Mon, Feb 24, 2014 12:39:43PM を使用します。

ステートメント	結果
	----+----1----+
<pre>options locale=English_UnitedStates; data; dy='Mon, Feb 24, 2014 12:39:43 PM'; y=input(dy,nldatmw200.); put y=; run;</pre>	1708864783

ステートメント	結果
<pre>options locale=German_Germany; dy='Mo, 24. Feb 2014 12.39 Uhr'; y=input(dy,nldatmw16.); put y; run;</pre>	1708864783

NLMNIAED 入力形式

アラブ首長国連邦の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIAED*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で数値を除算するよう指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('($12,345.67)',nlmniaed32.2);
y=input('($12,345.67)',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLAED 入力形式” \(464 ページ\)](#)

NLMNIAUD 入力形式

オーストラリアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIAUD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で数値を除算するよう指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmniaud32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLAUD 入力形式” \(465 ページ\)](#)

NLMNIBGN 入力形式

ブルガリアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIBGN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=put('(-1234.56789)',nlmnibgn32.2);
```

```
y=put('(-1234.56789)',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- “NLMLBGN 入力形式” (466 ページ)

NLMNIBRL 入力形式

ブラジルの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIBRL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnibr32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLBRL 入力形式” \(467 ページ\)](#)

NLMNICAD 入力形式

カナダの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNICAD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnicad32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNICAD 出力形式” \(155 ページ\)](#)

NLMNICHF 入力形式

リヒテンシュタインとスイスの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNICHF*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnicf32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNICHF 出力形式” \(156 ページ\)](#)

NLMNICNY 入力形式

中国の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNICNY $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnicny32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNICNY 出力形式” \(157 ページ\)](#)

NLMNICZK 入力形式

チェコ共和国の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNICZK $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmniczk32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLCZK 入力形式” \(471 ページ\)](#)

NLMNIDKK 入力形式

デンマーク、フェロー諸島およびグリーンランドの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIDKK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 912

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmndkk32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIDKK 出力形式” \(159 ページ\)](#)

NLMNIEEK 入力形式

エストニアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIEEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnieek32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLEEK 入力形式” \(473 ページ\)](#)

NLMNIEGP 入力形式

エジプトの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIEGP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmniegp32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLEGP 入力形式” \(474 ページ\)](#)

NLMNIEUR 入力形式

オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIEUR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnieur32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIEUR 出力形式” \(162 ページ\)](#)

NLMNIGBP 入力形式

英国の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIGBP $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnigbp32.2);
```

```
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIGBP 出力形式” \(163 ページ\)](#)

NLMNIHKD 入力形式

香港の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIHKD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnikd32.2);
y=input('$12,345.67'dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIHKD 出力形式” \(164 ページ\)](#)

NLMNIHRK 入力形式

クロアチアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIHRK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnihrk32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLHRK 入力形式” \(478 ページ\)](#)

NLMNIHUF 入力形式

ハンガリーの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIHUF*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnihof32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLHUF 入力形式” \(479 ページ\)](#)

NLMNIIDR 入力形式

インドネシアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIIDR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmniidr32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLIDR 入力形式” \(480 ページ\)](#)

NLMNIILS 入力形式

イスラエルの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIILS*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmniils32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIILS 出力形式” \(168 ページ\)](#)

NLMNIINR 入力形式

インドの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIINR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmniinr32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLINR 入力形式” \(482 ページ\)](#)

NLMNIJPY 入力形式

日本の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIJPY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmniipy32.2);
y=input('$12,345.67'dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIJPY 出力形式” \(170 ページ\)](#)

NLMNIKRW 入力形式

韓国の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIKRW*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnikrw32.2);
```

```
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- “NLMNILRW 入力形式” (484 ページ)

NLMNITL 入力形式

リトアニアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNITL $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnil32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLLTL 入力形式” \(485 ページ\)](#)

NLMNILVL 入力形式

ラトビアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNILVL $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnilvl32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLLVL 入力形式” \(486 ページ\)](#)

NLMNIMOP 入力形式

マカオの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIMOP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnimop32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLMOP 入力形式” \(487 ページ\)](#)

NLMNIMXN 入力形式

メキシコの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIMXN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnimxn32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLMXN 入力形式” \(488 ページ\)](#)

NLMNIMYR 入力形式

マレーシアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIMYR $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnyr32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIMYR 出力形式” \(176 ページ\)](#)

NLMNINOK 入力形式

ノルウェーの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNINOK $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmninok32.2);
```

```
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNINOK 出力形式” \(177 ページ\)](#)

NLMNINZD 入力形式

ニュージーランドの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNINZD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmninzd32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:

出力形式:

- [“NLMNINZD 出力形式” \(178 ページ\)](#)

NLMNIPLN 入力形式

ポーランドの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIPLN $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnipln32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIPLN 出力形式” \(179 ページ\)](#)

NLMNIRUB 入力形式

ロシアの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIRUB*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnirub32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIRUB 出力形式” \(180 ページ\)](#)

NLMNISEK 入力形式

スウェーデンの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNISEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnisek32.2);
```

```
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNISEK 出力形式” \(181 ページ\)](#)

NLMNISGD 入力形式

シンガポールの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNISGD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnisgd32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNISGD 出力形式” \(182 ページ\)](#)

NLMNITHB 入力形式

タイの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNITHB*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnithb32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLTHB 入力形式” \(496 ページ\)](#)

NLMNITRY 入力形式

トルコの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNITRY $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnitry32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNLTRY 入力形式” \(497 ページ\)](#)

NLMNITWD 入力形式

台湾の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNITWD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnitwd32.2);
```

```
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNITWD 出力形式” \(185 ページ\)](#)

NLMNIUSD 入力形式

プエルトリコと米国の国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIUSD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmniUSD32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIUSD 出力形式” \(186 ページ\)](#)

NLMNIZAR 入力形式

南アフリカの国際表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNIZAR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67'),nlmnizar32.2);
y=input('$12,345.67')dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNIZAR 出力形式” \(187 ページ\)](#)

NLMNLAED 入力形式

アラブ首長国連邦のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLAED*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlaid32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIAED 入力形式” \(427 ページ\)](#)

NLMNLAUD 入力形式

オーストラリアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLAUD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnloud32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLAUD 出力形式” \(189 ページ\)](#)

NLMNLBGN 入力形式

ブルガリアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLBGN $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input(-12345.67,nlmnlbgn32.2);
y=input(-12345.67,dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIBGN 入力形式” \(429 ページ\)](#)

NLMNLBRL 入力形式

ブラジルのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLBRL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlbrl32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIBRL 入力形式” \(430 ページ\)](#)

NLMNLCAD 入力形式

カナダのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLCAD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlcad32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- “NLMNLCAD 出力形式” (192 ページ)

NLMNLCHF 入力形式

リヒテンシュタインとスイスのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLCHF $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlchf32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLCHF 出力形式” \(193 ページ\)](#)

NLMNLCNY 入力形式

中国のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLCNY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlcny32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLCNY 出力形式” \(194 ページ\)](#)

NLMNLCZK 入力形式

チェコ共和国のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLCZK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlczk32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNICZK 入力形式” \(434 ページ\)](#)

NLMNLDKK 入力形式

デンマーク、フェロー諸島およびグリーンランドのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLDKK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmldkk32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- “NLMNLDKK 出力形式” (196 ページ)

NLMNLEEK 入力形式

エストニアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLEEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnleek32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIEEK 入力形式” \(436 ページ\)](#)

NLMNLEGP 入力形式

エジプトのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLEGP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 3

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmlegp32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIEGP 入力形式” \(437 ページ\)](#)

NLMNLEUR 入力形式

オーストリア、ベルギー、キプロス、エストニア、フィンランド、フランス、ドイツ、ギリシャ、アイルランド、イタリア、ラトビア、リトアニア、ルクセンブルク、マルタ、オランダ、ポルトガル、スロバキア、スロベニア、スペインのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLEUR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmneur32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLEUR 出力形式” \(199 ページ\)](#)

NLMNLGBP 入力形式

英国のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLGBP $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlgbp32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLGBP 出力形式” \(200 ページ\)](#)

NLMNLHKD 入力形式

香港のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLHKD_{w,d}

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlhkd32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLHKD 出力形式” \(201 ページ\)](#)

NLMNLHRK 入力形式

クロアチアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLHRK $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlhrk32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-1,234.57

関連項目:**入力形式:**

- [“NLMNIHRK 入力形式” \(441 ページ\)](#)

NLMNLHUF 入力形式

ハンガリーのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLHUF $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlhuf32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIHUF 入力形式” \(442 ページ\)](#)

NLMNLIDR 入力形式

インドネシアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLIDR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlidr32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIIDR 入力形式” \(443 ページ\)](#)

NLMNLILS 入力形式

イスラエルのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLILS $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlils32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLILS 出力形式” \(205 ページ\)](#)

NLMNLINR 入力形式

インドのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLINR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlir32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIINR 入力形式” \(445 ページ\)](#)

NLMNLJPY 入力形式

日本のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLJPY $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnljpy32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLJPY 出力形式” \(207 ページ\)](#)

NLMNLKRW 入力形式

韓国のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLKRW*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlkrw32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIKRW 入力形式” \(447 ページ\)](#)

NLMNLLTL 入力形式

リトアニアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLLTL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnl1t32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- “NLMNILTL 入力形式” (448 ページ)

NLMNLLVL 入力形式

ラトビアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLLVL*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlvl32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNILVL 入力形式” \(449 ページ\)](#)

NLMNLMOP 入力形式

マカオのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLMOP*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlmop32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIMOP 入力形式” \(450 ページ\)](#)

NLMNLMXN 入力形式

メキシコのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLMXN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlmxcn32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNIMXN 入力形式” \(451 ページ\)](#)

NLMNLMYR 入力形式

マレーシアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLMYR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlmyr32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLMYR 出力形式” \(213 ページ\)](#)

NLMNLNOK 入力形式

ノルウェーのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLNOK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlnok32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNZNOK 出力形式” \(214 ページ\)](#)

NLMNZNZD 入力形式

ニュージーランドのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNZNZD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlzdz32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLNZD 出力形式” \(215 ページ\)](#)

NLMNLPLN 入力形式

ポーランドのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLPLN*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlpln32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLRUB 出力形式” \(216 ページ\)](#)

NLMNLRUB 入力形式

ロシアのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLRUB $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlrub32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLRUB 出力形式” \(217 ページ\)](#)

NLMNLSEK 入力形式

スウェーデンのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLSEK*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlsek32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLSGD 出力形式” \(218 ページ\)](#)

NLMNLSGD 入力形式

シンガポールのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLSGD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlsgd32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- “NLMNLSGD 出力形式” (219 ページ)

NLMNLTHB 入力形式

タイのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLTHB*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlthb32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNITHB 入力形式” \(459 ページ\)](#)

NLMNLTRY 入力形式

トルコのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLTRY*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 4

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnltry32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**入力形式:**

- [“NLMNITRY 入力形式” \(460 ページ\)](#)

NLMNLTWD 入力形式

台湾のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLTWD*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnltd32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- “NLMNLUSD 出力形式” (222 ページ)

NLMNLUSD 入力形式

プエルトリコと米国のローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLUSD $w.d$

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、 d 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlusd32.2);
```

```
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLUSD 出力形式” \(223 ページ\)](#)

NLMNLZAR 入力形式

南アフリカのローカル表示形式で通貨データを読み込みます。

カテゴリ: 数値

配置: 左

構文

NLMNLZAR*w.d*

説明

w

出力フィールドの幅を指定します。

デフォルト 12

範囲 8-32

d

10^d で除算するかどうかを指定します。データに小数点が含まれる場合、*d* 値は無視されます。

デフォルト 2

範囲 0-28

例

次の例では、LOCALE=システムオプションが English_UnitedStates に設定されています。

```
x=input('$12,345.67',nlmnlzar32.2);
y=input('$12,345.67',dollar32.2);
```

ステートメント	結果
	----+----1----+
put x=;	-12345.67
put y=;	-12345.67

関連項目:**出力形式:**

- [“NLMNLZAR 出力形式” \(224 ページ\)](#)

NLMNY 入力形式

指定されたロケールで、通貨データをローカル表示形式で読み込み、そのデータを数値に変換します。

カテゴリ: 数値

構文

NLMNY*w.d*

説明

w

入力幅を指定します。

デフォルト 9

範囲 1-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-31

詳細

NLMNY*w.d* 入力形式では、指定されたロケールで、通貨データをローカル表示形式で読み込み、そのデータを数値に変換します。入力データから、桁区切り文字、小数点区切り文字、空白、通貨記号、閉じかっこを削除します。

比較

NLMNY*w.d* 入力形式では、NLMNYI*w.d* 入力形式とは逆の処理を行います。

NLMNY*w.d* 入力形式は DOLLAR*w.d* 入力形式と似ていますが、NLMNY*w.d* 入力形式がロケール固有である点が違います。

数値データをアポストロフィで囲むと、NLNUM はエラーメッセージを返します。アポストロフィはデータが文字であることを指定します。次の例では、数値 1 がアポストロフィで囲まれているため、エラーメッセージが生成されます。

```
data;
x=input("'1'",NLMNY32.);
put x;
run;
```

例

次の例では、入力値として\$12,345.67を使用します。

ステートメント	結果
	----+----1----+
options LOCALE=English_UnitedStates;	-12345.67
x=input('\$12,345.67'),nlmny32.2);	-12345.67
y=input('\$12,345.67'),dollar32.2);	
put x=;	
put y=;	

関連項目:

出力形式:

- [“NLMNY 出力形式” \(225 ページ\)](#)
- [“NLMNYI 出力形式” \(226 ページ\)](#)

入力形式:

- [“NLMNYI 入力形式” \(502 ページ\)](#)

NLMNYI 入力形式

指定ロケールで、通貨データを国際表示形式で読み込み、そのデータを数値に変換します。

カテゴリ: 数値

構文

NLMNYI*w.d*

説明

w

入力幅を指定します。

デフォルト 9

範囲 1-32

d

10^dで除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d*値は無視されます。

デフォルト 0

範囲 0-31

詳細

NLMNYI*w.d* 入力形式では、指定ロケールで、通貨データを国際表示形式で読み込み、そのデータを数値に変換します。入力データから、桁区切り文字、小数点区切り文字、空白、通貨記号、閉じかっこを削除します。

比較

NLMNYI*w.d* 入力形式では、NLMNY*w.d* 入力形式とは逆の処理を行います。

例

次の例では、入力値として 12,345.67 を使用します。

ステートメント	結果
	----+----1----+
options LOCALE=English_UnitedStates;	-12345.67
x=input('(USD12,345.67)',nlmnyi32.2);	-12345.67
y=input('\$-12,345.67)',dollar32.2);	
put x=;	
put y=;	

関連項目:

出力形式:

- [“NLMNY 出力形式” \(225 ページ\)](#)
- [“NLMNYI 出力形式” \(226 ページ\)](#)

入力形式:

- [“NLMNY 入力形式” \(501 ページ\)](#)

NLNUM 入力形式

指定されたロケールで、数値データをローカル表示形式で読み込み、そのデータを数値に変換します。

カテゴリ: 数値

構文

NLNUM*w.d*

説明

w

入力幅を指定します。

デフォルト 6
 範囲 1-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d* 値は無視されます。

デフォルト 0
 範囲 0-31

詳細

NLNUM*w.d* 入力形式では、指定されたロケールで、数値データをローカル表示形式で読み込み、そのデータを数値に変換します。入力データから、桁区切り文字、小数点区切り文字、空白、通貨記号、閉じかっこを削除します。

数値データをアポストロフィで囲むと、NLNUM はエラーメッセージを返します。アポストロフィはデータが文字であることを指定します。次の例では、数値 1 がアポストロフィで囲まれているため、エラーメッセージが生成されます。

```
data;
x=input("'1'",NLNUM32.);
put x=;
run;
```

比較

NLNUM*w.d* 入力形式では、NLNUMI*w.d* 入力形式とは逆の処理を行います。

例

次の例では、入力値として-1234356.78 を使用します。

ステートメント	結果
	----+----1----+
options locale=English_UnitedStates; x=input('-1,234,356.78',nlNum32.2); put x=;	-1234356.78

関連項目:**出力形式:**

- [“NLNUM 出力形式” \(227 ページ\)](#)
- [“NLMNYI 出力形式” \(226 ページ\)](#)

入力形式:

- [“NLNUMI 入力形式” \(505 ページ\)](#)

NLNUMI 入力形式

指定されたロケールで、数値データを国際表示形式で読み込み、そのデータを数値に変換します。

カテゴリ: 数値

構文

NLNUMI*w.d*

説明

w

入力幅を指定します。

デフォルト 6

範囲 1-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-31

詳細

NLNUMI*w.d* 入力形式では、指定されたロケールで、数値データを国際表示形式で読み込み、そのデータを数値に変換します。入力データから、桁区切り文字、小数点区切り文字、空白、通貨記号、閉じかっこを削除します。

比較

NLNUMI*w.d* 入力形式では、NLNUM*w.d* 入力形式とは逆の処理を行います。

例

次の例では、入力値として-1,234,356.78 を使用します。

ステートメント	結果
	----+----1----+
options locale=English_UnitedStates;	-1234356.78
x=input('-1,234,356.78', nlnumi32.2);	
put x=;	

関連項目:

出力形式:

- “NLNUM 出力形式” (227 ページ)
- “NLNUMI 出力形式” (229 ページ)
- “NLNUM 入力形式” (503 ページ)

NLPCT 入力形式

指定されたロケールで、百分率データをローカル表示形式で読み込み、そのデータを数値に変換します。

カテゴリ: 数値

構文

`NLPCT $w.d$`

説明

w

入力幅を指定します。

デフォルト 6

範囲 1-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、 d 値は無視されます。

デフォルト 0

範囲 0-31

詳細

NLPCT $w.d$ 入力形式では、指定されたロケールで百分率データをローカル表示形式で読み込み、そのデータを数値に変換します。値を 100 で除算し、桁区切り文字、小数点区切り文字、空白、パーセント記号、閉じかっこを入力データから削除します。

比較

NLPCT $w.d$ 入力形式では、NLPCTI $w.d$ 入力形式とは逆の処理を行います。NLPCT $w.d$ 入力形式は PERCENT $w.d$ 入力形式と似ていますが、NLPCT $w.d$ 入力形式がロケール固有である点が違います。

数値データをアポストロフィで囲むと、NLPCT はエラーメッセージを返します。アポストロフィはデータが文字であることを指定します。次の例では、数値 1 がアポストロフィで囲まれているため、エラーメッセージが生成されます。

```
data;
x=input("'1'",NLPCT32.);
```



```
put x=;
run;
```

例

次の例では、入力値として-12,345.67%を使用します。

ステートメント	結果
	----+----1----+
options LOCALE=English_UnitedStates;	-123.4567
x=input('-12,345.67%',nlpct32.2);	-123.4567
y=input('(12,345.67%),percent32.2);	
put x=;	
put y=;	

関連項目:

出力形式:

- [“NLPCT 出力形式” \(230 ページ\)](#)
- [“NLPCTI 出力形式” \(232 ページ\)](#)
- [“NLPCTI 入力形式” \(507 ページ\)](#)

NLPCTI 入力形式

指定されたロケールで百分率データを国際表示形式で読み込み、そのデータを数値に変換します。

カテゴリ: 数値

構文

NLPCTI*w.d*

説明

w

入力幅を指定します。

デフォルト 6

範囲 1-32

d

10^d で除算するかどうかを指定します。データに小数点区切り文字が含まれる場合、*d* 値は無視されます。

デフォルト 0

範囲 0-31

詳細

NLPCTIw.d 入力形式では、指定されたロケールで、百分率データを国際表示形式で読み込み、そのデータを数値に変換します。値を 100 で除算し、桁区切り文字、小数点区切り文字、空白、パーセント記号、閉じかっこを入力データから削除します。

比較

NLPCTIw.d 入力形式では、NLPCTw.d 入力形式とは逆の処理を行います。

例

次の例では、入力値として-12,345.67%を使用します。

ステートメント	結果
	----+----1----+
options LOCALE=English_UnitedStates;	-123.4567
x=input('-12,345.67%',nlpct32.2);	-123.4567
y=input('(12,345.67%),percent32.2);	
put x=;	
put y=;	

関連項目:

出力形式:

- [“NLPCT 出力形式” \(230 ページ\)](#)
- [“NLPCTI 出力形式” \(232 ページ\)](#)

入力形式:

- [“NLPCT 入力形式” \(506 ページ\)](#)

NLTIMAP 入力形式

指定されたロケールの時間値を a.m.および p.m.表示を使用して読み込み、その時間値をローカルの SAS 時間値に変換します。

カテゴリ: 日付と時間

構文

NLTIMAPw.

説明

w 入力幅を指定します。

デフォルト 10

範囲 4-200

例

次の例では、入力値として 04:24:43 p.m.を使用します。

ステートメント	結果
	----+----1----+
options locale=English_UnitedStates; y=input('04:24:43 PM',nltimap11.); put y time.;	16:24:43
options locale=German_Germany; y=input('16.24 Uhr',nltimap11.); put y time.;	16:24:00

関連項目:

出力形式:

- [“NLTIMAP 出力形式” \(240 ページ\)](#)

NLTIME 入力形式

指定されたロケールの時間値を読み込み、その時間値をローカルの SAS 時間値に変換します。

カテゴリ: 日付と時間

別名: NLTIMAP

構文

NLTIME*w*.

説明

w 入力幅を指定します。

デフォルト 20

範囲 10-200

例

次の例では、入力値として 16:24:43 を使用します。

ステートメント	結果
	----+----1----+
options locale=English_UnitedStates; y=input('16:24:43',nltime.); put y time.;	16:24:43
options locale=German_Germany; y=input('16.24 Uhr',nltime.); put y time.;	16:24:00

関連項目:

出力形式:

- [“NLTIME 出力形式” \(241 ページ\)](#)

\$UCS2B 入力形式

ビッグエンディアン、16ビット、UCS2形式のUNICODE(ユニコード)文字列を読み取り、その文字列を現在のSASセッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8はSAS Viyaがサポートしている唯一のSASセッションのエンコーディングです。

構文

\$UCS2B*w*.

説明

w

入力幅を指定します。16ビットサイズのUNICODE文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

比較

\$UCS2Bw.入力形式では、\$UCS2BEw.入力形式とは逆の処理を行います。同一の動作環境内のデータを変換する場合は、\$UCS2Xw.入力形式を使用します。異なる動作環境からのデータを処理する場合は、\$UCS2Bw.入力形式や\$UCS2Lw.入力形式を使用します。

例

この例では UTF-8 エンコーティングを使用しています。

ステートメント	結果
	----+----1----+
x=input('5927'x,\$ucs2b.); put x=\$hex6.;	x=E5A4A7

関連項目:

出力形式:

- ["\\$UCS2B 出力形式" \(242 ページ\)](#)
- ["\\$UCS2L 出力形式" \(244 ページ\)](#)
- ["\\$UCS2X 出力形式" \(247 ページ\)](#)
- ["\\$UTF8X 出力形式" \(263 ページ\)](#)

入力形式:

- ["\\$UCS2L 入力形式" \(512 ページ\)](#)
- ["\\$UCS2X 入力形式" \(514 ページ\)](#)
- ["\\$UTF8X 入力形式" \(529 ページ\)](#)

\$UCS2BE 入力形式

現在の SAS セッションのエンコーディングの文字列を読み込み、その文字列をビッグエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)に変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS2BEw.

説明

w

入力幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

比較

\$UCS2BE*w*.入力形式では、\$UCS2B*w*.入力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
<code>ucs2str=input ('大', \$ucs2be4.);</code> <code>put ucs2str=\$hex8.;</code>	ucs2str=00205927

関連項目:

出力形式:

- ["\\$UCS2B 出力形式" \(242 ページ\)](#)
- ["\\$UCS2BE 出力形式" \(243 ページ\)](#)

入力形式:

- ["\\$UCS2B 入力形式" \(510 ページ\)](#)

\$UCS2L 入力形式

リトルエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS2L*w*.

説明

w

入力幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

比較

\$UCS2L*w*.入力形式では、\$UCS2LE*w*.入力形式とは逆の処理を行います。同一の動作環境内のデータを変換する場合は、\$UCS2X*w*.入力形式を使用します。異なる動作環境からのデータを処理する場合は、\$UCS2B*w*.入力形式や\$UCS2L*w*.入力形式を使用します。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
<pre>x=input('2759'x,\$ucs2l.); put x=\$hex6.;</pre>	x=E5A4A7

関連項目:

出力形式:

- ["\\$UCS2B 出力形式" \(242 ページ\)](#)
- ["\\$UCS2L 出力形式" \(244 ページ\)](#)
- ["\\$UCS2X 出力形式" \(247 ページ\)](#)
- ["\\$UTF8X 出力形式" \(263 ページ\)](#)

入力形式:

- ["\\$UCS2B 入力形式" \(510 ページ\)](#)
- ["\\$UCS2X 入力形式" \(514 ページ\)](#)
- ["\\$UTF8X 入力形式" \(529 ページ\)](#)

\$UCS2LE 入力形式

現在の SAS セッションのエンコーディングにある文字列を読み込み、その文字列をリトルエンディアン、16 ビット、UCS2 形式の UNICODE(ユニコード)に変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS2LEw.

説明

w

入力幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

比較

\$UCS2LEw.入力形式では、\$UCS2Lw.入力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
ucs2str=input ('大', \$ucs2le4.); put ucs2str=\$hex4;	ucs2str=2759

関連項目:

出力形式:

- ["\\$UCS2L 出力形式" \(244 ページ\)](#)
- ["\\$UCS2LE 出力形式" \(246 ページ\)](#)

入力形式:

- ["\\$UCS2L 入力形式" \(512 ページ\)](#)

\$UCS2X 入力形式

16 ビット、UCS2 形式の UNICODE(ユニコード)文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS2Xw.

説明

w

出力フィールドの幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

比較

\$UCS2Xw.入力形式では、\$UCS2XEw.入力形式とは逆の処理を行います。同一の動作環境内のデータを変換する場合は、\$UCS2Xw.入力形式を使用します。異なる動作環境からのデータを処理する場合は、\$UCS2Bw.入力形式や\$UCS2Lw.入力形式を使用します。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
x=input('5927'x,\$ucs2x.); put x=\$hex6.;	x=E29D99

関連項目:

出力形式:

- ["\\$UCS2B 出力形式" \(242 ページ\)](#)
- ["\\$UCS2L 出力形式" \(244 ページ\)](#)
- ["\\$UCS2X 出力形式" \(247 ページ\)](#)
- ["\\$UTF8X 出力形式" \(263 ページ\)](#)

入力形式:

- ["\\$UCS2B 入力形式" \(510 ページ\)](#)
- ["\\$UCS2L 入力形式" \(512 ページ\)](#)
- ["\\$UTF8X 入力形式" \(529 ページ\)](#)

\$UCS2XE 入力形式

現在の SAS セッションのエンコーディングの文字列を読み込み、その文字列を、16 ビット、UCS2 形式の UNICODE(ユニコード)に変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS2XE*w*.

説明

w

入力幅を指定します。16 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 2-32767

比較

\$UCS2XE*w*.入力形式では、\$UCS2X*w*.入力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
ucs2str=input ('大', \$ucs2xe4.); put ucs2str=\$hex8;	ucs2str=20002759

関連項目:

出力形式:

- “\$UCS2X 出力形式” (247 ページ)
- “\$UCS2XE 出力形式” (248 ページ)

入力形式:

- “\$UCS2X 入力形式” (514 ページ)

\$UCS4B 入力形式

ビッグエンディアン、32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS4Bw.

説明

w

入力幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

比較

同一の動作環境内のデータを変換する場合は、\$UCS4Xw.入力形式.異なる動作環境からのデータを処理する場合は、\$UCS4Bw.入力形式や\$UCS4Lw.入力形式を使用します。

例

これらの例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
ucs4str=input('大', \$ucs4be8.); put ucs4str= \$hex16.;	ucs4str=0000002000005927

関連項目:

出力形式:

- ["\\$UCS4B 出力形式" \(249 ページ\)](#)

入力形式:

- ["\\$UCS4L 入力形式" \(518 ページ\)](#)

- [“\\$UCS4X 入力形式” \(519 ページ\)](#)

\$UCS4L 入力形式

リトルエンディアン、32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS4L*w*.

説明

w

入力幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

比較

同一の動作環境内のデータを変換する場合は、\$UCS4X*w*.入力形式.異なる動作環境からのデータを処理する場合は、\$UCS4B*w*.入力形式や\$UCS4L*w*.入力形式を使用します。

例

これらの例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+----2----+----3----+
z=put('!.com',\$UCS4L16.); put z \$hex32.;	2E0000000630000006F0000006D000000

関連項目:

出力形式:

- [“\\$UCS4L 出力形式” \(252 ページ\)](#)

入力形式:

- [“\\$UCS4B 入力形式” \(517 ページ\)](#)

- ["\\$UCS4X 入力形式" \(519 ページ\)](#)

\$UCS4X 入力形式

32 ビット、UCS4 形式の UNICODE(ユニコード)文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS4X*w*.

説明

w

入力幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

比較

\$UCS4X*w*.入力形式では、\$UCS4XE*w*.入力形式とは逆の処理を行います。
\$UCS4X*w*.入力形式は、同一の動作環境内のデータを処理する場合に使用します。
\$UCS4B*w* 入力形式や\$UCS4L*w*.入力形式は、異なる動作環境からのデータを処理する場合に使用します。

例

これらの例では UTF-8 エンコーディングとリトルエンディアン形式を使用します。

ステートメント	結果
	----+----1----+
ucs4=put('91e5'x,\$ucs4x.);	ucs4=FDFF0000
utf8=input(ucs4,\$ucs4x.);	utf8=EFBFBD20
put ucs4=\$hex8. utf8=\$hex8.;	
run;	

関連項目:

出力形式:

- “\$UCS2X 出力形式” (247 ページ)
- “\$UCS2B 出力形式” (242 ページ)
- “\$UCS2L 出力形式” (244 ページ)
- “\$UCS4X 出力形式” (254 ページ)
- “\$UTF8X 出力形式” (263 ページ)

入力形式:

- “\$UCS2B 入力形式” (510 ページ)
- “\$UCS2L 入力形式” (512 ページ)
- “\$UTF8X 入力形式” (529 ページ)

\$UCS4XE 入力形式

現在の SAS セッションのエンコーディングの文字列を読み込み、その文字列を、32 ビット、UCS4 形式の UNICODE(ユニコード)に変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UCS4XE*w*.

説明

w

入力幅を指定します。32 ビットサイズの UNICODE 文字が入る十分な幅を指定します。

デフォルト 8

範囲 4-32767

比較

\$UCS4XE*w*.入力形式では、\$UCS4X*w*.入力形式とは逆の処理を行います。

例

この例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+

ステートメント	結果
<pre>\$ucs4xe4('大', \$ucs4xe8.); put ucs4str=\$hex16;</pre>	<pre>ucs4str=2000000027590000</pre>

関連項目:

出力形式:

- “\$UCS4X 出力形式” (254 ページ)
- “\$UCS4XE 出力形式” (256 ページ)

入力形式:

- “\$UCS4X 入力形式” (519 ページ)

\$UESC 入力形式

UESC 表記でエンコードされた文字列を読み取り、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UESC*w*.

説明

w

出力フィールドの幅を指定します。

デフォルト 8

範囲 1-32767

詳細

0 から 9、a から z、A から Z など、すべての動作環境で使用可能でない文字は、UESC 表記で表す必要があります。\$UESC*w*. 入力形式はネストできます。

比較

\$UESC*w*. 入力形式では、\$UESCE*w*. 入力形式とは逆の処理を行います。

例

これらの例では、Linux 動作環境でサポートされる UTF-8 エンコーディングを使用します。

ステートメント	結果
	----+----1----+
<pre>x=input('/u5927', \$uesc10.); y=input('\uu5927', \$uesc10.); z=input('\uuu5927', \$uesc10.); put x; put y; put z;</pre>	<pre>大 \u5927 \uuu5927</pre>

関連項目:

出力形式:

- [“\\$UESC 出力形式” \(257 ページ\)](#)
- [“\\$UESCE 出力形式” \(258 ページ\)](#)

入力形式:

- [“\\$UESCE 入力形式” \(522 ページ\)](#)

\$UESCE 入力形式

現在の SAS セッションのエンコーディングを使用する文字列を読み取り、その文字列を UESC 表記に変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UESCE*w*.

説明

w

入力幅を指定します。

デフォルト 8

範囲 1-32767

詳細

\$UESCEw.入力形式はネストできます。

比較

\$UESCEw.入力形式では、\$UESCW.入力形式とは逆の処理を行います。

例

これらの例では、Linux 動作環境でサポートされる日本語(Shift_JIS)エンコーディングを使用します。

ステートメント	結果
	----+----1----+
<pre>x=input('大', \$uesce10.); y=input('¥u5927',\$uesce10.);</pre>	<pre>\u5927</pre>

関連項目:

出力形式:

- ["\\$UESC 出力形式" \(257 ページ\)](#)
- ["\\$UESCE 出力形式" \(258 ページ\)](#)

入力形式:

- ["\\$UESC 入力形式" \(521 ページ\)](#)

\$UNCR 入力形式

NCR 文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UNCR*w*.

説明

w 入力幅を指定します。

デフォルト 8

範囲 1-32767

詳細

入力文字列に含まれるのは、文字と NCR のみにする必要があります。各国語の文字は、NCR で表記する必要があります。

比較

\$UNCRw.入力形式では、\$UNCREw.入力形式とは逆の処理を行います。

例

これらの例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
<pre>x=input ('&#22823;', \$uncr10.); y=input('abc', \$uncr10.); put X; put Y;</pre>	<pre>大 abc</pre>

関連項目:

出力形式:

- [“\\$UNCR 出力形式” \(259 ページ\)](#)
- [“\\$UNCRE 出力形式” \(260 ページ\)](#)

入力形式:

- [“\\$UNCRE 入力形式” \(524 ページ\)](#)

\$UNCRE 入力形式

現在の SAS セッションのエンコーディングを使用する文字列を読み込み、その文字列を NCR に変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UNCREw.

説明

w

入力幅を指定します。

デフォルト 8

範囲 1-32767

詳細

出力文字列は、プレーンテキストと NCR に変換されます。各国語の文字は、NCR に変換されます。

比較

\$UNCREw.入力形式では、\$UNCRw.入力形式とは逆の処理を行います。

例

これらの例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
x=input (' 𐀀 abc', \$uncre12.); put x;	大abc

関連項目:

出力形式:

- [“\\$UNCR 出力形式” \(259 ページ\)](#)
- [“\\$UNCRE 出力形式” \(260 ページ\)](#)

入力形式:

- [“\\$UNCR 入力形式” \(523 ページ\)](#)

\$UPAREN 入力形式

UPAREN 表記でエンコードされた文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UPAREN*w*.

説明

w

入力幅を指定します。

デフォルト 8

範囲 1-32767

詳細

SAS セッションのエンコーディングに対応する UNICODE 表記がない場合、その表記は現在の SAS セッションのエンコーディングのまま残ります。

比較

\$UPAREN*w*.入力形式では、\$UPARENE*w*.入力形式とは逆の処理を行います。

例

これらの例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
<code>v=input('<u0061>', \$uparen10.);</code>	a
<code>w=input('<u0062>', \$uparen10.);</code>	b
<code>x=input('<u0063>', \$uparen10.);</code>	c
<code>y=input('<u0033>', \$uparen10.);</code>	3
<code>z=input('<u5927>', \$uparen10.);</code>	大
<code>put v;</code>	
<code>put w;</code>	
<code>put x;</code>	
<code>put y;</code>	
<code>put z;</code>	

関連項目:

出力形式:

- [“\\$UPAREN 出力形式” \(261 ページ\)](#)
- [“\\$UPARENE 出力形式” \(262 ページ\)](#)

入力形式:

- [“\\$UPARENE 入力形式” \(527 ページ\)](#)
- [“\\$UPAREN*P* 入力形式” \(528 ページ\)](#)

\$UPARENE 入力形式

現在の SAS セッションのエンコーディングを使用する文字列を読み込み、その文字列を UPAREN 表記に変換します。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UPARENE*w*.

説明

w

入力幅を指定します。

デフォルト 8

範囲 1-32767

比較

\$UPARENE*w*.入力形式では、\$UPAREN*w*.入力形式とは逆の処理を行います。

例

これらの例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
<pre>v=input('a',\$uparene10.); w=input('b',\$uparene10.); x=input('c',\$uparene10.); y=input('3',\$uparene10.); z=input('大',\$uparene10.); put v; put w; put x; put y; put z;</pre>	<pre><u0061> <u0062> <u0063> <u0033> <u5927></pre>

関連項目:

出力形式:

- ["\\$UPAREN 出力形式" \(261 ページ\)](#)

- “\$UPARENE 出力形式” (262 ページ)

入力形式:

- “\$UPAREN 入力形式” (525 ページ)
- “\$UPAREN P 入力形式” (528 ページ)

\$UPAREN P 入力形式

UPAREN 表記でエンコードされた文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換しますが、各国語文字は UPAREN 表記のエンコーディングのまま残ります。

カテゴリ: 文字

制限事項: UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。

構文

\$UPAREN P*w*.

説明

w

入力幅を指定します。

デフォルト 8

範囲 1-32767

詳細

UPAREN 表記に各国語文字が含まれ、その値が UNICODE の 0x00ff より大きい場合、その表記は UPAREN 表記のまま残ります。

例

これらの例では UTF-8 エンコーディングを使用しています。

ステートメント	結果
	----+----1----+
<pre>v=input('<u0061>', \$uparenp10.); w=input('<u0062>', \$uparenp10.); x=input('<u0063>', \$uparenp10.); y=input('<u0033>', \$uparenp10.); z=input('<u5927>', \$uparenp10.); put v; put w; put x; put y; put z;</pre>	<pre>a b c 3 <u5927></pre>

関連項目:**出力形式:**

- ["\\$UPAREN 出力形式" \(261 ページ\)](#)
- ["\\$UPARENE 出力形式" \(262 ページ\)](#)

入力形式:

- ["\\$UPAREN 入力形式" \(525 ページ\)](#)
- ["\\$UPARENE 入力形式" \(527 ページ\)](#)

\$UTF8X 入力形式

UTF-8 でエンコードされた文字列を読み込み、その文字列を現在の SAS セッションのエンコーディングに変換します。

カテゴリ: 文字

構文

\$UTF8X*w*.

説明

w

入力幅を指定します。

デフォルト 8

範囲 1-32767

比較

この例では、Linux 動作環境でサポートされる日本語(UTF-8)エンコーディングを使用します。

ステートメント	結果
	----+----1----+
x=input ('e5a4a7'x, \$utf8x3.); put x;	大

関連項目:**出力形式:**

- ["\\$UCS2B 出力形式" \(242 ページ\)](#)

- “\$UCS2L 出力形式” (244 ページ)
- “\$UCS2X 出力形式” (247 ページ)
- “\$UTF8X 出力形式” (263 ページ)

入力形式:

- “\$UCS2B 入力形式” (510 ページ)
- “\$UCS2L 入力形式” (512 ページ)
- “\$UCS2X 入力形式” (514 ページ)

7 部

各国語サポート関連のマクロ関数

14 章	
各国語サポート関連のマクロ関数のディクショナリ.....	533

14 章

各国語サポート関連のマクロ関数の ディクショナリ

カテゴリ別のマクロ	533
ディクショナリ	534
%KCOMPRES マクロ関数および%QKCOMPRES マクロ関数	534
%KINDEX マクロ関数	534
%KLEFT と%QKLEFT マクロ関数	535
%KLENGTH マクロ関数	535
%KSCAN と%QKSCAN 関数	536
%KSUBSTR と%QKSUBSTR マクロ関数	538
%KUPCASE と%QKUPCASE マクロ関数	540

カテゴリ別のマクロ

次の表に、SAS 各国語サポートマクロ関数の概要を示します。詳細については、それぞれの各国語サポート関連のマクロ関数に関する情報を参照してください。

カテゴリ	言語要素	説明
DBCS	%KCOMPRES マクロ関数および%QKCOMPRES マクロ関数 (p. 534)	複数の空白を圧縮し、先頭と末尾の空白を削除します。
	%KINDEX マクロ関数 (p. 534)	文字列の先頭文字の位置を返します。
	%KLEFT と%QKLEFT マクロ関数 (p. 535)	先頭の空白を削除し、引数を左詰めにします。
	%KLENGTH マクロ関数 (p. 535)	文字列の長さを返します。
	%KSCAN と%QKSCAN 関数 (p. 536)	文字列内の単語を位置指定して検索します。
	%KSUBSTR と%QKSUBSTR マクロ関数 (p. 538)	文字列の部分文字列を生成します。

カテゴリ	言語要素	説明
	%KUPCASE と %QKUPCASE マクロ関数 (p. 540)	値を大文字に変換します。

ディクショナリ

%KCMPRES マクロ関数および%QKCMPRES マクロ関数

複数の空白を圧縮し、先頭と末尾の空白を削除します。

カテゴリ: DBCS
種類: NLS マクロ関数

構文

%KCMPRES (*text* | *text expression*)

%QKCMPRES (*text* | *text expression*)

詳細

%KCMPRES マクロ関数および%QKCMPRES マクロ関数は、複数の空白を圧縮し、先頭と末尾の空白を削除します。%KCMPRES は、引数が引用符で囲まれている場合でも、引用符で囲まれていない結果を返します。%QKCMPRES は、引用符で囲まれた結果を返します。

%QKCMPRES は、次の特殊文字とニーモニック演算子がマスクされた結果を生成します。そのためマクロプロセッサは、それらの結果を、マクロ言語の要素ではなくテキストとして解釈します。

```
& % ' " ( ) + - * / < > = ^ ~ ; , # blank AND OR NOT EQ NE LE LT GE GT IN
```

%KINDEX マクロ関数

文字列の先頭文字の位置を返します。

カテゴリ: DBCS
種類: NLS マクロ関数

構文

%KINDEX (*source*, *string*)

必須引数

source

文字列またはテキスト式です。

string

文字列またはテキスト式です。

詳細

%KINDEX 関数は、*source* を検索して、最初に出現する *string* の先頭文字の位置を返します。*string* が見つからない場合、関数は 0 を返します。

例: 文字の検索

次のステートメントは、文字列で最初に出現する文字 **V** を検索します。

```
%let a=a very long value;
%let b=%kindex(&a,v);
%put V appears at position &b.;
```

ステートメントが実行されると、SAS ログに次の行が出力されます。

```
V appears at position 3.
```

%KLEFT と %QKLEFT マクロ関数

先頭の空白を削除し、引数を左詰めにします。

カテゴリ: DBCS
要件 MAUTOSOURCE システムオプション

構文

%KLEFT (*text* | *text expression*)

%QKLEFT (*text* | *text expression*)

詳細

%KLEFT と %QKLEFT マクロ関数は、先頭の空白を削除し、引数を左詰めにします。引数が次の特殊文字またはニーモニック演算子を含む場合は、QKLEFT を使用します。

%KLEFT は、引数が引用符で囲まれている場合でも、引用符で囲まれていない結果を返します。%QKLEFT は、次の特殊文字とニーモニック演算子がマスクされた結果を生成します。そのためマクロプロセッサは、それらの結果を、マクロ言語の要素ではなくテキストとして解釈します。

```
& % ' ' ( ) + - * / < > = ~ ^ ~ ; , # blank
AND OR NOT EQ NE LE LT GE GT IN
```

%KLENGTH マクロ関数

文字列の長さを返します。

カテゴリ: DBCS
種類: NLS マクロ関数

構文

%KLENGTH (*character string* | *text expression*)

詳細

引数が文字列の場合、%KLENGTH は文字列の長さを返します。引数がテキスト式の場合、%KLENGTH は変換された値の長さを返します。引数が NULL 値の場合、%KLENGTH は 0 を返します。

例: 文字列の長さを返す

次のステートメントは、文字列とテキスト式の長さを返します。

```
%let a=Happy;  
%let b=Birthday;  
%let c=René;  
%put The length of &a is %length(&a);  
%put The length of &b is %length(&b);  
%put The length of &c is %length(&c);  
%put The length of &a &b To &c is %length(&a &b to &c);
```

```
%put with LENGTH macro %length(&c);
```

ステートメントが実行されると、SAS ログに次の行が出力されます。

```
The length of Happy is 5.  
The length of Birthday is 8.  
The length of René is 4.  
The length of Happy Birthday To René is 22.
```

```
with LENGTH macro 5
```

%KSCAN と %QKSCAN 関数

文字列内の単語を位置指定して検索します。

カテゴリ: DBCS
種類: NLS マクロ関数

構文

%KSCAN (*argument*, *position*, *length*)

%QKSCAN (*argument*, *n*<,*delimiters* >)

必須引数

argument

文字列またはテキスト式です。*argument* が後述する特殊文字またはニーモニック演算子を含む場合は、%QKSCAN を使用します。

n

整数または整数を生成するテキスト式です。この整数は、単語の位置を表します。*n* が *argument* にある単語数より大きい場合、関数は NULL 文字列を返します。*n* が負の場合、%KSCAN は文字列の最後の単語から逆方向に検索して、単語を選択します。

delimiter

%QKSCAN が文字式内で単語区切り文字として使用する文字を指定します。

詳細

%KSCAN と %QKSCAN 関数は、*argument* を検索して、*n* 番目の単語を返します。単語とは、1 つ以上の区切り文字によって区切られた 1 つ以上の文字のことです。

%KSCAN では、引数がマクロ引用関数によりマスクされていても、特殊文字とニーモニック演算子をマスクせずに値を返します。%%QKSCAN は、次の特殊文字とニーモニック演算子をマスクした値を返します。

```
& % ' ( ) + - * / < > = ~ ^ ~ ; , # blank  
AND OR NOT EQ NE LE LT GE GT IN
```

%KSCAN 関数では、次の条件をすべて満たす部分文字列を *word* (単語) とします。

- 左境界が、区切り文字または文字列の先頭である
- 右境界が、区切り文字または文字列の末尾である
- 区切り文字を含まない

単語は、その文字列の先頭または末尾に区切り文字がある場合、またはその文字列に 2 つ以上の連続した区切り文字が含まれる場合には長さが 0 となります。

%KSCAN 関数に指定する引数が 2 つのみの場合、デフォルトの区切り文字は、コンピュータの使用文字が ASCII であるか EBCDIC であるかによって異なります。

- コンピュータの使用文字が ASCII の場合、次の文字がデフォルトの区切り文字です。

```
blank ! $ % & ( ) * + , - . / ; < ^ | / ; < ^ |
```

^文字が含まれない ASCII 環境では、%KSCAN 関数は代わりに ~文字を使用します。

- コンピュータの使用文字が EBCDIC の場合、次の文字がデフォルトの区切り文字です。

```
blank ! $ % & ( ) * + , - . / ; < ^ | / ; < ~ | ¢ |
```

%KSCAN 関数では、文字引数に NULL を指定できます。NULL 引数は、長さがゼロの文字列として処理されます。数値引数には NULL を指定できません。

例: %KSCAN と %QKSCAN の処理の比較

次の例に、%KSCAN と %QKSCAN の処理を示します。

```
%macro a;
```

```

aaaaaa
%mend a;
%macro b;
bbbbbb
%mend b;
%macro c;
cccccc
%mend c;
%let x=%nrstr(%a*%b*%c);
%put X: &x;
%put The third word in X, with KSCAN: %kscan(&x,3,*);
%put The third word in X, with QKSCAN: %qkscan(&x,3,*);

%PUT ステートメントは、ログに次の行を出力します。

X: %a*%b*%c
The third word in X, with KSCAN: ccccc
The third word in X, with QKSCAN: %c

```

%KSUBSTR と %QKSUBSTR マクロ関数

文字列の部分文字列を生成します。

カテゴリ: DBCS
種類: NLS マクロ関数

構文

%KSUBSTR (*argument*, *position*<, *length*>)

%QKSUBSTR (*argument*, *position*<, *length*>)

必須引数

argument

文字列またはテキスト式です。*argument* が後述する特殊文字またはニーモニック演算子を含む場合は、%QKSUBSTR を使用します。

position

整数または整数を生成する(テキスト、論理、算術)式です。この整数は、部分文字列の先頭文字の位置を表します。*position* が文字列の文字数より大きい場合、%KSUBSTR と %QKSUBSTR は警告メッセージを発行して、NULL 値を返します。

length

整数または整数を生成する(テキスト、論理、算術)式です(オプション)。この整数は、部分文字列の文字数を表します。*length* が *argument* 内の *position* 以降の文字数より大きい場合、%KSUBSTR と %QKSUBSTR は警告メッセージを発行して、*position* から文字列の末尾までの文字で構成される部分文字列を返します。デフォルトで、%KSUBSTR と %QKSUBSTR は *position* から文字列の末尾までの文字で構成される文字列を作成します。

詳細

%KSUBSTR と %QKSUBSTR 関数は、*argument* 内の *position* 位置から *length* の文字数分の文字で構成される部分文字列を生成します。

%KSUBSTR は、特殊文字とニーモニック演算子をマスクせずに値を返します。
 %QSUBSTR は、次の特殊文字とニーモニック演算子をマスクします。

```
& % ' " ( ) + - * / < > = ~ ^ ~ ; , # blank
AND OR NOT EQ NE LE LT GE GT IN
```

例

例 1: ファイル参照名を 8 文字に制約する

マクロ MAKEFREF は %KSUBSTR を使用して、ユーザーが割り当てたパラメータが 8 文字より長い場合、先頭の 8 文字をファイル参照名に割り当てます。

```
%macro makefref(fileref,file);
  %if %klength(&fileref) gt 8 %then
    %let fileref = %ksubstr(&fileref,1,8);
    filename &fileref "&file";
  %mend makefref;
%makefref(humanresource,/dept/humanresource/report96)
```

SAS は、次のステートメントを読み込みます。

```
FILENAME HUMANRES "/dept/humanresource/report96";
```

例 2: セグメントの長いマクロ変数値の保存

マクロ SEPMSG は、マクロ変数 MSG の値を 40 文字単位に分割して、単位ごとに異なる変数に保存します。

```
%macro sepmsg(msg);
  %let i=1;
  %let start=1;
  %if %length(&msg)>40 %then
    %do;
      %do %until(%klength(&&msg&i)<40);
        %let msg&i=%qsubstr(&msg,&start,40);
        %put Message &i is: &&msg&i;
        %let i=%eval(&i+1);
        %let start=%eval(&start+40);
        %let msg&i=%qsubstr(&msg,&start);
      %end;
      %put Message &i is: &&msg&i;
    %end;
  %else %put No subdivision was needed.;
%mend sepmsg;
%sepmsg(%nrstr(A character operand was found in the %EVAL function
or %IF condition where a numeric operand is required. A character
operand was found in the %EVAL function or %IF condition where a
numeric operand is required.));
```

このプログラムが実行されると、次の行が SAS ログに出力されます。

```
Message 1 is: A character operand was found in the %E
Message 2 is: AL function or %IF condition where a nu
Message 3 is: meric operand is required. A character
Message 4 is: operand was found in the %EVAL function
Message 5 is: or %IF condition where a numeric operan
Message 6 is: d is required.
```

例 3: %KSUBSTR と %QSUBSTR の処理の比較

%KSUBSTR は、C 言語で特殊文字とニーモニック演算子をマスクせずに結果を処理するため、名前が解決された結果を生成します。

```
%let a=one;
%let b=two;
%let c=%nrstr(&a &b);
%put C: &c;
%put With KSUBSTR: %ksubstr(&c,1,2);
%put With QSUBSTR: %qsubstr(&c,1,2);
```

これらのステートメントが実行されると、次の行が SAS ログに出力されます。

```
C: &a &b
With KSUBSTR: one
With QSUBSTR: &a
```

%KUPCASE と %QKUPCASE マクロ関数

値を大文字に変換します。

カテゴリ: DBCS
種類: NLS マクロ関数

構文

%KUPCASE (*character string* | *text expression*)

%QKUPCASE (*character string* | *text expression*)

詳細

%KUPCASE と %QKUPCASE 関数は、引数内の小文字を大文字に変換します。
%KUPCASE は、特殊文字とニーモニック演算子をマスクせずに値を返します。

引数が次の特殊文字またはニーモニック演算子を含む場合は、%QKUPCASE を使用します。
%QKUPCASE は、次の特殊文字とニーモニック演算子をマスクして値を返します。

```
& % ' ( ) + - * / < > = ~ ^ ~ ; , # blank
AND OR NOT EQ NE LE LT GE GT IN
```

値を比較する場合、マクロ機能では比較前に小文字から大文字に自動的に変換しないため、%KUPCASE と %QKUPCASE が役に立ちます。

例**例 1: 比較する値を大文字に変更する**

この例では、マクロ RUNREPT は、マクロ変数 MONTH の入力値と文字列 DEC を比較します。大文字に変換された値が DEC の場合、PROC FSVIEW はデータセット REPORTS.ENDYEAR で実行されます。DEC ではない場合、PROC FSVIEW は、REPORTS データライブラリの月名のデータセットで実行されます。

```
%macro runrept(month);
  %if %kupcase(&month)=DEC %then
    %str(proc fsview data=reports.endyear; run;);
```

```
%else %str(proc fsview data=reports.&month; run;);  
%mend runrept;
```

%IF 条件を満たすには、次のいずれかの方法でマクロを起動します。

```
%runrept(DEC)  
%runrept(Dec)  
%runrept(dec)
```

例 2: %KUPCASE と %QKUPCASE の比較

次のステートメントは、%KUPCASE と %QKUPCASE により生成される結果を示します。

```
%let a=begin;  
%let b=%nrstr(&a);  
%put KUPCASE produces: %kupcase(&b);  
%put QKUPCASE produces: %qkupcase(&b);
```

ステートメントが実行されると、SAS ログに次の行が出力されます。

```
KUPCASE produces: BEGIN  
QKUPCASE produces: &A
```


8 部

各国語サポート関連のシステム オプション

15 章

各国語サポート関連のシステムオプションのディクショナリ..... 545

15 章

各国語サポート関連のシステムオプションのディクショナリ

カテゴリ別のシステムオプションエントリ	545
ディクショナリ	546
BOMFILE システムオプション	546
DFLANG=システムオプション	547
ENCODING システムオプション	549
LOCALE システムオプション	550
LOCALEDATA システムオプション	551
LOGLANGCHG システムオプション	552
LOGLANGENG システムオプション	553
LSWLANG システムオプション	554
MAPEBCDIC2ASCII=システムオプション	555
NLDECSEPARATOR システムオプション	556
ODSLANGCHG システムオプション	557
RSASIOTRANSERROR システムオプション	559
SORTSEQ=システムオプション	559
TIMEZONE=システムオプション	560
VALIDMEMNAME=システムオプション	562
VALIDVARNAME=システムオプション	564

カテゴリ別のシステムオプションエントリ

SAS システムオプションの言語コントロールカテゴリは、各国語サポートの影響を受けます。次の表に、SAS システムオプションの概要を示します。詳細については、各 SAS システムオプションのディクショナリエントリを参照してください。

カテゴリ	言語要素	説明
SORT: プロシジャオプション	SORTSEQ=システムオプション (p. 559)	SORT および SQL プロシジャが現在の SAS セッションで使用する、言語固有の照合順序を指定します。
環境コントロール: 言語コントロール	DFLANG=システムオプション (p. 547)	国際化対応の日付の入力形式および出力形式の言語を指定します。
	ENCODING システムオプション (p. 549)	SAS セッションのデフォルト文字セットエンコーディングを指定します。

カテゴリ	言語要素	説明
	LOCALE システムオプション (p. 550)	地域の言語、地域の規則および文化を反映する、SAS セッションの属性一式を指定します。
	LOCALEDATA システムオプション (p. 551)	ロケール情報のソースデータベースを指定します。
	LOGLANGENG システムオプション (p. 553)	起動後に LOCALE オプションを設定した場合に、SAS ログメッセージテキストに英語を使用するかを指定します。
	MAPEBCDIC2ASCII=システムオプション (p. 555)	EBCDIC から ASCII へのトランスコードと、ASCII から EBCDIC へのトランスコードで SAS ユーザーが使用する変換テーブルを指定します。
	ODSLANGCHG システムオプション (p. 557)	ODS 出力のテキスト言語を変更できるかを設定します。
	TIMEZONE=システムオプション (p. 560)	ユーザーローカルタイムゾーンを指定します。
言語コントロール	LSWLANG システムオプション (p. 554)	SAS 起動時に LOGLANGCHG オプションまたは ODSLANGCHG オプションを設定する際に、言語切り替え機能で使用する言語を指定します。
ファイル: SAS ファイル	RSASIoTERROR システムオプション (p. 559)	リモートアプリケーションから無効なデータが読み込まれたときにトランスコーディングエラーを表示します。
	VALIDMEMNAME=システムオプション (p. 562)	SAS データセット、SAS データビューおよびアイテムストアの命名規則を指定します。
	VALIDVARNAME=システムオプション (p. 564)	SAS セッション中に作成および処理可能な有効な SAS 変数名の規則を指定します。
ファイル: 外部ファイル	BOMFILE システムオプション (p. 546)	Unicode でエンコードされた外部ファイルにバイトオーダーマーク(BOM)接頭辞を書き出すかどうかを指定します。

ディクショナリ

BOMFILE システムオプション

Unicode でエンコードされた外部ファイルにバイトオーダーマーク(BOM)接頭辞を書き出すかどうかを指定します。

該当要素: 構成ファイル、SAS 起動時、OPTIONS ステートメント、**[SAS システムオプション]** ウィンドウ

カテゴリ: ファイル: 外部ファイル

PROC OPTIONS GROUP= EXTFILES

構文

BOMFILE | NOBOMFILE

説明

BOMFILE

Unicode でエンコードされたファイルを外部ファイルに書き出すときに、バイトオーダーマーク(BOM)接頭辞を書き出すように指定します。

NOBOMFILE

Unicode でエンコードされたファイルを外部ファイルに書き出すときに、BOM 接頭辞を書き出さないように指定します。

詳細

BOMFILE システムオプションは、Unicode でエンコードされた外部ファイルを読み込むときには適用されません。

BOM は、Unicode データストリームの先頭の署名です。BOM のサイズはエンコーディングにより異なります。

DFLANG=システムオプション

国際化対応の日付の入力形式および出力形式の言語を指定します。

該当要素: 構成ファイル、SAS 起動時、OPTIONS ステートメント、[SAS システムオプション] ウィンドウ

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS GROUP= LANGUAGECONTROL

デフォルト: English

構文

DFLANG=*language*, locale

説明

language

国際化対応の日付の入力形式および出力形式に使用する言語を指定します。

language の値として有効なのは、次の言語です。

- Afrikaans
- Catalan
- Croatian
- Czech
- Danish
- Dutch
- English

- Finnish
- French
- German
- Hungarian
- Italian
- Japanese
- Macedonian
- Norwegian
- Polish
- Portuguese
- Russian
- Slovenian
- Spanish
- Swedish
- Swiss_French
- Swiss_German

locale

LOCALE システムオプションで指定されたロケールがアクティブなロケールになります。

詳細

DFLANG システムオプションの値は、SAS セッション中に変更できますが、一度に使用できる言語は 1 つのみです。language の値では、大文字と小文字は区別されません。

DFLANG=locale を指定すると、LOCALE ステートメントのシステムオプションに指定されているロケールが有効なロケールになります。locale または language は DFLANG システムオプションでサポートされる言語である必要があります。

次の例では、国際化対応の日付の入力形式および出力形式にはドイツ語 (German) が使用されます。ドイツ語ロケールの POSIX 名は de_DE になります。ドイツ語ロケールは DFLANG システムオプションでサポートされています。

```
option locale=de_DE; /* German locale */
option DFLANG=locale;
```

次の例では、国際化対応の日付の入力形式および出力形式には英語 (English) が使用されます。マルタ語 (Maltese) は DFLANG でサポートされていないので、デフォルトロケールは English になります。

```
option locale=mt_MT; /* Maltese locale */
option DFLANG=locale;
```

DFLANG=locale を指定すると、日付形式の出力は、LOCALE システムオプションで指定されたロケールで表示されます。出力の日付形式をコントロールするために、DFLANG のロケールとして起動時に設定された LOCALE システムオプションに基づく値を使用します。DFLANG が有効な言語に設定されていない場合、出力の日付形式はデフォルトでは英語になります。次の例では、ロケールはフランス語に設定されています。

```
Sas.exe -locale French
Proc print data=sashelp.class ; run ;
```

```
mercredi 09 mars 2011 14 h 25
```

DFLANG=JAPANESE を設定した場合、DFLANG は **DFLANG=LOCALE** の場合と同じように動作します。

値 **JAPANESE** は、日本語版イメージを実行している顧客が、日本語向けに書式化された SAS ログや SAS リスティングで日付および時刻スタンプを確認できるようにするためにサポートされています。すべての顧客に同じ機能を提供するために、後で値 **LOCALE** が追加されています。

- 起動時に DFLANG が設定されていない場合には、LOCALE オプションによってオプション値が設定されます。
- SAS ログの日付および時刻スタンプのデフォルトの表示形式は英語です。DFLANG として LOCALE または JAPANESE が設定されている場合、SAS ログ内の日付および時刻スタンプは現在の SAS LOCALE に従ってフォーマットされます。
- SAS リストの日付および時刻スタンプのデフォルト表示形式は英語です。DFLANG として LOCALE または JAPANESE が設定されている場合、SAS リストまたは ODS 宛先に送信される出力の各ページに含まれる日付および時刻スタンプは、DFLANG=LOCALE が設定された直後に有効になった SAS LOCALE に従ってフォーマットされます。
- LOCALE オプションに対する変更は、DTRESET オプションが有効な場合を除き、日付および時刻スタンプのフォーマット方法に影響しません。その場合、日付および時刻スタンプは現在の SAS LOCALE に従ってフォーマットされます。

値 **JAPANESE** を使用して日本語イメージを処理して、SAS ログの日時スタンプや日本語向けにフォーマットされたリストを表示できます。

関連項目:

[DTRESET システムオプション](#)

ENCODING システムオプション

SAS セッションのデフォルト文字セットエンコーディングを指定します。

該当要素: 構成ファイル、SAS 起動時

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS LANGUAGECONTROL
GROUP=

構文

-ENCODING= *encoding-value*

必須引数

encoding-value
UTF-8

注 UTF-8、Unicode エンコーディングのみがサポートされているエンコーディングです。

詳細

文字セットエンコーディングは、コードポイントと呼ばれる数値にマップされた文字のセットです。

UTF-8 は SAS Viya がサポートしている唯一の SAS セッションのエンコーディングです。SAS セッションのエンコーディングは、次のように ENCODING=オプションの値で決まります。ENCODING=が指定されていない場合は、UTF-8 がデフォルトで設定されます。

関連項目:

概念的な情報:

- [“各国語サポート関連のロケール概念の概要” \(5 ページ\)](#)
- [“概要: 各国語サポート関連のエンコーディング” \(9 ページ\)](#)
- [“トランスコーディングの概要” \(19 ページ\)](#)
- [18 章, “SAS 言語要素中のエンコーディング値” \(613 ページ\)](#)

LOCALE システムオプション

地域の言語、地域の規則および文化を反映する、SAS セッションの属性一式を指定します。

該当要素: 構成ファイル、SAS 起動時、OPTIONS ステートメント、**[SAS システムオプション]** ウィンドウ

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS GROUP= LANGUAGECONTROL

デフォルト: English_UnitedStates

動作環境: SASV9_OPTIONS 環境変数を用いて指定することも可能

構文

-LOCALE *locale-name*

LOCALE=*locale-name*

必須引数

locale-name

ロケール値(SAS 名および POSIX 名)の一覧については、[“PAPERSIZE と DFLANG オプションの LOCALE=の値” \(599 ページ\)](#)を参照してください。

詳細

LOCALE=システムオプションは、地域の規則、言語および文化を反映するロケールの指定に使用します。

LOCALE=システムオプションの値に ENCODING=システムオプションの値との互換性がない場合、文字セットエンコーディングは、ENCODING=システムオプションの値によって決まります。

LOCALE=の値を設定すると、次のシステムオプションの値は、明示的に値を指定しない限り、変更されます。

DATESTYLE=

LOCALE=が設定されると、DATESTYLE=システムオプションには選択されたロケールに対応する値が使用されます。

DFLANG=

LOCALE=が設定されると、DFLANG=システムオプションは選択されたロケールに対応する値に設定されます。

PAPERSIZE=

LOCALE=が設定されると、PAPERSIZE=システムオプションは選択されたロケールに対応する値に設定され、ODS プリンタがそのロケールの優先測定単位(インチまたはセンチメートル)に設定されます。

関連項目:

概念的な情報:

- [2 章, “各国語サポート関連のロケール” \(5 ページ\)](#)
- [“PAPERSIZE と DFLANG オプションの LOCALE=の値” \(599 ページ\)](#)

システムオプション:

- [“ENCODING システムオプション” \(549 ページ\)](#)
- [“DFLANG=システムオプション” \(547 ページ\)](#)

LOCALEDATA システムオプション

ロケール情報のソースデータベースを指定します。

該当要素: 構成ファイル、SAS 起動時

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS LANGUAGECONTROL

GROUP=

デフォルト: SASLOCALE

構文

LOCALEDATA=SASLOCALE | REGISTRY

説明

SASLOCALE

内部 SAS データベースのロケール定義を使用します。

REGISTRY

SAS レジストリからのロケール定義を使用します。

詳細

SASLOCALE 値システムオプションは、ロケール定義のソースとして内部データベースを使用することを示します。このデータベースでは、カスタマイズは行えません。

REGISTRY 値は、外部データベースを指定します。外部データベースでは、PROC LOCALEDATA を使用してロケール定義をカスタマイズできます。

LOGLANGCHG システムオプション

SAS ログのテキスト出力の言語を変更できるかを指定します。

該当要素: 構成ファイル、SAS 起動時

**PROC OPTIONS
GROUP=** LOGCONTROL

デフォルト: NOLOGLANGCHG

構文

LOGLANGCHG | NOLOGLANGCHG

説明

LOGLANGCHG

起動後に SAS ログメッセージの言語を変更できるようにする場合に指定します。

NOLOGLANGCHG

起動後に SAS ログメッセージの言語を変更できないようにする場合に指定します。

詳細

結果は条件によって異なります。

ODSLANGCHG	LOGLANGCHG	結果
ON	ON	ODS 出力および SAS ログに使用する言語の変更可。変更には LSWLANG=オプション設定を使用。
OFF	OFF	ODS 出力と SAS ログ出力に使用する言語の変更不可。
OFF	ON	SAS ログに使用する言語の変更可能。変更には LSWLANG=オプション設定を使用。

ODSLANGCHG	LOGLANGCHG	結果
ON	OFF	ODS 出力に使用する言語の変更可能。変更には LSWLANG=オプション設定を使用。

関連項目:

- [“ODSLANGCHG システムオプション” \(557 ページ\)](#)
- [“LSWLANG システムオプション” \(554 ページ\)](#)
- [“LOGLANGENG システムオプション” \(553 ページ\)](#)

LOGLANGENG システムオプション

起動後に LOCALE オプションを設定した場合に、SAS ログメッセージテキストに英語を使用するかを指定します。

該当要素: 構成ファイル、SAS 起動時

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS GROUP= LANGUAGECONTROL

デフォルト: NOLOGLANGENG

構文

LOGLANGENG | NOLOGLANGENG

説明

LOGLANGENG

英語の SAS ログメッセージを表示するように言語スイッチ機能を使用します。

NOLOGLANGENG

英語の SAS ログメッセージを表示するように言語スイッチ機能を使用しません。

詳細

LOGLANGENG オプションと、LOGLANGCHG、ODSLANGCHG および LSWLANG オプションは互いに影響します。

- LOGLANGENG を指定した場合、LSWLANG=EN と LOGLANGCHG を指定した場合と同じように機能します。
- ODSLANGCHG を指定しても、LOGLANGENG を指定できます。
- NOLOGLANGENG は、SAS セッションに影響しません。
- 起動時は、構成ファイル(!sasroot\lnls)と LOCALE オプションに基づき SAS メッセージの言語が決定されます。起動後は LOCALE オプションを設定した場

合でも、LOGLANGENG オプションを指定しているならば、SAS ログは英語で出力されます。

- 起動時に LOGLANGENG を指定すると、セッション中に LOCALE の値が変更された場合でも、SAS ログは引き続き英語で出力されます。
- LOGLANGCHG を指定すると、SAS ログは LSWLANG=オプションによって制御されます。
- ODSLANGCHG を指定すると、ODS テキストは LSWLANG=オプションによって制御されます。
- 起動時に LOGLANGENG を指定せずに、LOCALE をセッション中に変更した際、起動時に設定した言語で SAS ログが出力されることがあります。この現象は、LOGLANGCHG と LSWLANG=をともに指定し、LSWLANG の値に LOCALE 以外を指定している場合に発生します。

例

例として、LOGLANGENG が指定されているフランス語の SAS セッションを考えます。

LOGLANGENG が指定されると、自動的に、LSWLANG=EN が指定され、LOGLANGCHG が設定されます。この SAS セッションでは、SAS ログは常に英語で出力されますが、ODS 出力の言語は現在の SAS 構成に基づき決定されます。

関連項目:

- [“LSWLANG システムオプション” \(554 ページ\)](#)
- [“LOGLANGCHG システムオプション” \(552 ページ\)](#)
- [“ODSLANGCHG システムオプション” \(557 ページ\)](#)

LSWLANG システムオプション

SAS 起動時に LOGLANGCHG オプションまたは ODSLANGCHG オプションを設定する際に、言語切り替え機能で使用する言語を指定します。

該当要素: 構成ファイル、SAS 起動時

カテゴリ: 言語コントロール

PROC OPTIONS GROUP= LANGUAGECONTROL

デフォルト: LOCALE

構文

LSWLANG=LOCALE | *language*

必須引数

LOCALE

LOCALE 引数を指定すると、SAS9.4 より前の動作が維持されます。SAS メッセージテキストの言語として、LOCALE=オプションの値が使用されます。

language

ODS 出力と SAS ログ出力に使用する言語を指定します。次の値を指定できます。

コード	言語
EN	英語
FR	フランス語
IT	イタリア語
DE	ドイツ語
ES	スペイン語
ZH	簡体字中国語
ZT	繁体字中国語
JA	日本語
KO	韓国語
PL	ポーランド語
RU	ロシア語

関連項目:

- [“ODSLANGCHG システムオプション” \(557 ページ\)](#)
- [“LOGLANGCHG システムオプション” \(552 ページ\)](#)
- [“LOGLANGENG システムオプション” \(553 ページ\)](#)

MAPEBCDIC2ASCII=システムオプション

EBCDIC から ASCII へのトランスコードと、ASCII から EBCDIC へのトランスコードで SAS ユーザーが使用する変換テーブルを指定します。

該当要素: 構成ファイル、SAS 起動時

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS GROUP= LANGUAGECONTROL

別名: MAPE2A

操作: MAPEBCDIC2ASCII=システムオプションは、SAS セッションに使用する変換テーブルを指定します。MAPEBCDIC2ASCII で指定されたテーブルは、\$EBCDIC 出力形式や \$ASCII 出力形式などのコンポーネントで使用されます。

構文

MAPEBCDIC2ASCII= TRANTAB *catalog-entry*

必須引数

catalog-entry

変換テーブルを含む SAS カタログの TRANTAB エントリを指定します。
entry-name.type を指定した場合、SAS は指定された名前を
SASUSER.PROFILE、SASHELP.LOCALE、SASHELP.HOST の順に検索します。

詳細

MAPEBCDIC2ASCII=は、各国語の要件をサポートします。

注意:

利用方法がよくわからない場合は変換テーブルを変更しないでください。変換テーブルは SAS により各国語サポートを提供するために内部的に利用されます。変換テーブルの利用方法がよくわからない場合や、技術的な支援がない場合は設定を変更しないでください。

NLDECSEPARATOR システムオプション

小数点にロケール依存の数値出力を生成するか、引き続き米国英語設定の数値形式を出力するかを指定します。

該当要素: 構成ファイル、SAS 起動時、OPTIONS ステートメント、[SAS システムオプション] ウィンドウ

**PROC OPTIONS
GROUP=** LANGUAGECONTROL

別名: NLD

デフォルト: NONLSDECSEPARATOR

構文

NLDECSEPARATOR | NONLDECSEPARATOR

必須引数

NLDECSEPARATOR

小数点区切り文字について、ロケール依存の数値出力を有効にします。

NONLDECSEPARATOR

小数点区切り文字について、ロケール依存の数値出力を無効にします。

詳細

小数点としてピリオドを使用する数値を生成する場合は、NLDECSEPARATOR オプションとともに BESTDOTX 出力形式を使用してください。BESTDOTX 出力形式を使用しないと、出力にカンマが含まれます(例: 1,2)。SAS は、この出力を数値とは解釈しません。この値が後続の代入ステートメントで使用されると、エラーになります。

CALL SYMPUT などの一部の SAS 関数は、数値の文字列への変換にデフォルトで BEST 出力形式を使用します。NLDECSEPARATOR が有効の場合、BEST 出力形式は現在のロケールの指定に応じた区切り文字を使用します。ロケールで区切り文字としてピリオドが使用されている場合、出力される文字列はこの値を数値として評価する以降の SAS ステートメントで使用できます。ただし、区切り文字がカンマなど他の文字である場合は、SAS コードがそれを数値として使用しようとしたときに構文エラーになります。

このタイプのエラーを避けるため、値の出力形式に BESTDOTX を使用してから CALL SYMPUT ステートメント内で代入してください。BESTDOTX は、出力で小数点として必ずピリオドを使用します。たとえば、次の SAS コードでは、変数 X の値の出力形式に BESTDOTX が使用されてから、変数 X の値がマクロ変数 &MACX に代入されています。&MACX の値が以降で SAS 変数 Y に代入されても、構文エラーになりません。

```
data a;
  x=1.2;
  CALL SYMPUT('macx', put(x, bestdotx.));
run;

data b;
  y=&macx
run;
```

関連項目:

[“BESTDOTX 出力形式” \(86 ページ\)](#)

ODSLANGCHG システムオプション

ODS 出力のテキスト言語を変更できるかを設定します。

該当要素: 構成ファイル、SAS 起動時

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS GROUP= LANGUAGECONTROL

デフォルト: UNICODE サーバー以外のサーバーでは、ODSLANGCHG は設定しません。

ヒント: 言語切り替え機能に使用される言語は、LSWLANG=オプションで制御されます。デフォルトでは、LSWLANG は LOCALE に設定され、言語切り替え機能の言語には LOCALE の言語が使用されます。

構文

ODSLANGCHG | NOODSLANGCHG

説明

ODSLANGCHG

起動後に ODS 出力の言語を変更できるようにする場合に指定します。

NOODSLANGCHG

起動後に ODS 出力の言語を変更できないようにする場合に指定します。

詳細

言語切り替え機能(LSW)により、起動後に SAS メッセージと ODS テンプレートの言語を変更できます。ODS 出力の言語切り替え機能を使用できるようにするには、ODSLANGCHG を指定する必要があります。

起動時は、構成ファイルに基づき、SAS メッセージと ODS テンプレートの言語が決定されます。ODSLANGCHG オプションが指定されると、地域化された SAS メッセージと ODS テンプレートが使用できる場合は、ODS 出力の言語は LSWLANG=設定に合わせて変更されます。

ユーザーは ODSLANGCHG は指定できますが、ロケール言語への翻訳はできません。たとえば、ODSLANGCHG を指定している場合に、フランス語の SAS セッションを開始してからロケールをギリシャ語に設定すると、NLDATE はギリシャ語で表示されますが、出力はフランス語で表示されます。SAS ではギリシャ語への翻訳は行っていないため、出力はフランス語で表示されます。ODSLANGCHG が指定されて LSWLANG=LOCALE が設定されている場合、ODS PATH は LOCALE 設定に基づいて更新されて、地域化されたテンプレートが含まれるようになります。

比較

また、その地域化されたメッセージが、ODS 出力の生成に使用されます。

起動時に ODSLANGCHG が指定されない場合、ODS 出力は構成ファイルに指定された言語で表示されます。

例

例 1 では、ODSLANGCHG が指定されていない(NOODSLANGCHG が指定された)フランス語のサーバーを考えます。

フランス語のクライアントアプリケーションがこのサーバーに接続すると、ODS 出力や、NL 出力形式を使用してフォーマットされた日付はフランス語で表示されます。英語のクライアントアプリケーションがこのサーバーに接続し、サーバーのロケールを英語に変更すると、ODS 出力はフランス語で表示され、NL 出力形式を使用してフォーマットされた日付は英語で表示されます。

例 2 では、ODSLANGCHG が指定され、LSWLANG=LOCALE が設定されたフランス語のサーバーを考えます。

フランス語のクライアントアプリケーションがこのサーバーに接続すると、ODS 出力や、NL 出力形式を使用してフォーマットされた日付はフランス語で表示されます。英語のクライアントアプリケーションがこのサーバーに接続し、サーバーのロケールを英語に変更すると、ODS 出力や、NL 出力形式を使用してフォーマットされた日付は英語で表示されます。

例 3 では、ODSLANGCHG が指定され、LSWLANG=English が設定されたフランス語のサーバーを考えます。

フランス語のクライアントアプリケーションがこのサーバーに接続すると、ODS 出力は英語で表示され、NL 出力形式を使用してフォーマットされた日付はフランス語で表示されます。英語のクライアントアプリケーションがこのサーバーに接続し、サーバーのロケールを英語に変更すると、ODS 出力や、NL 出力形式を使用してフォーマットされた日付は英語で表示されます。

関連項目:

- [“LOGLANGENG システムオプション” \(553 ページ\)](#)

- “[LOGLANGCHG システムオプション](#)” (552 ページ)
- “[LSWLANG システムオプション](#)” (554 ページ)

RSASIOTRANSERROR システムオプション

リモートアプリケーションから無効なデータが読み込まれたときにトランスコーディングエラーを表示します。

該当要素: 構成ファイル、SAS 起動時、OPTIONS ステートメント、**SAS システムオプション** ウィンドウ

カテゴリ: ファイル: SAS ファイル

PROC OPTIONS GROUP= SASFILES

デフォルト: RSASIOTRANSERROR

構文

RSASIOTRANSERROR | NORSASIOTRANSERROR

説明

RSASIOTRANSERROR

リモートアプリケーションから無効な値が読み込まれた場合に、トランスコーディングエラーを表示するように指定します。

NORSASIOTRANSERROR

リモートアプリケーションから無効な値が読み込まれた場合に、トランスコーディングエラーを表示しないように指定します。

詳細

RSASIOTRANSERROR システムオプションにより、SAS Enterprise Guide や SAS Enterprise Miner などの SASIO のリモートユーザーは、無効なデータ値を無視できます。通常は、無効なデータ値がリモートアプリケーションから読み込まれると、トランスコーディングエラーが発生します。

SORTSEQ=システムオプション

SORT および SQL プロシジャが現在の SAS セッションで使用する、言語固有の照合順序を指定します。

該当要素: 構成ファイル、SAS 起動時、OPTIONS ステートメント、**[SAS システムオプション]** ウィンドウ

カテゴリ: SORT: プロシジャオプション

PROC OPTIONS GROUP= SORT

構文

SORTSEQ= | <LINGUISTIC>

説明

LINGUISTIC

指定された言語のルールに従って文字を並べ替えます。ルールおよびデフォルトの照合順序は、現在のロケール設定で指定された言語に基づきます。実装は、International Components for Unicode (ICU)ライブラリによって提供されています。詳細については、[“Linguistic Sorting of Data Sets and ICU” \(SAS Viya Data Management and Utility Procedures Guide\)](#)を参照してください。

例

この例は、SORTSEQ と一緒に PROC SORT および PROC SQL を使用した場合の機能を示しています。

```
options sortseq=linguistic;
proc sort data=sashelp.class out=foo1;
by name;
run;
proc sql;
create table foo2 as select * from sashelp.class order by name;
quit;
run;
```

言語オプション(SORTSEQ=LINGUISTIC)が Base SAS プロシジャの PROC SORT で指定された場合、SAS システムは ICU 照合を提供します。SAS 9.4 の 3 番目のメンテナンスリリースから、SQL プロシジャ内で SORTSEQ=を使用し、SORTSEQ=LINGUISTIC システムオプションを指定することによっても、言語照合を指定できるようになりました。

注: SORTSEQ=LINGUISTIC システムオプションが指定された場合、PROC SORT と PROC SQL のみが影響を受けます。

言語オプションが指定された場合、SAS システムは ICU ライブラリを、Unicode 照合アルゴリズム(UCA)のリファレンス実装および事実上の標準として使用します。UCA アルゴリズムや国際化された Unicode コンポーネント(ICU)ライブラリ実装の詳細については、[Download the ICU 4.8 Release](#) および [CLDR 2.0 Release Note](#) を参照してください。

関連項目:

[“照合順序” \(14 ページ\)](#)

TIMEZONE=システムオプション

ユーザーローカルタイムゾーンを指定します。

該当要素: 構成ファイル、SAS コマンド、OPTIONS ステートメント、SASV9_OPTIONS 環境変数

カテゴリ: 環境コントロール: 言語コントロール

PROC OPTIONS GROUP= LANGUAGECONTROL

別名: TZ= (TIMEZONE=の使用が必須の制限されたオプション構成ファイル内は除く)

注: このオプションは、サイト管理者は制限できます。詳細については、[“Restricted Options” \(SAS Viya System Options: Reference\)](#)を参照してください。

構文

TIMEZONE='time-zone-name' | 'time-zone-ID'

構文の説明

time-zone-name

3 または 4 文字のタイムゾーン名を指定します。たとえば、EST は東部標準時間のタイムゾーン名です。

デフォルト BLANK。SAS サーバータイムゾーンとクライアントタイムゾーンが同じであることを示します。

参照項目: タイムゾーン名のリストについては、[付録 1, “タイムゾーン ID とタイムゾーン名” \(627 ページ\)](#)を参照してください。

time-zone-ID

SAS で定義されている *region/area* の値を指定します。タイムゾーン ID を指定したとき、SAS が使用するタイムゾーンはタイムゾーン名と夏時間ルールを考慮して決定されます。

注 タイムゾーン ID は JAVA タイムゾーン名と互換性があります。

参照項目: タイムゾーン ID のリストについては、[付録 1, “タイムゾーン ID とタイムゾーン名” \(627 ページ\)](#)を参照してください。

詳細

システムが特定のタイムゾーンを使用するように、TIMEZONE=オプションはタイムゾーン ID またはタイムゾーン名に対して設定します。タイムゾーン設定は次の SAS コンポーネントに影響します。

- イベントやログに記録される時間
- データセット作成や変更の時間
- DATE()関数
- DATETIME()関数
- TIME()関数
- TODAY()関数
- タイムゾーン関数 TZONEOFF()、TZONEID()、TZONENAME()、TZONES2U()、TZONEU2S()、
- タイムゾーン出力形式 B8601DXw.、E8601DXw.、B8601LXw.、E8601LXw.、B8601TXw.、E8601TXw.、NLDATMZw.、NLDATMTZw.、NLDATMWZw.

タイムゾーンを設定するには、タイムゾーン ID とタイムゾーン名を指定します。タイムゾーン ID では、リージョンとエリアをスラッシュ(/)で区切ります。たとえば、America/New_York や Asia/Osaka がタイムゾーン ID です。

タイムゾーン名は 3 または 4 文字のタイムゾーンの名前です。たとえば、EST は東部標準時間で JST は日本標準時間です。SAS は、時間値を使う前に夏時間を考慮したタイムゾーンルールに則って時間を決定します。

タイムゾーン名には異なるロケールで有効のものもあります。たとえば、CST は中部夏時間、キューバ夏時間、そして中国夏時間を意味します。SAS は LOCALE=

システムオプションの値を使って、どのリージョンとエリアを使えばいいのか判断します。TIMEZONE='CST'で LOCALE='zh_CN'の場合、SAS は Asia/Beijing タイムゾーンを使用します。タイムゾーン名がそのロケールに存在しない場合、システムはすべてのタイムゾーンを検索して最初にマッチしたタイムゾーンを設定します。

このオプションが制限されていて TIMEZONE=の値がデフォルト値の BLANK である場合、タイムゾーンの動作はタイムゾーン情報を使用しないものになります。

関連項目:

[“SAS でのタイムゾーンの指定” \(25 ページ\)](#)

VALIDMEMNAME=システムオプション

SAS データセット、SAS データビューおよびアイテムストアの命名規則を指定します。

該当要素: 構成ファイル、SAS コマンド、OPTIONS ステートメント、SASV9_OPTIONS 環境変数

カテゴリ: ファイル: SAS ファイル

PROC OPTIONS GROUP= SASFILES

デフォルト: 出荷時のデフォルト値は COMPATIBLE です。

適用対象: BASE エンジン

制限事項: VALIDMEMNAME の値に関係なく、メンバ名の最後に特殊文字#を付けて、その後に 3 桁の数字を続けることはできません。これは、世代データセットの命名規則と競合するためです。このようなメンバ名を使用すると、結果としてエラーが発生します。

操作: SAS Studio では、コードのサブミット前後に毎回 VALIDMEMNAME=COMPAT が設定されます。詳細については、“[System Options in SAS Studio](#)” ([SAS Viya System Options: Reference](#))を参照してください。

注: このオプションは、サイト管理者は制限できます。詳細については、“[Restricted Options](#)” ([SAS Viya System Options: Reference](#))を参照してください。

構文

VALIDMEMNAME=COMPATIBLE | EXTEND

構文の説明

COMPATIBLE

SAS データセット名、SAS データビュー名またはアイテムストア名が次の規則に従う必要があることを指定します。

- 名前の長さは、最大 32 文字まで指定できます。
- 名前は、英字(A から Z、a から z)またはアンダースコアで始める必要があります。開始文字以外には、英字、数字、アンダースコアを使用できません。
- 名前には、空白またはアンダースコア以外の特殊文字を含めることができません。

- 名前には大文字と小文字を混在させることができます。メンバ名は SAS 内部で大文字に変換されます。このため、メンバ名の大文字と小文字の組み合わせを変更して、異なる変数を表すことはできません。たとえば、**customer**、**Customer**、および **CUSTOMER** はすべて同じメンバ名を表します。ディスク上に名前がどのように保存されるかは、動作環境によって決まります。

別名 COMPAT

EXTEND

SAS データセット名、SAS データビュー名またはアイテムストア名が次の規則に従う必要があることを指定します。

- 名前には各国語文字を含めることができます。
- 名前にはハ * ? " < > | : - . 以外の特殊文字を含めることができます。
- 名前には、少なくとも 1 文字が必要です(文字、数字、有効な特殊文字、および各国語文字)。
- 名前の長さは、最大 32 バイトまで指定できます。
- NULL のバイトは使用できません。
- 名前は空白または '(ピリオド)で始めることはできません。
- メンバが作成されるときに先頭と末尾の空白は削除されます。
- 名前には大文字と小文字を混在させることができます。メンバ名は SAS 内部で大文字に変換されます。このため、メンバ名の大文字と小文字の組み合わせを変更して、異なる変数を表すことはできません。たとえば、**customer**、**Customer**、および **CUSTOMER** はすべて同じメンバ名を表します。名前がどのように表示されるかは、動作環境によって決まります。

要件 VALIDMEMNAME=EXTEND が設定されていて、SAS データセット名、SAS データビュー名およびアイテムストア名に空白や特殊文字や各国語文字が含まれる場合、それらは SAS 名リテラルとして書き込まれなければなりません。パーセント記号(%)またはアンパサンド(&)のいずれかを使用する場合、SAS マクロ機能との相互作用を避けるために、名前リテラルに一重引用符を使用する必要があります。

動作環境 SAS ファイルを物理名で直接参照する場合、最後の埋め込みピリオドが拡張子区切り文字になります。ファイルの物理参照でピリオドを含む SAS メンバ名を使う場合、ファイルの拡張子を必ず追加してください。たとえば、データセット名 my.member を物理ファイルとして参照する場合は、SET ステートメント set './saslib/my.member.sas7bdat' のように、参照名にファイル拡張子 sas7bdat を追加します。

ヒント 名前は大文字で表示されます。

例 data "August Purchases"n;

data 'Años de empleo'n.;

注意 SAS 全体で、名前リテラル構文に 32 バイト制限を超える SAS メンバ名を指定したり、埋め込まれている引用符が多すぎたりする場合、予期しない結果になる可能性があります。VALIDMEMNAME=EXTEND シス

テムオプションの目的は、埋め込み空白や各国語文字を許可するなど、他の DBMS メンバの命名規則との互換性を持たせることです。

詳細

VALIDMEMNAME=EXTEND の場合、SAS データセット名、SAS データビュー名およびアイテムストア名でが使用できる有効な文字は、次の文字まで拡張されます。

- 各国語文字
- サードパーティのデータベースでサポートされる文字
- ファイル名で一般的に使用される文字

DATA、VIEW および ITEMSTORE という SAS メンバの種類のみで、文字の拡張がサポートされます。CATALOG や PROGRAM などのその他のメンバの種類では、拡張文字はサポートされません。関連付けられた DATA メンバがある場合のみ存在する INDEX と AUDIT では、拡張文字がサポートされます。

関連項目:

システムオプション:

- [“VALIDVARNAME=システムオプション” \(564 ページ\)](#)

VALIDVARNAME=システムオプション

SAS セッション中に作成および処理可能な有効な SAS 変数名の規則を指定します。

該当要素:	構成ファイル、SAS コマンド、OPTIONS ステートメント、SASV9_OPTIONS 環境変数
カテゴリ:	ファイル: SAS ファイル
PROC OPTIONS GROUP=	SASFILES
デフォルト:	出荷時のデフォルト値は V7 です。
操作:	SAS Studio では、コードのサブミット前に毎回、プリファレンス設定に VALIDVARNAME= が設定されます。詳細については、 “System Options in SAS Studio” (SAS Viya System Options: Reference) を参照してください。
注:	このオプションは、サイト管理者は制限できます。詳細については、 “Restricted Options” (SAS Viya System Options: Reference) を参照してください。

構文

VALIDVARNAME=V7 | UPCASE | ANY

構文の説明

V7

変数名が次の規則に従う必要があることを指定します。

- SAS 変数名の長さは、最大 32 文字まで指定できます。

- 最初の文字には、英字(A から Z、a から z)またはアンダースコアを使用する必要があります。開始文字以外には、英字、数字、アンダースコアを使用できます。
- 末尾の空白は無視されます。変数名の配置は左揃えになります。
- 変数名には、空白またはアンダースコア以外の特殊文字を含めることができません。
- 変数名には大文字と小文字を混在させることができます。SAS では、変数を初めて参照した際に使用されていた大文字と小文字の組み合わせと同じ組み合わせで変数名の保存、書き込みが行われますが、変数名の処理時には、SAS 内部ではすべて大文字に変換されます。このため、変数名の大文字と小文字の組み合わせを変更して、異なる変数を表すことはできません。たとえば、**cat**、**Cat**、および **CAT** はすべて同じ変数を表します。
- 変数には、特殊な SAS 自動変数名(_N_、_ERROR など)や、変数リスト名(_NUMERIC_、_CHARACTER_、_ALL など)を割り当てないでください。

例 `season='summer';`

`percent_of_profit=percent;`

UPCASE

変数名が V7 と同じ規則に従うように指定します。ただし、SAS の以前のバージョンにある変数名が大文字という規則は除きます。

ANY

SAS 変数名が次の規則に従う必要があることを指定します。

- 名前には、空白、各国語文字、特殊文字、マルチバイト文字など、どの文字でも使用できます。
- 名前の長さは最大 32 バイトです。
- 名前に NULL のバイトを含めることはできません。
- 先頭の空白は保持されますが、末尾の空白は無視されます。
- 名前には、少なくとも 1 文字が必要です。すべて空白の名前は使用できません。
- 名前には大文字と小文字を混在させることができます。SAS では、変数を初めて参照した際に使用されていた大文字と小文字の組み合わせと同じ組み合わせで変数名の保存、書き込みが行われますが、変数名の処理時には、SAS 内部ではすべて大文字に変換されます。このため、変数名の大文字と小文字の組み合わせを変更して、異なる変数を表すことはできません。たとえば、**cat**、**Cat**、および **CAT** はすべて同じ変数を表します。

参照
 項
 目:

例 `'% of profit'n=percent;`

`'items@warehouse'n=itemnum;`

注意 SAS 全体で、名前リテラル構文に 32 バイト制限を超える SAS メンバ名を指定したり、埋め込まれている引用符が多すぎたりする場合、予期しない結果になる可能性があります。VALIDVARNAME=ANY システムオプションの目的は、埋め込み空白や各国語文字を許可するなど、他の DBMS 変数(列)の命名規則との互換性を持たせることです。

関連項目:

システムオプション:

- [“VALIDMEMNAME=システムオプション” \(562 ページ\)](#)

9 部

各国語サポート関連のコマンド、 ステートメント、プロシジャオプ ション

16 章

各国語サポート関連のコマンド、ステートメントおよび
プロシジャオプションのディクショナリ 569

16 章

各国語サポート関連のコマンド、 ステートメントおよびプロシジャ オプションのディクショナリ

カテゴリ別の各国語サポート関連のコマンド、ステートメント、およびプロシジャオプション	569
ディクショナリ	570
CHARSET=オプション	570
照合順序オプション	571
CVPBYTES=オプション、CVPENGINE=オプション、 CVPMULTIPLIER=オプション	577
ENCODING=オプション	583
INENCODING=/OUTENCODING=オプション	587
ODSCHARSET=オプション	588
ODSTRANTAB=オプション	589
PROC SQL の TRANSCODE=列修飾子	590
TRANSCODE=オプション	590
TRANTAB=オプション	593
XMLENCODING=オプション	593
TRANTAB ステートメント	594

カテゴリ別の各国語サポート関連のコマンド、ステートメント、およびプロシジャオプション

選択した SAS ステートメントのオプションに対するデータアクセスカテゴリとデータセットコントロールカテゴリは、各国語サポートの影響を受けます。次の表に、ステートメントオプションの概要を示します。詳細については、各ステートメントオプションのディクショナリエントリを参照してください。

カテゴリ	言語要素	説明
ODS: サードパーティによるフォーマティング	CHARSET=オプション (p. 570)	出力の META 宣言で生成される文字セットを指定します。
	TRANTAB=オプション (p. 593)	文字データをトランスコーディングする場合に使用する変換テーブルを指定します。
情報	TRANSCODE=オプション (p. 590)	文字変数をトランスコードするかどうかを指示する、ATTRIB ステートメントの属性(1 つ以上の変数に出力形式、入力形式、ラベル、長さに関連付ける)を指定します。

カテゴリ	言語要素	説明
データアクセス	CVPBYTES=オプション、 CVPENGINE=オプション、 CVPMULTIPLIER=オプション (p. 577)	SAS ファイルをトランスコードするために必要とされる文字変数の属性を指定します。
	ENCODING=オプション (p. 583)	外部ファイルの入力または出力処理のエンコーディングを優先指定してトランスコードします。
	INENCODING=/ OUTENCODING=オプション (p. 587)	SAS ライブラリの SAS データセットの読み込みまたは書き出し時のエンコーディングを無効にして変更します。
	ODSCHARSET=オプション (p. 588)	出力の META 宣言で生成される文字セットを指定します。
	ODSTRANTAB=オプション (p. 589)	出力ファイルの XML ドキュメントをトランスコードするときに使用する変換テーブルを指定します。
	XMLENCODING=オプション (p. 593)	外部ドキュメントのインポートまたはエクスポート時に、XML ドキュメントのエンコーディングより優先される値を指定します。

ディクショナリ

CHARSET=オプション

出力の META 宣言で生成される文字セットを指定します。

該当要素: ODS MARKUP ステートメントと ODS HTML ステートメントの LIBNAME ステートメント

カテゴリ: ODS: サードパーティによるフォーマティング

構文

CHARSET=*character-set* ;

必須引数

character-set

HTML 出力の META タグで使用する文字セットを指定します。

エンコーディングの例は UTF-8 です。インターネットで使用する公式の文字セットは、IANA (Internet Assigned Numbers Authority) によって登録されます。IANA は、ポート、プロトコルとエンタープライズ番号、オプション、コードとタイプなどのさまざまなインターネットプロトコルパラメータの中央レジストリです。文字セット値の一覧については、www.unicode.org/reports/tr22/index.html および www.iana.org/assignments/character-sets を参照してください。

character set は、このコンテキストの *encoding-value* に似ています。ただし、*character set* という用語は、インターネットでの使用に適したエンコーディングを識別するために使用されます。

例: ODS MARKUP ステートメントの META 宣言で生成される出力

```
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
```

関連項目:

概念的な情報:

- 3 章, “各国語サポート関連のエンコーディング” (9 ページ)

照合順序オプション

PROC SORT の照合順序を指定します。

該当要素: PROC SORT ステートメント

注: PROC SORT ステートメントは、SAS データセットのオブザベーションを 1 つ以上の文字変数または数値変数を基準に並べ替えます。

構文

```
PROC SORT collating-sequence-option <other option(s)> ;
```

オプション

オプションには、1 つの *collating-sequence-option* と複数の *other options* を含めることができます。2 種類のオプションの順序は重要でなく、両方の種類を同じ PROC SORT ステップに含める必要はありません。PROC SORT *collating-sequence-option* についてのみ次に説明します。

DANISH NORWEGIAN

デンマーク語とノルウェー語に従って文字を並べ替えます。

デンマーク語とノルウェー語の照合順序は、[図 16.1 \(572 ページ\)](#)に示されています。

POLISH

ポーランド語の規則に従って文字を並べ替えます。

FINNISH SWEDISH

フィンランド語とスウェーデン語の規則に従って文字を並べ替えます。フィンランド語とスウェーデン語の照合順序は、[図 16.1 \(572 ページ\)](#)に示されています。

NORWEGIAN

DANISH を参照してください。

SWEDISH

FINNISH を参照してください。

SORTSEQ=collating-sequence

照合順序を指定します。collating-sequence には、collating-sequence-option、変換テーブル、エンコーディング、キーワード LINGUISTIC のいずれかを指定できます。1つの照合順序のみ指定できます。詳細については、“照合順序” (14 ページ)を参照してください。

照合順序の説明を次に示します。

collating—sequence—option | translation table

SAS で提供される変換テーブルかユーザー定義の変換テーブル、または 1つの PROC SORT ステートメントの collating-sequence-option を指定します。

使用可能な変換テーブルは、次のとおりです。

- DANISH
- EBCDIC
- FINNISH
- ITALIAN
- NORWEGIAN
- POLISH
- REVERSE
- SPANISH
- SWEDISH

次の図に、各言語で英数字が並べ替えられる方法を示します。

図 16.1 英数字の各言語での並べ替え

Danish:	0123456789ABCDEF GHIJ KLMNOPQRST UVWXYZÆØÅabcde fghijklmnopqrstu vwxyzæøå
Finnish:	0123456789ABCDEF GHIJ KLMNOPQRST UVWXYZÄÅöabcde fghijklmnopqrstu vwxyzääö
Italian:	0123456789AÀBCCÐEÈÊËFGHIÌJ KLMNOÒPQRSTUÙVWXYZaàbcçdeéèfghii jklmnoðpqrstuùvwxyz
Norwegian:	0123456789ABCDEF GHIJ KLMNOPQRST UVWXYZÆØÅabcde fghijklmnopqrstu vwxyzæøå
Spanish:	0123456789AÁaáBbCcDdEÉeéFfGgHhIÍiíJjKkLlMmNnÑñOóoóPpQqRrSsTtUÚuúVvWwXxYyZz
Swedish:	0123456789ABCDEF GHIJ KLMNOPQRST UVWXYZÄÅöabcde fghijklmnopqrstu vwxyzääö

制限事項 PROC SORT ステップでは、1つの collating-sequence-option のみ指定できます。

ヒント SORTSEQ=照合順序オプションは、かっこを使用せず、引数に関連付けずに指定します。照合順序を指定する例を次に示します。
proc sort data=mydata SORTSEQ=ASCII;

encoding-value

エンコーディング値を指定します。この結果は、指定したエンコーディングで表示される文字データのバイナリ照合と同じです。サポートされるエンコーディング値は“データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値” (613 ページ)を参照してください。

制限事項 SORTSEQ=オプションで指定したエンコーディングを認識する SAS System の一部またはプロシジャは、PROC SORT のみです。

ヒント エンコーディング値に英数字またはアンダースコア以外の文字が含まれる場合、その値を引用符で囲む必要があります。

参照項目: “データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値” (613 ページ)で指定可能なエンコーディングの一覧。

LINGUISTIC<(collating—rules)>

指定した言語のルールに従って文字を並べ替える、言語照合を指定します。ルールおよびデフォルトの照合順序オプションは、現在のロケール設定で指定した言語に基づきます。International Components for Unicode (ICU)ライブラリにより実装され、Unicode 照合アルゴリズム (UCA)と大部分で互換性のある結果を作成します。

別名 UCA

制限事項 SORTSEQ=LINGUISTIC オプションは、PROC SORT の SORTSEQ=オプションおよび PROC SQL で使用できます。

注 また、SQL プロシジャ内で SORTSEQ=を使用することや、SORTSEQ=LINGUISTIC システムオプションを指定することによっても、言語照合を指定できるようになりました。詳細については、“SORTSEQ=システムオプション” (559 ページ)および “SORT Procedure” (*SAS Viya Data Management and Utility Procedures Guide*)を参照してください。

照合規則は、PROC SORT でのみサポートされており、システムオプションや PROC SQL の SORTSEQ=オプションではサポートされていません。

ヒント collating-rules はかっこで囲む必要があります。複数の照合ルールを指定できます。

言語照合で並べ替えたデータセットで BY 処理を実行する場合、データセットが適切に処理されるように NOBYSORTED システムオプションの指定が必要な場合があります。BY 処理は、照合順序処理とは異なる方法で実行されます。

参照項目: The “ICU License” (*SAS Viya Data Management and Utility Procedures Guide*)

言語照合の詳細については、“照合順序” (14 ページ)を参照してください。

Unicode 照合アルゴリズム(UCA)の仕様は、<http://www.unicode.org>で確認できます。

LINGUISTIC オプションで指定可能な collation-rules を示します。これらのルールにより言語照合順序が変更されます。

ALTERNATE_HANDLING=SHIFTED

スペース、句読点、記号などの変数文字の処理を制御します。このオプションを指定しない(デフォルト値の Non-Ignorable を使用する)場合、これらの変数文字間の違いは、通常の文字間の違いと同様に重要です。ALTERNATE_HANDLING オプションを指定する場合、これらの変数文字はあまり重要ではありません。

デフォルト NON_IGNOREABLE

ヒント SHIFTED 値は、QUATERNARY に設定された STRENGTH=と組み合わせてよく使用されます。このような場合、スペース文字、句読点、記号が文字列の比較時に考慮されますが、文字列のその他の要素(基本文字、アクセント記号、大文字と小文字)が同一の場合に限定されます。

CASE_FIRST=

大文字と小文字の順序を指定します。この引数は、TERTIARY、QUATERNARY または IDENTICAL レベルでのみ有効です。次の表に、CASE_FIRST 引数の値と情報を示します。

値	説明
UPPER	大文字、小文字の順序で並べ替えます。
LOWER	小文字、大文字の順序で並べ替えます。

COLLATION=

利用可能な COLLATION=値を次の表に示します。照合値を選択しない場合、ユーザーのロケールのデフォルト照合が選択されます。

値	説明
BIG5HAN	ラテン語にピンインの並べ替えを指定し、中国語、日本語、韓国語の文字に big5 文字セットの並べ替えを指定します。
DIRECT	ヒンディー語のバリエントを指定します。
GB2312HAN	ラテン語にピンインの並べ替えを指定し、中国語、日本語、韓国語の文字に gb2312han 文字セットの並べ替えを指定します。
PHONEBOOK	文字の並べ替えに電話帳スタイルを指定します。 PHONEBOOK はドイツ語でのみ選択します。
PINYIN	1文字ずつピンインに変換する方法を使用した中国語、日本語、韓国語の文字の並べ替えを指定します。この並べ替えは、簡体字中国語でよく使用されます。
POSIX	ポータブルオペレーティングシステムインターフェイスです。このオプションでは、文字の"C"ロケール並べ替えを指定します。

値	説明
STROKE	英字以外の書体の並べ替えを指定します。中国語、日本語、韓国語、ベトナム語で STROKE を選択します。この並べ替えは、繁体字中国語でよく使用されません。
TRADITIONAL	文字の並べ替えに従来のスタイルを指定します。たとえば、スペイン語で TRADITIONAL を選択します。

LOCALE=locale_name

POSIX 名(例: ja_JP)形式でロケール名を指定します。PROC SORT でサポートされるロケールと POSIX 値の一覧については、[“PAPERSIZE と DFLANG オプションの LOCALE=の値” \(599 ページ\)](#)を参照してください。

制限事項 次のロケールは PROC SORT でサポートされていません。

- Afrikaans_SouthAfrica, af_ZA
- Cornish_UnitedKingdom, kw_GB

NUMERIC_COLLATION=

数を示す文字のかわりに、数値でテキスト内の整数値を並べ替えます。

値	説明
ON	数値で番号を並べ替えます。たとえば、"8 Main St."は"45 Main St."より前になります。
OFF	文字の値で数を並べ替えます。たとえば、"45 Main St."は"8 Main St."より前になります。

デフォルト OFF

STRENGTH=

強度の値は、照合レベルに関連付けられます。5つの照合レベル値があります。次の表に、5つのレベルの情報を示します。強度のデフォルト値は、ロケールに関連付けられます。

値	照合の種類	説明
PRIMARY または 1	PRIMARY では、基本文字間の差異を指定します("a"<"b"など)。	この差異は最強です。たとえば、辞書は基本文字により異なるセクションに分割されません。

値	照合の種類	説明
SECONDARY または 2	文字のアクセント記号が第 2 差異となります("as"<"às"<"at"など)。	文字列内に第 1 差異がある場合は、第 2 差異は無視されず。文字間のその他の差異も、言語に応じて第 2 差異と考慮される場合があります。
TERTIARY または 3	大文字と小文字の差異は、第 3 レベルで区別されます("ao"<"Ao"<"aò"など)。	文字列内に第 1 または第 2 差異がある場合は、第 3 差異は無視されます。別の例として、通常の仮名と小文字の仮名の違いがあります。
QUATERNARY または 4	レベル 1 から 3 で句読点が無視される場合、句読点の有無で単語を区別するために追加レベルが使用されます("ab"<"a-b"<"aB"など)。	句読点の無視が必須の場合や日本語テキストを処理する場合は、第 4 レベルを使用する必要があります。第 1、第 2 または第 3 差異がある場合は、この差異は無視されます。
IDENTICAL または 5	他のすべてのレベルが等しい場合、最終的な決定をするために同一レベルが使用されます。各文字列の正規化形式 D (NFD) の Unicode コードポイント値がこのレベルで比較され、レベル 1 から 4 で違いがないことが確認されます。	2 つの文字列間でコードポイント値のみが異なることはほとんどないため、このレベルは必要最小限に使用します。たとえば、ヘブライ語の詠唱マークのみがこのレベルで区別されます。

別名 Level=

注意:

データの並べ替えにホストの並べ替えユーティリティを使用する場合、**SORTSEQ=**オプションで変換テーブルベースの照合順序を指定すると、**BY** 変数の文字が破損する可能性があります。詳細については、現在の作環境向けのドキュメントの **PROC SORT** を参照してください。

詳細

PROC SORT ステートメントの照合順序オプションは、**SAS** データセットのオプションを 1 つ以上の文字または数値変数で並べ替えます。

表 16.1 オプション

タスク	オプション
照合順序の指定	
デンマーク語の指定	DANISH (571 ページ)
フィンランド語の指定	FINNISH (571 ページ)
ノルウェー語の指定	NORWEGIAN (571 ページ)

タスク	オプション
ポーランド語の指定	POLISH (571 ページ)
スウェーデン語の指定	SWEDISH (571 ページ)
カスタマイズされた順序の指定	NATIONAL (572 ページ)
前述の照合順序(ASCII、EBCDIC、DANISH、FINNISH、ITALIAN、NORWEGIAN、POLISH、SPANISH、SWEDISH、NATIONAL)、システムで提供されたその他の変換テーブル名(POLISH、SPANISH)、ユーザーが作成した変換テーブル名のいずれかを指定します。エンコーディングを指定できます。また、ロケールに適切な照合順序にするために、キーワード LINGUISTIC または UCA を指定できます。	SORTSEQ= (572 ページ)

関連項目:

- [“照合順序” \(14 ページ\)](#)
- [“SORT Procedure” \(SAS Viya Data Management and Utility Procedures Guide\)](#)

システムオプション:

- [“SORTSEQ=システムオプション” \(559 ページ\)](#)

CVPBYTES=オプション、CVPENGINE=オプション、CVPMULTIPLIER=オプション

SAS ファイルをトランスコードするために必要とされる文字変数の属性を指定します。

該当要素: LIBNAME ステートメント

カテゴリ: データアクセス

参照項目: LIBNAME、SAS/ACCESS

構文

```
LIBNAME libref 'SAS data-library'; <CVPBYTES=bytes> <CVPENGINE=engine>
<CVPMULTIPLIER=multiplier> <CVPVARCHAR=YES | NO>
```

オプション引数

CVPBYTES=bytes

トランスコーディングを必要とする SAS データファイルを処理するときに、拡張する文字変数の長さをバイト数で指定します。CVP エンジン、長さを拡張することにより、文字データの切り捨てが発生しないようにします。文

文字変数の長さは、指定した値を現在の長さに追加して増やします。0 から 32766 までの値を指定できます。

たとえば、次の LIBNAME ステートメントでは、CVPBYTES=オプションを指定して CVP エンジンを暗黙的に割り当てています。

```
libname expand 'SAS data-library' cvpbytes=5;
```

文字変数の長さは、5 バイト追加して増やします。長さが 10 の文字変数は長さ 15 に、長さが 100 の文字変数は長さ 105 に増加します。

デフォルト CVPBYTES=を指定すると、指定値に従って文字変数の長さを拡張するために CVP エンジンを自動的に使用します。CVP エンジンを明示的に割り当てても、CVPBYTES=または CVPMULTIPLIER=を指定しない場合は、CVPMULTIPLIER=1.5 を使用して文字変数を長くします。

制限事項 CVP エンジンは、SAS データファイルをサポートしていますが、SAS のビュー、カタログおよびアイテムストアなどはサポートしていません。

CVP エンジンは、入力(読み込み)処理でのみ使用できます。

CVP エンジンなどのエンジンが混在するライブラリ連結では、SAS データファイルのみが処理されます。たとえば、COPY プロシジャを実行する場合は、SAS データファイルのみコピーされます。

要件 指定するバイト数は、どのような拡張にも適応できる大きさにする必要があります。この大きさが不十分だと、データが切り捨てられ、その結果 SAS ログにエラーメッセージが書き込まれます。

操作 CVPBYTES=と CVPMULTIPLIER=の両方を指定できません。いずれか 1 つのオプションを指定してください。

参照項目: [“CVP エンジンを使用した文字データ切り捨てへの対応” \(23 ページ\)](#)

CVPENGINE=engine

トランスコーディングを必要とする SAS データファイルを処理するために使用するエンジンを指定します。トランスコードする文字変数の長さが CVP エンジンによって拡張されるため、文字データの切り捨ては発生しません。次に、指定されたエンジンで実際のファイル処理を実行します。

別名 CVPENG

デフォルト SAS は、デフォルトの SAS エンジンを使用します。

参照項目: [“CVP エンジンを使用した文字データ切り捨てへの対応” \(23 ページ\)](#)

CVPMULTIPLIER=multiplier

トランスコーディングを必要とする SAS データファイルを処理するときに、文字変数の長さを拡張するための乗数値を指定します。CVP エンジンは、長さを拡張することにより、文字データの切り捨てが発生しないようにします。文字変数の長さは、指定した値を現在の長さに乗算して増やします。1 から 5 までの乗数値を指定できます。

たとえば、次の LIBNAME ステートメントでは、CVPMULTIPLIER=オプションを指定して CVP エンジンを暗黙的に割り当てています。

```
libname expand 'SAS data-library' cvpmultiplier=2.5;
```


文字変数の長さは 2.5 倍になります。長さが 10 の文字変数は長さ 25 に、長さが 100 の文字変数は長さ 250 に増加します。

別名 CVPMULT

デフォルト CVPMULTIPLIER=を指定すると、指定値に従って文字変数の長さを拡張するために CVP エンジンを実動的に使用します。CVP エンジンを明示的に割り当てても、CVPMULTIPLIER=または CVPBYTES=を指定しない場合は、CVPMULTIPLIER=1.5 を使用して文字変数を長くします。

制限事項 CVP エンジンは、SAS データファイルをサポートしていますが、SAS のビュー、カタログおよびアイテムストアなどはサポートしていません。

CVP エンジンは、入力(読み込み)処理でのみ使用できます。

CVP エンジンなどのエンジンが混在するライブラリ連結では、SAS データファイルのみが処理されます。たとえば、COPY プロシジャを実行する場合は、SAS データファイルのみコピーされます。

要件 指定するバイト数は、どのような拡張にも対応できる大きさにする必要があります。この大きさが不十分だと、データが切り捨てられ、その結果 SAS ログにエラーが書き込まれます。

操作 CVPMULTIPLIER=と CVPBYTES=の両方を指定できません。いずれか 1 つのオプションを指定してください。

参照項目: [“CVP エンジンを使用した文字データ切り捨てへの対応” \(23 ページ\)](#)

CVPVARCHAR=YES | NO

入力ファイル処理の際に、固定幅文字変数を可変幅文字に変換するか否かを指定します。固定幅文字変数のバイト長は、SAS セッションエンコーディングにおける 1 文字あたり最大バイト数に、指定された固定幅文字長を掛けた値です。

デフォルト No

操作 CVPVARCHAR=YES を指定した場合、CVPMULTIPLIER=オプションや CVPBYTES=オプションを指定しても無視されます。

注 CHAR カラム以下にある文字列データの末尾に空白があれば削除します。

TRANSCODE=NO という形式の固定幅文字変数は、変換の対象外です。

詳細

CVPVARCHAR=YES である場合、文字変数パディング(CVP、Character Variable Padding)エンジンは、CHAR で定義された変数を VARCHAR データタイプに変換します。CVP は入力処理専用なので、自動的に VARCHAR データタイプで保存されることはありません。

SAS セッションの終了時、データは保存されず、VARCHAR への変換結果も失われてしまいます。変更を保存したければ、SET ステートメントまたは PROC

COPY と、VARCHAR に対応したエンジンを使ってください。BASE エンジンなど、VARCHAR に非対応のエンジンでデータを保存した場合、新しいデータセットの文字カラムは CHAR データタイプに戻ってしまいます。新しいデータセットの CHAR 変数の長さは、VARCHAR の保存に必要なバイト数と同じです。

データを読み込むプロシジャが VARCHAR に対応していなければ、**CVPVARCHAR=YES** と指定しても無視されます。CVP エンジンは、CVPBYTES、CVPMULTIPLIER、またはデフォルトの乗数を使って、データ内の文字カラムの長さを拡張します。

例: CVP エンジンの使い方

次の例に、CVP エンジンを使用して文字データの切り捨てを回避する方法を示します。例では、Wlatin2 エンコーディングの一部の各国語文字を含む MYFILES.WLATIN2 という名前の SAS データセットを使用します。データセットは SBCS セッションで、SAS 9 内に作成されます。このコード例は SAS9 内で実行します。

```
libname myfiles 'SAS-Data Library';
data myfiles.wlatin2 (encoding=wlatin2);
  var1='A';
  var2='Š';
  var3='š';
  var4='†';
;
proc print data=myfiles.wlatin2;
run;
```

Obs	var1	var2	var3	var4
1	A	Š	š	†

これは、SAS 9 で実行した、MYFILES.WLATIN2 での PROC CONTENTS 出力です。エンコーディングが Wlatin2 で、各文字変数の長さが 1 バイトであることを示しています。

アウトプット 16.1 MYFILES.WLATIN2 の PROC CONTENTS 出力

The SAS System			
The CONTENTS Procedure			
Data Set Name	MYFILES.WLATIN2	Observations	1
Member Type	DATA	Variables	4
Engine	V9	Indexes	0
Created	08/17/2016 09:00:44	Observation Length	4
Last Modified	08/17/2016 09:00:44	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin2 Central Europe (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	15881
Obs in First Data Page	1
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	U:\wlatin2.sas7bdat
Release Created	9.0401M3
Host Created	W32_7PRO

Alphabetic List of Variables and Attributes			
#	Variable	Type	Len
1	var1	Char	1
2	var2	Char	1
3	var3	Char	1
4	var4	Char	1

次のコードは SAS Viya で実行してください。デフォルトセッションのエンコーディングは UTF-8 です。

```
options msglevel=i;
libname myfiles 'SAS data-library';
data myfiles.utf8;
  set myfiles.wlatin2;
run;
```

この DATA ステップでは、新しいデータセット MYFILES.UTF8 を要求します。さらに、データを UTF-8 エンコーディングのこの新しいデータセットに読み込む(つまり、Wlatin2 から UTF-8 へのデータのトランスコードが必要)ことを要求します。トランスコードの際に文字データの切り捨てが発生した場合、要求結果はエラーとなります。新規データセット MYFILES.UTF8 が作成されますが、データは何も含まれていません。

ログ 16.1 トランスコーディングエラーの SAS ログ

情報: データファイル MYFILES.UTF8.DATA の表示形式が別のホストにとってのネイティブであるか、ファイルエンコーディングがセッションエンコーディングと一致していません。クロス環境データアクセスが使用され、パフォーマンスの低下と追加の CPU リソースが必要となる場合もあります。ERROR: Some character data was lost during transcoding in the data set MYFILES.UTF8.NOTE: The data step has been abnormally terminated.NOTE: The SAS System stopped processing this step because of errors.NOTE: There were 1 observations read from the data set MYFILES.WLATIN2.WARNING: The data set MYFILES.UTF8 may be incomplete.このステップは、0 オブザベーション、4 変数で停止しました。

次のコードは SAS Viya で実行するもので、変数の長さを拡張します。

```
options msglevel=i;
libname myfiles 'SAS data-library';
libname expand cvp 'SAS data-library' cvpbytes=2;
data myfiles.utf8;
  set expand.wlatin2;
run;
```

この例で CVP エンジンでは、それぞれの長さに 2 バイトを追加して文字変数の長さを拡張します。データは、Wlatin2 から UTF-8 にトランスコードされることにより、UTF-8 エンコーディングで新規ファイルに読み込まれます。拡張された文字変数の長さが原因のデータの切り捨ては発生せず、新規データセットが正常に作成されます。

ログ 16.2 MYFILES.UTF8 の SAS ログ出力

```
57 options msglevel=i; 58 libname myfiles 'directory path'; NOTE: Libref MYFILES was
successfully assigned as follows: Engine: V9 Physical Name: /directory path 59 libname expand
cvp '/directory path' cvpbytes=2; WARNING: Libref EXPAND refers to the same physical library as
MYFILES.NOTE: Libref EXPAND was successfully assigned as follows: Engine: CVP Physical Name: /
directory path 60 data myfiles.utf8; 61 set expand.wlatin2; INFO: Data file
EXPAND.WLATIN2.DATA is in a format that is native to another host, or the file encoding does not match
the session encoding.Cross Environment Data Access will be used, which might require additional CPU
resources and might reduce performance.62 run; NOTE: There were 1 observations read from the
data set EXPAND.WLATIN2.NOTE: The data set MYFILES.UTF8 has 1 observations and 4 variables.NOTE:
DATA statement used (Total process time): real time 0.02 seconds cpu time 0.01 seconds
```

最後に示すのは MYFILES.UTF8 の PROC CONTENTS 出力です。これは、UTF-8 エンコーディングで表示され、各文字変数の長さは 3 です。

アウトプット 16.2 MYFILES.UTF8 の PROC CONTENTS 出力

The CONTENTS Procedure			
Data Set Name	MYFILES.UTF8	Observations	1
Member Type	DATA	Variables	4
Engine	V9	Indexes	0
Created	08/17/2016 09:32:34	Observation Length	12
Last Modified	08/17/2016 09:32:34	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	5401
Obs in First Data Page	1
Number of Data Set Repairs	0
Filename	/u/daharr/utf8.sas7bdat
Release Created	V.0301M0
Host Created	Linux
Inode Number	83031400
Access Permission	rw-r--r--
Owner Name	daharr
File Size	128KB
File Size (bytes)	131072

Alphabetic List of Variables and Attributes			
#	Variable	Type	Len
1	var1	Char	3
2	var2	Char	3
3	var3	Char	3
4	var4	Char	3

ENCODING=オプション

外部ファイルの入力または出力処理のエンコーディングを優先指定してトランスコードします。

該当要素: %INCLUDE ステートメント、FILE ステートメント、FILENAME ステートメント、FILENAME ステートメントの EMAIL (SMTP)アクセス方式、INFILE ステートメント、ODS ステートメント、FILE コマンド、INCLUDE コマンド

カテゴリ: データアクセス

構文

ENCODING= 'encoding-value'

オプション引数

ENCODING= '*encoding-value*'

外部ファイルの読み込み、書き出し、コピー、保存に使用するエンコーディングを指定します。ENCODING=の値は、外部ファイルのエンコーディングが現在のセッションエンコーディングとは異なることを示します。

外部ファイルを使用してデータの読み込み、書き出し、コピー、保存を行うと、SAS はデータをセッションエンコーディングから指定されたエンコーディングにトランスコードします。

詳細については、“データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値” (613 ページ)を参照してください。

デフォルト SAS は、現在のセッションエンコーディングを使用します。

詳細

次のテーブルでは、ENCODING オプションが対応するステートメントでどのように使われるかについて説明します。

%INCLUDE ステートメント:	SAS ステートメントおよびデータ行の指定したソースファイルからの読み込み。
FILE ステートメント:	外部ファイルへの書き出し
FILENAME ステートメント:	外部ファイルからの読み込み、外部ファイルへの書き出し
FILENAME ステートメントの EMAIL (SMTP)アクセス方式:	プログラムによる電子メール送信
INFILE ステートメント:	外部ファイルからの読み込み
ODS ステートメント:	SAS プロシジャおよび DATA ステップの出力の生成、保存または再作成に使用する Output Delivery System (ODS)機能の制御
FILE コマンド:	ウィンドウのコンテンツの外部ファイルへの保存
INCLUDE コマンド:	外部ファイルの現在のウィンドウへのコピー

例

例 1: FILE ステートメントを使用した外部ファイルへの書き出しのエンコーディングの指定

この例では、SAS データセットから外部ファイルを作成します。現在のセッションエンコーディングは UTF-8 ですが、外部ファイルのエンコーディングは LATIN1 にする必要があります。デフォルトで、SAS は外部ファイルを現在のセッションエンコーディングを使用して書き出します。

外部ファイルへのデータの書き出しに使用するエンコーディングを指定するには、ENCODING=オプションを次のように指定します。

```
libname myfiles 'SAS data-library';
filename outfile 'external-file';
data _null_;
  set myfiles.cars;
  file outfile encoding="latin1";
  put Make Model Year;
run;
```

例 2: FILENAME ステートメントを使用した外部ファイルの読み込みのエンコーディングの指定

この例では、外部ファイルから SAS データセットを作成します。外部ファイルは WLATIN1 文字セットエンコーディングで、現在の SAS セッションは UTF-8 エンコーディングです。デフォルトでは、SAS は外部ファイルがセッションエンコーディングと同じエンコーディングであると想定します。この場合、文字データは新しい SAS データセットに正しく書き出されません。

外部ファイルの読み込み時に使用するエンコーディングを指定するには、ENCODING=オプションを次のように指定します。

```
libname myfiles 'SAS data-library';

filename extfile 'external-file' encoding="utf-8";
data myfiles.unicode;
  infile extfile;
  input Make $ Model $ Year;
run;
```

外部ファイルが WLATIN1 エンコーディングであると指定すると、SAS は、新しい SAS データセットへの書き出し時に外部ファイルを WLATIN1 から現在のセッションエンコーディングにトランスコードします。そのため、データは新しいデータセットに UTF-8 で正しく書き出されます。

例 3: FILENAME ステートメントを使用した外部ファイルへの書き出しのエンコーディングの指定

この例では、SAS データセットから外部ファイルを作成します。デフォルトで、SAS は外部ファイルを現在のセッションエンコーディングを使用して書き出します。現在のセッションエンコーディングは UTF-8 ですが、外部ファイルのエンコーディングは WLATIN1 にする必要があります。

外部ファイルへのデータ書き出し時に使用するエンコーディングを指定するには、ENCODING=オプションを次のように指定します。

```
libname myfiles 'SAS data-library';
filename outfile 'external-file' encoding="WLATIN1";
data _null_;
  set myfiles.cars;
  file outfile;
  put Make Model Year;
run;
```

外部ファイルを WLATIN1 エンコーディングにするように指定すると、SAS は、外部ファイルへの書き出し時にデータを UTF-8 から指定された Wlatin1 エンコーディングにトランスコードします。

例 4: メッセージ本文と添付ファイルのエンコーディングの変更

この例では、メッセージ本文と添付ファイルのテキストエンコーディングを変更する方法を説明します。

```
filename mymail email 'Joe.Developer@sas.com';
data _null_;
  file mymail
    subject='Text Encoding'
    encoding=greek
    attach=('C:\My Files\Test.out'
    content_type='text/plain'
    encoding='ebcdic1047'
    outencoding='latin1');
run;
```

プログラムでは、次の操作が実行されます。

- ENCODING=メールオプションで、メッセージ本文を送信前にギリシャ語 (ISO)でエンコードするように指定します。
- ATTACH=メールオプションの場合、添付ファイルオプション ENCODING= で、SAS に読み込まれる添付ファイルのエンコーディングを西ヨーロッパ諸語 (EBCDIC)に指定します。
- SMTP やその他のメールインターフェイスは EBCDIC をサポートしないため、添付ファイルオプション OUTENCODING=によって、添付ファイルは送信前に西ヨーロッパ諸語 (ISO)に変換されます。

例 5: INFILE=ステートメントを使用した外部ファイル読み込みのエンコーディングの指定

この例では、外部ファイルから SAS データセットを作成します。外部ファイルのエンコーディングは WLATIN2 で、現在の SAS セッションエンコーディングは UTF-8 です。デフォルトでは、SAS は外部ファイルがセッションエンコーディングと同じエンコーディングであると想定します。この場合、文字データは新しい SAS データセットに正しく書き出されません。

外部ファイルの読み込み時に使用するエンコーディングを指定するには、ENCODING=オプションを次のように指定します。

```
libname myfiles 'SAS data-library';
filename extfile 'external-file';
data myfiles.unicode;
  infile extfile encoding="utf-8";
  input Make $ Model $ Year;
run;
```

外部ファイルが WLATIN2 エンコーディングであると指定すると、SAS は、新しい SAS データセットへの書き出し時に外部ファイルを WLATIN2 から現在のセッションエンコーディングにトランスコードします。そのため、データは新しいデータセットに UTF-8 で正しく書き出されます。

関連項目:**ステートメント:**

- [“FILE Statement” \(SAS Viya Statements: Reference\)](#)
- [“FILENAME Statement” \(SAS Viya Statements: Reference\)](#)
- [“INFILE Statement” \(SAS Viya Statements: Reference\)](#)

INENCODING=/OUTENCODING=オプション

SAS ライブラリの SAS データセットの読み込みまたは書き出し時のエンコーディングを無効にして変更します。

該当要素: LIBNAME ステートメント

カテゴリ: データアクセス

構文

INENCODING= ANY | ASCIIANY | EBCDICANY | *encoding-value*

OUTENCODING= ANY | ASCIIANY | EBCDICANY | *encoding-value*

説明

ANY

ASCII エンコーディングと EBCDIC エンコーディング間でトランスコーディングを行わないように指定します。

NOTE: ANY は BINARY の指定と同等です。データはバイナリであるため、実際のエンコーディングには意味がありません。

ASCIIANY

混在したエンコーディングが ASCII エンコーディングと想定して、トランスコーディングを行わないように指定します。

EBCDICANY

混在したエンコーディングが EBCDIC エンコーディングと想定して、トランスコーディングを行わないように指定します。

encoding-value

エンコーディング値を指定します。エンコーディング値の一覧については、[18 章, "SAS 言語要素中のエンコーディング値" \(613 ページ\)](#)を参照してください。

詳細

INENCODING=オプションは、SAS ライブラリの SAS データセットの読み込みに使用されます。OUTENCODING=オプションは、SAS ライブラリの SAS データセットの書き出しに使用されます。

INENCODING=値または OUTENCODING=値は、LIST 引数を使用する場合、SAS ログに書き出されます。

INENCODING=および OUTENCODING=は、混在したエンコーディングが含まれる既存のライブラリを使用する場合に、最も適しています。混在したエンコーディングが含まれるライブラリを読み取るには、INENCODING=を ASCIIANY または EBCDICANY に設定できます。別個のデータセットを書き出すには、OUTENCODING=を使用して特定のエンコーディングを指定できます。このエンコーディングがデータセットの作成時に適用されます。

比較

- セッションエンコーディングは、ENCODING=システムオプションを使用して指定します。

- SAS ライブラリのデータセットを読み取るためのエンコーディングを指定するには、LIBNAME ステートメントの INENCODING=オプションを入力ファイルに使用します。LIBNAME ステートメントオプションと ENCODING=データセットオプションを両方とも指定すると、データセットオプションが使用されます。
- SAS ライブラリにデータセットを書き出すためのエンコーディングを指定するには、LIBNAME ステートメントの OUTENCODING=オプションを出力ファイルに使用します。LIBNAME ステートメントオプションと ENCODING=データセットオプションを両方とも指定すると、データセットオプションが使用されます。
- COPY プロシジャの場合、デフォルトの CLONE オプションでは、OUTENCODING=オプションで指定したエンコーディング値ではなく、入力データセットのエンコーディング属性を使用します。

関連項目:

- [“概要: 各国語サポート関連のエンコーディング” \(9 ページ\)](#)

ステートメント:

- [“LIBNAME Statement” \(SAS Viya Statements: Reference\)](#)

システムオプション:

- [“ENCODING システムオプション” \(549 ページ\)](#)
- [“LOCALE システムオプション” \(550 ページ\)](#)

データセットオプション:

- [“ENCODING=データセットオプション” \(41 ページ\)](#)

ODSCHARSET=オプション

出力の META 宣言で生成される文字セットを指定します。

該当要素: XMLV2 エンジンの LIBNAME ステートメント

カテゴリ: データアクセス

構文

ODSCHARSET=*character-set*;

必須引数

character-set

XML エンジンの LIBNAME ステートメントでは、ENCODING=属性で使用する文字セットを指定します。

エンコーディングの例は ISO-8859-1 です。インターネットで使用する公式の文字セットは、IANA (Internet Assigned Numbers Authority) によって登録されます。IANA は、ポート、プロトコルとエンタープライズ番号、オプション、コードとタイプなど、さまざまなインターネットプロトコルパラメー

タの中央レジストリです。文字セット値の一覧については、www.unicode.org/reports/tr22/index.html および www.iana.org/assignments/character-sets を参照してください。

character set は、このコンテキストの *encoding-value* に似ています。ただし、*character set* という用語は、インターネットでの使用に適したエンコーディングを識別するために使用されます。

詳細

XML ドキュメントによっては、XML 宣言が不要なものもあります。XML 宣言は、ドキュメントの文字エンコーディングがデフォルトの UTF-8 または UTF-16 以外で、上位プロトコルでエンコーディングが決まっていなかった場合にのみ必要となります。

XML エンジンの LIBNAME ステートメントの ODSCHARSET オプションで出力する XML ドキュメントに使う文字セットを指定します。

関連項目:

概念的な情報:

- [3 章, “各国語サポート関連のエンコーディング” \(9 ページ\)](#)

ステートメント:

- *SAS Viya XML LIBNAME Engine: ユーザーガイド*

ODSTRANTAB=オプション

出力ファイルの XML ドキュメントをトランスコードするときに使用する変換テーブルを指定します。

該当要素: XMLV2 エンジンの LIBNAME ステートメント

カテゴリ: データアクセス

構文

TRANTAB ='translation-table'

オプション引数

translation-table

出力ファイルに使用する変換テーブルを指定します。変換テーブルは、文字セットの文字(通常の文字、表語文字、数字、句読点、記号、コントロール文字など)を数値にマッピングするエンコーディング方法です。変換テーブルの例として、文字を EBCDIC から ASCII-ISO に変換するものがあります。*table-name* には、SAS で提供される変換テーブルまたはユーザー定義の変換テーブルを指定できます。値は、SASUSER.PROFILE カタログまたは SASHELP.HOST カタログの SAS カタログエントリ名である必要があります。

詳細

エンコーディングを指定する場合、LOCALE=システムオプションを使用する方法が推奨されます。

関連項目:**概念的な情報:**

- [2 章, “各国語サポート関連のロケール” \(5 ページ\)](#)の概念の説明

システムオプション:

- [“LOCALE システムオプション” \(550 ページ\)](#)

ステートメント:

- *SAS Viya XML LIBNAME Engine: ユーザーガイド*

PROC SQL の TRANSCODE=列修飾子

文字列に値をトランスコードできるかどうかを指定します。

該当要素: SQL プロシジャの列修飾子の構成要素

構文

TRANSCODE=YES|NO

必須引数

TRANSCODE=YES|NO

文字列について、値をトランスコードできるかどうかを指定します。トランスコーディングを実行しないようにするには、TRANSCODE=NO を使用します。CREATE TABLE AS ステートメントを使用してテーブルを作成する場合、作成されるテーブルの特殊文字の列のトランスコーディング属性は、TRANSCODE=列修飾子を使用して変更しない限り、入力元テーブルの属性と同じであることに注意してください。

デフォルト YES

制限事項	トランスコーディングの抑止は、V6TAPE エンジンではサポートされていません。
-------------	--

関連項目:

- [4 章, “各国語サポート関連のトランスコーディング” \(19 ページ\)](#)
- *SAS Viya Data Management and Utility Procedures Guide*

TRANSCODE=オプション

文字変数をトランスコードするかどうかを指示する、ATTRIB ステートメントの属性(1 つ以上の変数に出力形式、入力形式、ラベル、長さに関連付ける)を指定します。

該当要素: DATA ステップの ATTRIB ステートメント

カテゴリ: 情報

種類: 宣言型
参照項目: ATTRIB ステートメント “ATTRIB Statement” (*SAS Viya Statements: Reference*).

構文

ATTRIB *variable-list(s) attribute-list(s)* ;

必須引数

variable-list

属性に関連付ける変数の名前を付けます。

ヒント SAS で許可される形式の変数リストが表示されます。

attribute-list

variable-list に割り当てる属性を 1 つ以上指定します。ATTRIB ステートメントには複数の属性を指定できます。属性の一覧については、“ATTRIB Statement” (*SAS Viya Statements: Reference*) を参照してください。

TRANSCODE= YES | NO

文字変数をトランスコードするかどうかを指定します。トランスコーディングを実行しないようにするには、TRANSCODE=NO を使用します。詳細については、“トランスコーディングの概要” (19 ページ) を参照してください。

デ
フ
ォ
ル
ト

制限事項 SAS Workspace Server クライアントによっては、TRANSCODE=NO 属性がサポートされない場合があります。TRANSCODE=NO が指定された変数は、SAS Viya では返されません。SAS 9.4 以前のリリースでは、TRANSCODE=NO が指定された変数はトランスコードされます。SAS の旧リリースでは、TRANSCODE=NO 属性が指定された変数を含む SAS 9.4 データセットにアクセスできません。

操作 VTRANSCODE 関数および VTRANSCODEX 関数を使用すると、文字変数についてトランスコーディングがオンとオフのどちらであるかを返すことができます。

TRANSCODE=属性が NO に設定された文字変数がデータセット内にあると、PROC CONTENTS は、データセット内の変数ごとに、TRANSCODE=値が含まれるトランスコーディング列を出力します。データセット内のすべての変数がデフォルトの TRANSCODE=値 (YES) に設定された場合、トランスコーディング列は出力されません。

例

例 1: TRANSCODE=オプションと SET ステートメントの併用

SET ステートメントを使用して複数のデータセットからデータセットを 1 つ作成すると、出力データセット内の変数の TRANSCODE=属性が、最初のデータセ

ット内の変数の TRANSCODE=値と同じに設定されます。この例では、B が最初のデータセットで、データセット B 内の Z の TRANSCODE=属性が NO であるため、データセット A 内の変数 Z の TRANSCODE=属性は NO です。

```
data b;
  length z $4;
  z = 'ice';
  attrib z transcode = NO;
data c;
  length z $4;
  z = 'snow';
  attrib z transcode = YES;
data a;
  set b;
  set c;
  /* Check transcode setting for variable Z */
  rc1 = vtranscode(z);
  put rc1=;
run;
```

例 2: TRANSCODE=オプションと MERGE ステートメントの併用

MERGE ステートメントを使用して複数のデータセットからデータセットを 1 つ作成すると、出力データセット内の変数の TRANSCODE=属性が、最初のデータセット内の変数の TRANSCODE=値と同じに設定されます。この例では、C が最初のデータセットで、データセット C 内の Z の TRANSCODE=属性が YES であるため、データセット A 内の変数 Z の TRANSCODE=属性は YES です。

```
data b;
  length z $4;
  z = 'ice';
  attrib z transcode = NO;
data c;
  length z $4;
  z = 'snow';
  attrib z transcode = YES;
data a;
  merge c b;
  /* Check transcode setting for variable Z */
  rc1 = vtranscode(z);
  put rc1=;
run;
```

注: TRANSCODE=属性は、変数が入力データセットまたは ATTRIB TRANSCODE=ステートメントで最初に検出されたときに設定されます。SET または MERGE ステートメントが、ATTRIB TRANSCODE=ステートメントよりも前に現れ、TRANSCODE=属性が SET ステートメントと矛盾する場合、エラーメッセージが表示されます。

関連項目:

関数:

- [“VTRANSCODE 関数” \(399 ページ\)](#)
- [“VTRANSCODEX 関数” \(400 ページ\)](#)

TRANTAB=オプション

文字データをトランスコーディングする場合に使用する変換テーブルを指定します。

該当要素: ODS MARKUP ステートメントと ODS RTF ステートメント

カテゴリ: ODS: サードパーティによるフォーマティング

構文

TRANTAB = (*translation-table*)

オプション引数

translation-table

出力ファイルに使用する変換テーブルを指定します。変換テーブルは、文字セットの文字(通常の文字、表語文字、数字、句読点、記号、コントロール文字など)を数値にマッピングするエンコーディング方法です。変換テーブルの例として、文字を EBCDIC から ASCII-ISO に変換するものがあります。*table-name* には、SAS で提供される変換テーブルまたはユーザー定義の変換テーブルを指定できます。値は、SASUSER.PROFILE カタログまたは SASHELP.HOST カタログの SAS カタログエントリ名である必要があります。

詳細

エンコーディングを指定する場合は ENCODING=の使用をお勧めします。

XMLENCODING=オプション

外部ドキュメントのインポートまたはエクスポート時に、XML ドキュメントのエンコーディングより優先される値を指定します。

該当要素: XML エンジンの LIBNAME ステートメント

カテゴリ: データアクセス

構文

XMLENCODING = '*encoding-value*'

詳細

XML エンジンの LIBNAME ステートメントは、外部ドキュメントをインポートまたはエクスポートするために、SAS ライブラリ参照名を XML ドキュメントに関連付けます。

比較

オプション

encoding-value

外部ファイルの読み込み、書き出し、コピー、保存に使用するエンコーディングを指定します。XMLENCODING=の値は、外部ファイルのエンコーディングが現在のセッションエンコーディングとは異なることを示します。

詳細については、“データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値” (613 ページ)を参照してください。

encoding-value のデフォルトは現在のセッションのエンコーディングです。

関連項目:**ステートメント:**

- *SAS Viya XML LIBNAME Engine: ユーザーガイド*

TRANTAB ステートメント

SAS ファイルをエクスポートまたは転送するための文字データのトランスコーディング時に使用する、変換テーブルを指定します。

該当要素: CPORT プロシジャ、UPLOAD プロシジャ、DOWNLOAD プロシジャ

制限事項: 1 つの TRANTAB ステートメントに対して指定できる変換テーブルは 1 つのみです。追加の変換テーブルを指定するには、追加の TRANTAB ステートメントを使用します。

操作: TRANTAB ステートメントでは、エクスポートまたは転送される SAS ファイルの文字セットに適用する、カスタマイズした変換テーブル(たとえば、EBCDIC 文字の ASCII 文字へのマッピング)を指定します。TRANTAB=システムオプションでは、ファイル転送を含む、SAS セッションで使用される変換テーブルを指定します。

構文

```
TRANTAB NAME=translation-table-name <TYPE=(etype-list) <OPT=DISP | SRC | (DISP SRC)> >;
```

必須引数**NAME=*translation-table-name***

エクスポート(PROC CPORT)または転送(PROC UPLOAD または PROC DOWNLOAD)する SAS カタログに適用する変換テーブルの名前を指定します。SASUSER.PROFILE カタログまたは SASHELP.HOST カタログのカタログエントリの名前として指定する *translation-table-name*SASUSER.PROFILE カタログが先に検索され、SASHELP.HOST カタログはその次に検索されます。

大半はデフォルトの変換テーブルが使用されますが、アプリケーションに異なる各国語文字が必要な場合など、追加の変換テーブルの適用が必要になることもあります。

デフォルト以外の変換テーブルを指定するには、次の 2 つの方法があります。

- プロシジャの起動時に変換テーブルを指定するには、必要に応じて、プロシジャで TRANTAB ステートメントを使用します。

- SAS セッションまたはジョブ全体(すべてのファイルエクスポートまたは転送を含む)に変換テーブルを指定するには、TRANTAB=システムオプションを使用します。

オプション引数

TYPE=(etype-list)

指定したタイプのエントリにのみ変換テーブルを適用します。etype-list には、1つ以上のエントリのタイプを指定できます。たとえば、カタログエントリのタイプには、DATA や FORMAT などがあります。etype-list に指定するエントリタイプが1つのみの場合、かっこを省略します。

デフォルトで、UPLOAD、DOWNLOAD および CPORT プロシジャは、指定したすべてのカタログエントリに変換テーブルを適用します。

OPT=DISP | SRC | (DISP SRC)

OPT=DISP 指定したカタログエントリのうち、ウィンドウ表示を作成するエントリにのみ変換テーブルを適用します。

OPT=SRC 指定したカタログエントリのうち、タイプが SOURCE のエントリにのみ変換テーブルを適用します。

OPT=(DISP SRC) 指定したカタログエントリのうち、ウィンドウ表示を作成するか、タイプが SOURCE のエントリにのみ変換テーブルを適用します。

OPT=オプションを指定しない場合、UPLOAD または DOWNLOAD プロシジャは、変換テーブルを指定したカタログ内のすべてのエントリに適用します。

デフォルト PROC CPORT、PROC UPLOAD および PROC DOWNLOAD は、変換テーブルを指定したカタログ内のすべてのエントリとデータセットに適用します。

詳細

変換テーブルは、各国語の要件をサポートするために SAS 6 で導入されました。SAS 8.2 では、変換テーブルの直接使用より優れた、LOCALE=システムオプションが導入されました。SAS 9.3 では、TRANTAB ステートメントが下位互換性を維持するためにサポートされています。ただし、SAS 8.2 以降のリリースでは、LOCALE=システムオプションを使用することをお勧めします。詳細については、TS-639:Data Conversion Issues in V6-V8 を参照してください。このテクニカルサポートの注記では、英語以外の言語を使用している場合について説明しています。<http://support.sas.com/techsup/technote/ts639.pdf>

PROC CPORT はネットワークを介して SAS ファイルをエクスポートする場合に使用します。PROC UPLOAD と PROC DOWNLOAD はネットワークを介して SAS ファイルを転送する場合に使用します。

TRANTAB ステートメントの使用時、PROC UPLOAD または PROC DOWNLOAD ステートメントに INCAT=および OUTCAT=オプションを指定する必要があります。

例

例 1

次の情報をプロシジャ機能に適用します。

- PROC CPORT ステートメントオプション: FILE=
- TRANTAB ステートメントオプション: TYPE=

この例では、カスタマイズした変換テーブルを、PROC CPORT がエクスポートする前にトランスポートファイルに適用する方法を示します。この例では、カスタマイズした変換テーブル TTABLE1 はすでに作成されたと想定します。

例 2: プログラム

ライブラリ参照を割り当てます。</userSuppliedValue>LIBNAME ステートメントと FILENAME ステートメントは、ソースライブラリのライブラリ参照名とトランスポートファイルのファイル参照名をそれぞれ割り当てます。

```
libname source
'\sashq\root\pub\pubdoc\doc\901\authoring\proc\miscsrc\sasfiles\cport';
filename tranfile 'trans3';
proc trantab table=ascii;
save table=ttable1;
```

```
libname source 'SAS data-library';
filename tranfile 'transport-file'
  host-option(s)-for-file-characteristics;
```

変換の詳細を適用します。</userSuppliedValue>TRANTAB ステートメントは、カスタマイズされた変換テーブル TTABLE1 で指定した変換を適用します。TYPE=によって、変換の適用対象が FORMAT エントリのみ制限されます。

```
proc cport catalog=source.formats file=tranfile;
  trantab name=ttable1 type=(format);
run;
```

例 3: SAS ログ

NOTE: Proc CPORT はカタログ SOURCE.FORMATS の移送を開始します。NOTE: カタログのエントリ数は 2 です。最大レコード長は 104 です。NOTE: エントリ REVENUE.FORMAT を移送します。NOTE: エントリ DEPT.FORMATC を移送します。

関連項目:

概念的な情報:

- 4 章, “各国語サポート関連のトランスコーディング” (19 ページ)

10 部

ロケールとトランスコーディングの値

17 章	
LOCALE=システムオプションの値	599
18 章	
SAS 言語要素中のエンコーディング値	613

17 章

LOCALE=システムオプションの値

PAPERSIZE と DFLANG オプションの LOCALE=の値 599

PAPERSIZE と DFLANG オプションの LOCALE=の値

次の表に、SAS 名または Posix 名を使用して指定した有効な LOCALE=値を表示します。エイリアス名も示されています。エイリアス名のないロケールもあります。

表 17.1 LOCALE=システムオプションの値

SAS 名	Posix ロケール	エイリアス
Afrikaans_SouthAfrica	af_ZA	Afrikaans af
Albanian_Albania	sq_AL	Albanian sq
Arabic_Algeria	ar_DZ	
Arabic_Bahrain	ar_BH	
Arabic_Egypt	ar_EG	
Arabic_India	ar_IN	
Arabic_Iraq	ar_IQ	
Arabic_Jordan	ar_JO	
Arabic_Kuwait	ar_KW	
Arabic_Lebanon	ar_LB	
Arabic_Libya	ar_LY	

SAS名	Posix ロケール	エイリアス
Arabic_Morocco	ar_MA	
Arabic_Oman	ar_OM	
Arabic_Qatar	ar_QA	
Arabic_SaudiArabia	ar_SA	
Arabic_Sudan	ar_SD	
Arabic_Syria	ar_SY	
Arabic_Tunisia	ar_TN	
Arabic_UnitedArabEmirates	ar_AE	Arabic ar
Arabic_Yemen	ar_YE	
Basque_Spain	eu_ES	Basque Euskara eu
Bengali_India	bn_IN	Bengali bn
Bosnian_BosniaHerzegovina	bs_BA	Bosnian bs
Bulgarian_Bulgaria	bg_BG	Bulgarian bg
Byelorussian_Belarus	be_BY	Byelorussian Belarusian Byelorussian_Belarus be
Catalan_Spain	ca_ES	Catalan ca
Chinese_China	zh_CN	Chinese zh
Chinese_HongKong	zh_HK	
Chinese_Macau	zh_MO	

SAS 名	Posix ロケール	エイリアス
Chinese_Singapore	zh_SG	
Chinese_Taiwan	zh_TW	
Cornish_UnitedKingdom	kw_GB	Cornish kw
Croatian_BosniaHerzegovina	hr_BA	
Croatian_Croatia	hr_HR	Croatian hr
Czech_CzechRepublic	cs_CZ	Czech cs
Danish_Denmark	da_DK	Danish da
Dutch_Belgium	nl_BE	
Dutch_Netherlands	nl_NL	Dutch nl
English_Australia	en_AU	
English_Belgium	en_BE	
English_Botswana	en_BW	
English_Canada	en_CA	
English_Caribbean	en_CB	
English_HongKong	en_HK	
English_India	en_IN	
English_Ireland	en_IE	
English_Jamaica	en_JM	
English_Malta	en_MT	
English_NewZealand	en_NZ	
English_Philippines	en_PH	

SAS名	Posix ロケール	エイリアス
English_Singapore	en_SG	English ja
English_SouthAfrica	en_ZA	
English_UnitedKingdom	en_GB	
English_UnitedStates	en_US	English ja
English_Zimbabwe	en_ZW	
Estonian_Estonia	et_EE	Estonian et
Faroese_FaroeIslands	fo_FO	Faroese fo
Finnish_Finland	fi_FI	Finnish fi
French_Belgium	fr_BE	
French_Canada	fr_CA	
French_France	fr_FR	French fr
French_Luxembourg	fr_LU	
French_Switzerland	fr_CH	
German_Austria	de_AT	
German_Germany	de_DE	German de
German_Liechtenstein	de_LI	
German_Luxembourg	de_LU	
German_Switzerland	de_CH	
Greek_Cyprus	el_CY	
Greek_Greece	el_GR	Greek el

SAS 名	Posix ロケール	エイリアス
Greenlandic_Greenland	kl_GL	Greenlandic kl
Hebrew_Israel	he_IL	Hebrew he
Hindi_India	hi_IN	Hindi hi
Hungarian_Hungary	hu_HU	Hungarian hu
Icelandic_Iceland	is_IS	Icelandic is
Indonesian_Indonesia	id_ID	Indonesian id
Irish_Ireland	ga_IE	Irish ga
Italian_Italy	it_IT	Italian it
Italian_Switzerland	it_CH	
Japanese_Japan	ja_JP	Japanese ja
Kazakh_Kazakhstan	kk_KZ	Kazakh kk
Korean_Korea	ko_KR	Korean ko Korean_SouthKorea
Latvian_Latvia	lv_LV	Latvian lv
Lithuanian_Lithuania	lt_LT	Lithuanian lt
Macedonian_Macedonia	mk_MK	Macedonian mk

SAS名	Posix ロケール	エイリアス
Malay_Malaysia	ms_MY	Malay ms
Maltese_Malta	mt_MT	Maltese mt
Marathi_India	mr_IN	Marathi mr
NorwegianBokmal_Norway	nb_NO	NorwegianBokmal nb
NorwegianNynorsk_Norway	nn_NO	NorwegianNynorsk nn
Norwegian_Norway	no_NO	Norwegian no
Persian_India	fa_IN	
Persian_Iran	fa_IR	Persian fa
Polish_Poland	pl_PL	Polish pl
Portuguese_Brazil	pt_BR	
Portuguese_Portugal	pt_PT	Portuguese pt
Romanian_Romania	ro_RO	Romanian ro
Russian_Russia	ru_RU	Russian ru
Russian_Ukraine	ru_UA	
Serbian_BosniaHerzegovina	sr_BA	
Serbian_Montenegro	sr_ME	
Serbian_Serbia	sr_RS	Serbian sr
SerbianLatin_BosniaHerzegovina	sh_BA	

SAS 名	Posix ロケール	エイリアス
SerbianLatin_Montenegro	sh_ME	
SerbianLatin_Serbia	sh_RS	SerbianLatin sh
Slovak_Slovakia	sk_SK	Slovak Slovakian Slovakian_Slovakia sk
Slovenian_Slovenia	sl_SI	Slovenian sl
Spanish_Argentina	es_AR	
Spanish_Bolivia	es_BO	
Spanish_Chile	es_CL	
Spanish_Colombia	es_CO	
Spanish_CostaRica	es_CR	
Spanish_DominicanRepublic	es_DO	
Spanish_Ecuador	es_EC	
Spanish_ElSalvador	es_SV	
Spanish_Guatemala	es_GT	
Spanish_Honduras	es_HN	
Spanish_Mexico	es_MX	
Spanish_Nicaragua	es_NI	
Spanish_Panama	es_PA	
Spanish_Paraguay	es_PY	
Spanish_Peru	es_PE	
Spanish_PuertoRico	es_PR	
Spanish_Spain	es_ES	Spanish es

SAS 名	Posix ロケール	エイリアス
Spanish_UnitedStates	es_US	
Spanish_Uruguay	es_UY	
Spanish_Venezuela	es_VE	
Swedish_Sweden	sv_SE	Swedish sv
Tamil_India	ta_IN	Tamil ta
Telugu_India	te_IN	Telugu te
Thai_Thailand	th_TH	Thai th
Turkish_Turkey	tr_TR	Turkish tr
Ukrainian_Ukraine	uk_UA	Ukrainian uk
Vietnamese_Vietnam	vi_VN	Vietnamese vi

次の表に、有効な Posix 値を示します。DFLANG、DATESTYLE、PAPERSIZE システムオプションは、自動的に設定されます。

次に、例を示します。

```
sas -locale arabic_algeria
```

Arabic_Algeria の LOCALE=値を指定した場合の、対応するシステムオプションのデフォルト設定を次に示します。

```
DFLANG=English
DATESTYLE=DMY
PAPERSIZE=A4
```

表 17.2 LOCALE=システムオプションに基づいた DFLANG、DATESTYLE、PAPERSIZE システムオプションのデフォルト値

Posix ロケール	DFLANG=	DATESTYLE=	PAPERSIZE=
af_ZA	English	YMD	A4
ar_AE	English	DMY	A4

Posix ロケール	DFLANG=	DATESTYLE=	PAPERSIZE=
ar_BH	English	DMY	A4
ar_DZ	English	DMY	A4
ar_EG	English	DMY	A4
ar_IN	English	DMY	A4
ar_IQ	English	DMY	A4
ar_JO	English	DMY	A4
ar_KW	English	DMY	A4
ar_LB	English	DMY	A4
ar_LY	English	DMY	A4
ar_MA	English	DMY	A4
ar_OM	English	DMY	A4
ar_QA	English	DMY	A4
ar_SA	English	DMY	A4
ar_SD	English	DMY	A4
ar_SY	English	DMY	A4
ar_TN	English	DMY	A4
ar_YE	English	DMY	A4
be_BY	English	DMY	A4
bg_BG	English	YMD	A4
bn_IN	English	DMY	A4
bs_BA	English	YMD	A4
ca_ES	English	DMY	A4
cs_CZ	Czech	DMY	A4
da_DK	Danish	DMY	A4
de_AT	German	DMY	A4

Posix ロケール	DFLANG=	DATESTYLE=	PAPERSIZE=
de_CH	Swiss_ German	DMY	A4
de_DE	German	DMY	A4
de_LI	German	DMY	A4
de_LU	German	DMY	A4
el_CY	English	DMY	A4
el_GR	English	DMY	A4
en_AU	English	DMY	A4
en_BE	English	DMY	A4
en_BW	English	DMY	A4
en_CA	English	DMY	letter
en_CB	English	MDY	A4
en_GB	English	DMY	A4
en_HK	English	DMY	A4
en_IE	English	DMY	A4
en_IN	English	DMY	A4
en_JM	English	DMY	A4
en_MT	English	DMY	A4
en_NZ	English	DMY	A4
en_PH	English	MDY	letter
en_SG	English	DMY	A4
en_US	English	MDY	letter
en_ZA	English	DMY	A4
en_ZW	English	DMY	A4
es_AR	Spanish	DMY	A4
es_BO	Spanish	DMY	A4

Posix ロケール	DFLANG=	DATESTYLE=	PAPERSIZE=
es_CL	Spanish	DMY	letter
es_CO	Spanish	DMY	letter
es_CR	Spanish	DMY	letter
es_DO	Spanish	DMY	A4
es_EC	Spanish	DMY	A4
es_ES	Spanish	DMY	A4
es_GT	Spanish	DMY	letter
es_HN	Spanish	MDY	A4
es_MX	Spanish	DMY	letter
es_NI	Spanish	MDY	letter
es_PA	Spanish	MDY	letter
es_PE	Spanish	DMY	A4
es_PR	Spanish	MDY	letter
es_PY	Spanish	DMY	A4
es_SV	Spanish	MDY	letter
es_US	Spanish	DMY	letter
es_UY	Spanish	DMY	A4
es_VE	Spanish	DMY	letter
et_EE	English	DMY	A4
eu_ES	English	DMY	A4
fa_IN	English	YMD	A4
fa_IR	English	YMD	A4
fi_FI	Finnish	DMY	A4
fo_FO	English	DMY	A4
fr_BE	French	DMY	A4

Posix ロケール	DFLANG=	DATESTYLE=	PAPERSIZE=
fr_CA	French	DMY	letter
fr_CH	Swiss_French	DMY	A4
fr_FR	French	DMY	A4
fr-LU	French	DMY	A4
ga_IE	English	DMY	A4
gv_GB	English	DMY	A4
he_IL	English	DMY	A4
hi_IN	English	DMY	A4
hr_BA	Croatian	YMD	A4
hr_HR	Croatian	YMD	A4
hu_HU	Hungarian	YMD	A4
id_ID	English	DMY	A4
is_IS	English	DMY	A4
it_CH	Italian	DMY	A4
it_IT	Italian	DMY	A4
ja_JP	Japanese	YMD	A4
kk_KZ	English	DMY	A4
kl_GL	English	DMY	A4
ko_KR	Locale	YMD	A4
kw_GB	English	DMY	A4
lt_LT	English	YMD	A4
lv_LV	English	YMD	A4
mk_MK	English	DMY	A4
mr_IN	English	DMY	A4
ms_MY	English	DMY	A4

Posix ロケール	DFLANG=	DATESTYLE=	PAPERSIZE=
mt_MT	English	DMY	A4
nb_NO	Norwegian	DMY	A4
nl_BE	Dutch	DMY	A4
nl_NL	Dutch	DMY	A4
nn_NO	Norwegian	DMY	A4
no_NO	Norwegian	DMY	A4
pl_PL	Polish	YMD	A4
pt_BR	Portuguese	DMY	A4
pt_PT	Portuguese	DMY	A4
ro_RO	English	DMY	A4
ru_RU	Russian	DMY	A4
ru_UA	Russian	DMY	A4
sh_BA	English	DMY	A4
sh_ME	English	DMY	A4
sh_RS	English	DMY	A4
sk_SK	English	DMY	A4
sl_SI	Slovenian	YMD	A4
sq_AL	English	YMD	A4
sr_BA	English	DMY	A4
sr_ME	English	DMY	A4
sr_RS	English	DMY	A4
sv_SE	Swedish	YMD	A4
ta_IN	English	DMY	A4
te_IN	English	DMY	A4
th_TH	English	DMY	A4

Posix ロケール	DFLANG=	DATESTYLE=	PAPERSIZE=
tr_TR	English	DMY	A4
uk_UA	English	DMY	A4
vi_VN	English	DMY	A4
zh_CN	Locale	YMD	A4
zh_HK	Locale	YMD	A4
zh_MO	Locale	YMD	A4
zh_SG	Locale	DMY	A4
zh_TW	Locale	YMD	A4

18 章

SAS 言語要素中のエンコーディング値

エンコーディング値を使用する SAS 言語要素の概要	613
データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値	613

エンコーディング値を使用する SAS 言語要素の概要

SAS セッションのエンコーディングが SAS ファイルのエンコーディングまたは SAS ファイル内に存在するデータと異なる場合、トランスコーディングが必要になります。SAS ファイルが Western Latin1 エンコーディングで作成され、UTF-8 エンコーディングを使用している SAS Viya に接続されている SAS Studio に読み込まれたとします。SAS Viya がそのファイルに正常にアクセスできるように、SAS データファイルを Western Latin1 エンコーディングから UTF-8 エンコーディングにトランスコードする必要があります。トランスコーディングオプションを含む SAS 言語要素などの、トランスコーディングの概念の詳細については、[4 章, “各国語サポート関連のトランスコーディング” \(19 ページ\)](#) を参照してください。

データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値

次の表は、各動作環境でデータのトランスコーディングに使用される SBCS、DBCS および Unicode エンコーディング値のリストです。エンコーディング値は、トランスコーディングのオプションを含む SAS 言語要素で有効です。

注: エンコーディング値にハイフン(-)が含まれる場合は、エンコーディング値を引用符で囲んでください。

表 18.1 データのトランスコーディングに使用する SBCS、DBCS および Unicode エンコーディング値

エンコーディング名	短縮名	説明	1 文字の最大バイト
aarabic	aara	アラビア語 (Macintosh)	1
acroatian	acro	MacOS 36 Croatian	1
acyrillic	acyr	MacOS 7 Cyrillic	1
agreek	agrk	ギリシャ語 (Macintosh)	1
ahebrew	aheb	ヘブライ語 (Macintosh)	1
aiceland	aice	アイスランド語 (Macintosh)	1
any	anye	トランスコーディ ング指定無し	1
arabic	arab	アラビア語(ISO)	1
aroman	arom	Roman (Macintosh)	1
aromania	arma	MacOS 38 Romania	1
athai	atha	MacOS 21-Thai	1
aturkish	atur	トルコ語 (Macintosh)	1
aukrainian	aukr	ウクライナ語 (Macintosh)	1
big5	big5	繁体字中国語(Big5)	2
cyrillic	cyrI	キリル語(ISO)	1
dec-cn	zvms	簡体字中国語(DEC)	4
dec-jp	jvms	日本語(DEC)	2
dec-tw	yvms	繁体字中国語(DEC)	4
ebcdic037	e037	北米諸語(EBCDIC)	1
ebcdic275	e275	ブラジル(EBCDIC)	1

エンコーディング名	短縮名	説明	1文字の最大バイト
ebcdic424	e424	ヘブライ語 (EBCDIC)	1
ebcdic425	e425	アラビア語 (EBCDIC)	1
ebcdic500	e500	国際化対応 (EBCDIC)	1
ebcdic838	e838	タイ語(EBCDIC)	1
ebcdic870	e870	中央ヨーロッパ諸 語(EBCDIC)	1
ebcdic875	e875	ギリシャ語 (EBCDIC)	1
ebcdic905	e905	ラテン諸語 3 EBCDIC	1
ebcdic924	e924	ヨーロッパ諸語 (EBCDIC)	1
ebcdic1025	ecyr	キリル語(EBCDIC)	1
ebcdic1026	etur	トルコ語(EBCDIC)	1
ebcdic1047	elat	西欧(EBCDIC)	1
ebcdic1097	e097	ペルシア諸語 (EBCDIC)	1
ebcdic1112	ebal	バルト語(EBCDIC)	1
ebcdic1122	eest	エストニア語 (EBCDIC)	1
ebcdic1130	evie	ベトナム語 (EBCDIC)	1
ebcdic1137	e137	デーバナーガリー 文字(EBCDIC)	1
ebcdic1140	e140	北米諸語(EBCDIC)	1
ebcdic1141	e141	オーストリア/ドイ ツ(EBCDIC)	1
ebcdic1142	e142	デンマーク/ノルウ ェー(EBCDIC)	1

エンコーディング名	短縮名	説明	1文字の最大バイト
ebcdic1143	e143	フィンランド/スウェーデン(EBCDIC)	1
ebcdic1144	e144	イタリア(EBCDIC)	1
ebcdic1145	e145	スペイン(EBCDIC)	1
ebcdic1146	e146	英国(EBCDIC)	1
ebcdic1147	e147	フランス(EBCDIC)	1
ebcdic1148	e148	国際化対応(EBCDIC)	1
ebcdic1149	e149	アイスランド語(EBCDIC)	1
ebcdic1153	e153	Latin 2 Euro EBCDIC	1
ebcdic1154	e154	Cyrillic Euro EBCDIC	1
ebcdic1155	e155	Turkey Euro EBCDIC	1
ebcdic1156	e156	Baltic Euro EBCDIC	1
ebcdic1157	e157	Estonia Euro EBCDIC	1
ebcdic1158	e158	Cryillic Ukraine Euro EBCDIC	1
ebcdic1160	e160	cp1160 EBCDIC	1
ebcdic1164	e164	cp1164 EBCDIC	1
ebcdicany	eany	すべての EBCDIC エンコーディングと互換性のあるデータセットを作成可能	1
euc-cn	zeuc	簡体字中国語(EUC)	2
euc-jp	jeuc	日本語(EUC)	4
euc-kr	keuc	韓国語(EUC)	4
euc-tw	yeuc	繁体字中国語(EUC)	4

エンコーディング名	短縮名	説明	1文字の最大バイト
fujitsu-cn	zfuj	簡体字中国語 (FACOM)	4
fujitsu-jp	jfug	日本語(FACOM)	4
fujitsu-ko	kfuj	韓国語(FACOM)	4
fujitsu-tw	yfuj	繁体字中国語 (FACOM)	4
gb18030	gbke	簡体字中国語 (GB18030)	4
greek	grek	ギリシャ語(ISO)	1
hebrew	hebr	ヘブライ語(ISO)	1
hitachi-cn	zhit	簡体字中国語 (HITAC)	6
hitachi-jp	jhit	日本語(HITAC)	6
hitachi-ko	khit	韓国語(HITAC)	6
hitachi-tw	yhit	繁体字中国語 (HITAC)	4
hitsas-jp	jhts	日本語(XHITAC)	4
hitsas-ko	khts	韓国語(XHITAC)	4
hitsas-tw	yhts	繁体字中国語 (XHITAC)	4
hp15-tw	yhpx	繁体字中国語 (HP15)	2
ibm-1381	zpce	簡体字中国語 (PCIBM)	2
ibm-930	j930	Japanese katakana	4
ibm-933	kibm	韓国語(IBM)	4
ibm-935	zibm	簡体字中国語(IBM)	4
ibm-937	yibm	繁体字中国語(IBM)	4
ibm-939	jibm	日本語(IBM)	4
ibm-942	j942	日本語(PCIBM)	2

エンコーディング名	短縮名	説明	1文字の最大バイト
ibm-949	kpce	韓国語(PCIBM)	2
iso2022cncns	zist	繁体字中国語 (ISO-2022)	4
iso2022cn gb	ziso	簡体字中国語 (ISO-2022)	4
iso2022jp	jiso	日本語(ISO-2022)	8
iso2022kr	kiso	韓国語(ISO-2022)	4
kz1048	kz48	Kazakh cp1048 Cyrillic	1
latin1	lat1	西ヨーロッパ諸語 (ISO)	1
latin2	lat2	中央ヨーロッパ諸 語(ISO)	1
latin3	lat3	ラテン諸語 3(ISO 8859/3)	1
latin4	lat4	ラテン諸語 4(ISO 8859/4)	1
latin5	lat5	トルコ語(ISO)	1
latin6	lat6	バルト語(ISO)	1
latin7	lat7	バルト語	1
latin8	lat8	ラテン諸語 8(ISO 8859/14)	1
latin9	lat9	ヨーロッパ諸語 (ISO)	1
latin10	lat10	南東ヨーロッパ言 語	1
macos-1	jmac	日本語(PCMAC)	2
macos-2	ymac	繁体字中国語 (PCMAC)	2
macos-3	kmac	韓国語(PCMAC)	2
macos-25	zmac	簡体字中国語 (PCMAC)	2

エンコーディング名	短縮名	説明	1文字の最大バイト
ms-932	j932	日本語(PCMS)	2
ms-936	zwin	簡体字中国語 (PCMS)	2
ms-949	kwin	韓国語(PCMS)	2
ms-950	ywin	繁体字中国語 (PCMS)	2
msdos720	p720	アラビア語(MS- DOS)	1
msdos737	p737	ギリシャ語(MS- DOS)	1
msdos775	p775	バルト語(MS-DOS)	1
open_ed-037	eous	米国(Open Edition)	1
open_ed-275	eobr	ブラジル (OpenEdition)	1
open_ed-424	eoiv	ヘブライ語 (OpenEdition)	1
open_ed-425	ea2	アラビア語 (OpenEdition)	1
open_ed-838	eoht	タイ語 (OpenEdition)	1
open_ed-870	eol2	中央ヨーロッパ諸 語(OpenEdition)	1
open_ed-875	eoel	ギリシャ語 (OpenEdition)	1
open_ed-905	eol3	ラテン諸語 3(Open Edition EBCDIC)	1
open_ed-924	eolt	ヨーロッパ諸語 (OpenEdition)	1
open_ed-930	oe30	カタカナ (OpenEdition)	4
open_ed-933	oe33	韓国語 (OpenEdition)	4
open_ed-935	oe35	簡体字中国語 (OpenEdition)	4

エンコーディング名	短縮名	説明	1文字の最大バイト
open_ed-937	oe37	繁体字中国語 (OpenEdition)	4
open_ed-939	oe39	日本語(IBM)	4
open_ed-939E	oe9e	日本語(IBM 拡張)	4
open_ed-1025	eo3y	キリル語 (OpenEdition)	1
open_ed-1026	eo3r	トルコ語 (OpenEdition)	1
open_ed-1047	eo11	西ヨーロッパ諸語 (OpenEdition)	1
open_ed-1097	eo3a	ペルシア諸語 (OpenEdition EBCDIC)	1
open_ed-1112	eo3l	バルト語 (OpenEdition)	1
open_ed-1122	eo3t	エストニア語 (OpenEdition)	1
open_ed-1130	eo3i	ベトナム語 (OpenEdition)	1
open_ed-1140	eo40	北米諸語 (OpenEdition)	1
open_ed-1141	eo41	オーストリア/ドイツ (OpenEdition)	1
open_ed-1142	eo42	デンマーク/ノルウェー (OpenEdition)	1
open_ed-1143	eo43	フィンランド/スウェーデン (OpenEdition)	1
open_ed-1144	eo44	イタリア (OpenEdition)	1
open_ed-1145	eo45	スペイン (OpenEdition)	1
open_ed-1146	eo46	英国(OpenEdition)	1
open_ed-1147	eo47	フランス (OpenEdition)	1

エンコーディング名	短縮名	説明	1文字の最大バイト
open_ed-1148	eo48	国際化対応 (OpenEdition)	1
open_ed-1149	eoIS	アイスランド (OpenEdition EBCDIC)	1
open_ed-1153	eo53	Latin 2 Euro OpenEdition EBCDIC	1
open_ed-1154	eo54	Cyrillic Euro OpenEdition EBCDIC	1
open_ed-1155	eo55	Turkey Euro OpenEdition EBCDIC	1
open_ed-1156	eo56	Baltic Euro OpenEdition EBCDIC	1
open_ed-1157	eo57	Estonia Euro OpenEdition EBCDIC	1
open_ed-1158	eo58	Cryrillic Ukraine Euro 1158 OpenEdition EBCDIC	1
open_ed-1160	eo60	1160 OpenEdition EBCDIC	1
open_ed-1164	eo64	1164 OpenEdition EBCDIC	1
open_ed-1166	eo66	1166 OpenEdition kz-cyr	1
pciscii806	p806	インド(PC)	1
pcoem437	p437	米国(IBM-PC)	1
pcoem850	p850	西ヨーロッパ諸語 (IBM-PC)	1
pcoem852	p852	中央ヨーロッパ諸 語(IBM-PC)	1
pcoem857	p857	トルコ語(IBM-PC)	1

エンコーディング名	短縮名	説明	1文字の最大バイト
pcoem858	p858	ヨーロッパ諸語 (IBM-PC)	1
pcoem860	p860	ポルトガル語(MS- DOS)	1
pcoem862	p862	ヘブライ語(IBM- PC)	1
pcoem863	p863	フランス語(カナダ) (IBM-PC)	1
pcoem864	p864	アラビア語(IBM- PC)	1
pcoem865	p865	北欧諸語(IBM-PC)	1
pcoem866	p866	キリル語(IBM-PC)	1
pcoem869	p869	ギリシャ語(IBM- PC)	1
pcoem874	p874	タイ語(IBM-PC)	1
pcoem921	p921	バルト語(IBM-PC)	1
pcoem922	p922	エストニア語(IBM- PC)	1
pcoem1129	pvie	ベトナム語(IBM- PC)	1
pc1098	po98	ペルシア諸語(PC)	1
roman8	rom8	HP Roman 8	1
shift-jis	SJIS	日本語(SJIS)	2
shift-jis04	sjs4	Shift_JIS-2004 サロ ゲートペアサポー ト	2
thai	thai	タイ語(ISO)	1
us-ascii	ansi	すべての ASCII エ ンコーディングと 互換性のあるデー タセットを作成可 能	1
utf-8	utf8	Unicode (UTF-8)	4

エンコーディング名	短縮名	説明	1文字の最大バイト
utf-16be	u16b	Unicode (UTF-16BE)	2*
utf-16le	u16l	Unicode (UTF-16LE)	2*
utf-32be	u32b	Unicode (UTF-32BE)	4**
utf-32le	u32l	Unicode (UTF-32LE)	4**
warabic	wara	アラビア語 (Windows)	1
wbaltic	wbal	バルト語 (Windows)	1
wcyrillic	wcyr	キリル語 (Windows)	1
wgreek	wgrk	ギリシャ語 (Windows)	1
whebrew	wheb	ヘブライ語 (Windows)	1
wlatin1	wlt1	西ヨーロッパ諸語 (Windows)	1
wlatin2	wlt2	中央ヨーロッパ諸語 (Windows)	1
wturkish	wtur	トルコ語 (Windows)	1
wvietnamese	wvie	ベトナム語 (Windows)	1

* UTF-16BE と UTF-16LE は各文字 2 バイトの固定長になります。

** UTF-32BE と UTF-32LE は各文字 4 バイトの固定長になります。

11 部

付録

付録 1	
タイムゾーン ID とタイムゾーン名.....	627

付録 1

タイムゾーン ID とタイムゾーン名

エリア: Africa (アフリカ)	627
エリア: America (アメリカ-北、中央、および南)	629
エリア: Antarctica (南極)	639
エリア: Asia (アジア)	639
エリア: Atlantic (大西洋)	643
エリア: Australia (オーストラリア)	644
エリア: その他	646
エリア: Europe (ヨーロッパ)	651
エリア: Pacific (太平洋)	656

エリア: Africa (アフリカ)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Africa/Abidjan	GMT	CI	グリニッジ標準時間	00:00
Africa/Accra	GMT	GH	グリニッジ標準時間	00:00
Africa/Bamako	GMT	ML	グリニッジ標準時間	00:00
Africa/Banjul	GMT	GM	グリニッジ標準時間	00:00
Africa/Bissau	GMT	GW	グリニッジ標準時間	00:00
Africa/Conakry	GMT	GN	グリニッジ標準時間	00:00
Africa/Dakar	GMT	SN	グリニッジ標準時間	00:00
Africa/Freetown	GMT	SL	グリニッジ標準時間	00:00
Africa/Lome	GMT	TG	グリニッジ標準時間	00:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Africa/Monrovia	GMT	LR	グリニッジ標準時間	00:00
Africa/Nouakchott	GMT	MR	グリニッジ標準時間	00:00
Africa/Ouagadougou	GMT	BF	グリニッジ標準時間	00:00
Africa/Sao_Tome	GMT	ST	グリニッジ標準時間	00:00
Africa/Timbuktu	GMT	ML	グリニッジ標準時間	0.00
Africa/Casablanca	WET WEST	MA MA	西ヨーロッパ時間 西ヨーロッパ夏時間	00:00 01.00
Africa/El_Aaiun	WET WEST	EH EH	西ヨーロッパ時間 西ヨーロッパ夏時間	00:00 01.00
Africa/Algiers	CET	DZ	中央ヨーロッパ時間	1:00
Africa/Ceuta	CET CEST	ES	中央ヨーロッパ時間 中央ヨーロッパ夏時間	1:00 2:00
Africa/Tunis	CET	TN	中央ヨーロッパ時間	1:00
Africa/Bangui	WAT	CF	西アフリカ時間	1:00
Africa/Brazzaville	WAT	CG	西アフリカ時間	1:00
Africa/Douala	WAT	CM	西アフリカ時間	1:00
Africa/Kinshasa	WAT	CD	西アフリカ時間	1:00
Africa/Lagos	WAT	NG	西アフリカ時間	1:00
Africa/Libreville	WAT	GA	西アフリカ時間	1:00
Africa/Luanda	WAT	AO	西アフリカ時間	1:00
Africa/Malabo	WAT	GQ	西アフリカ時間	1:00
Africa/Ndjamena	WAT	TD	西アフリカ時間	1:00
Africa/Niamey	WAT	NE	西アフリカ時間	1:00
Africa/Porto-Novo	WAT	BJ	西アフリカ時間	1:00
Africa/Windhoek	WAT WAST	NA	西アフリカ時間 西アフリカ夏時間	1:00 2:00
Africa/Blantyre	CAT	MW	中央アフリカ時間	2:00
Africa/Bujumbura	CAT	BI	中央アフリカ時間	2:00
Africa/Gaborone	CAT	BW	中央アフリカ時間	2:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Africa/Harare	CAT	ZW	中央アフリカ時間	2:00
Africa/Kigali	CAT	RW	中央アフリカ時間	2:00
Africa/Lubumbashi	CAT	CD	中央アフリカ時間	2:00
Africa/Lusaka	CAT	ZM	中央アフリカ時間	2:00
Africa/Maputo	CAT	MZ	中央アフリカ時間	2:00
Africa/Cairo	EET	EG	東ヨーロッパ時間	2:00
	EET	EG	東ヨーロッパ夏時間	2:00
Africa/Tripoli	EET	LY	東ヨーロッパ時間	2:00
Africa/Johannesburg	SAST	ZA	南アフリカ標準時間	2:00
Africa/Maseru	SAST	LS	南アフリカ標準時間	2:00
Africa/Mbabane	SAST	SZ	南アフリカ標準時間	2:00
Africa/Addis_Ababa	EAT	ET	東アフリカ時間	3:00
Africa/Asmara	EAT	ER	東アフリカ時間	03.00
Africa/Asmera	EAT	ER	東アフリカ時間	3:00
Africa/Dar_es_Salaam	EAT	TZ	東アフリカ時間	3:00
Africa/Djibouti	EAT	DJ	東アフリカ時間	3:00
Africa/Juba	EAT	SS	東アフリカ時間	3:00
Africa/Kampala	EAT	UG	東アフリカ時間	3:00
Africa/Khartoum	EAT	SD	東アフリカ時間	3:00
Africa/Mogadishu	EAT	SO	東アフリカ時間	3:00
Africa/Nairobi	EAT	KE	東アフリカ時間	3:00

エリア: America (アメリカ-北、中央、および南)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Adak	HAST	US	ハワイ・アリューシャン標準時間	-10:00
America/Adak	HADT	US	ハワイ・アリューシャン夏時間	-9:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Anchorage	AKST	US	アラスカ標準時間	-9:00
America/Anchorage	AKDT	US	アラスカ夏時間	-8:00
America/Anguilla	AST	AI	アトランティック標準時間	-4:00
America/Antigua	AST	AG	アトランティック標準時間	-4:00
America/Araguaina	BRT	BR	ブラジル標準時間	-3:00
America/Argentina/ Buenos_Aires	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/ Catamarca	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/ ComodRivadavia	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/Cordoba	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/Jujuy	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/La_Rioja	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/Mendoza	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/ Rio_Gallegos	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/Salta	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/San_Juan	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/San_Luis	ART	AR	西アルゼンチン標準時間	-3:00
America/Argentina/Tucuman	ART	AR	アルゼンチン標準時間	-3:00
America/Argentina/Ushuaia	ART	AR	アルゼンチン標準時間	-3:00
America/Aruba	AST	AW	アトランティック標準時間	-4:00
America/Asuncion	PYT	PY	パラグアイ標準時間	-4:00
America/Asuncion	PYST	PY	パラグアイ夏時間	-3:00
America/Atikokan	EST	CA	東部標準時間	-5:00
America/Atka	HAST	US	ハワイ・アリューシャン標準時間	-10:00
America/Atka	HADT	US	ハワイ・アリューシャン夏時間	-9:00
America/Bahia	BRT	BR	ブラジル標準時間	-3:00
America/Bahia_Banderas	CST	MX	中部標準時間	-6:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Bahia_Banderas	CDT	MX	中部夏時間	-5:00
America/Barbados	AST	BB	アトランティック標準時間	-4:00
America/Belem	BRT	BR	ブラジリア標準時間	-3:00
America/Belize	CST	BZ	中部標準時間	-6:00
America/Blanc-Sablon	AST	CA	アトランティック標準時間	-4:00
America/Boa_Vista	AMT	BR	アマゾン標準時間	-4:00
America/Bogota	COT	CO	コロンビア標準時間	-5:00
America/Boise	MST	US	山岳部標準時間	-7:00
America/Boise	MDT	US	山岳部夏時間	-6:00
America/Buenos_Aires	ART	AR	アルゼンチン標準時間	-3:00
America/Cambridge_Bay	MST	CA	山岳部標準時間	-7:00
America/Cambridge_Bay	MDT	CA	山岳部夏時間	-6:00
America/Campo_Grande	AMT	BR	アマゾン標準時間	-4:00
America/Campo_Grande	AMST	BR	アマゾン夏時間	-3:00
America/Cancun	CST	MX	中部標準時間	-6:00
America/Cancun	CDT	MX	中部夏時間	-5:00
America/Caracas	VET	VE	ベネズエラ時間	-1:27
America/Catamarca	ART	AR	アルゼンチン標準時間	-3:00
America/Cayenne	GFT	GF	フランス領ギアナ時間	-3:00
America/Cayman	EST	KY	東部標準時間	-5:00
America/Chicago	CST	US	中部標準時間	-6:00
America/Chicago	CDT	US	中部夏時間	-5:00
America/Chihuahua	MST	MX	山岳部標準時間	-7:00
America/Chihuahua	MDT	MX	山岳部夏時間	-6:00
America/Coral_Harbour	EST	CA	東部標準時間	-5:00
America/Cordoba	ART	AR	アルゼンチン標準時間	-3:00
America/Costa_Rica	CST	CR	中部標準時間	-6:00
America/Creston	MST	CA	山岳部標準時間	-7:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Cuiaba	AMT	BR	アマゾン標準時間	-4:00
America/Cuiaba	AMST	BR	アマゾン夏時間	-3:00
America/Curacao	AST	CW	アトランティック標準時間	-4:00
America/Danmarkshavn	GMT	GL	グリニッジ標準時間	00:00
America/Dawson	PST	CA	太平洋標準時間	-8:00
America/Dawson	PDT	CA	太平洋夏時間	-7:00
America/Dawson_Creek	MST	CA	山岳部標準時間	-7:00
America/Denver	MST	US	山岳部標準時間	-7:00
America/Denver	MDT	US	山岳部夏時間	-6:00
America/Detroit	EST	US	東部標準時間	-5:00
America/Detroit	EDT	US	東部夏時間	-4:00
America/Dominica	AST	DM	アトランティック標準時間	-4:00
America/Edmonton	MST	CA	山岳部標準時間	-7:00
America/Edmonton	MDT	CA	山岳部夏時間	-6:00
America/Eirunepe	ACT	BR	アマゾン標準時間	-5:00
America/El_Salvador	CST	SV	中部標準時間	-6:00
America/Ensenada	PST	MX	太平洋標準時間	-8:00
America/Ensenada	PDT	MX	太平洋夏時間	-7:00
America/Fort_Wayne	EST	US	東部標準時間	-5:00
America/Fort_Wayne	EDT	US	東部夏時間	-4:00
America/Fortaleza	BRT	BR	ブラジル標準時間	-3:00
America/Glace_Bay	AST	CA	アトランティック標準時間	-4:00
America/Glace_Bay	ADT	CA	アトランティック夏時間	-3:00
America/Godthab	WGT	GL	西グリーンランド標準時間	-3:00
America/Godthab	WGST	GL	西グリーンランド夏時間	-2:00
America/Goose_Bay	AST	CA	アトランティック標準時間	-4:00
America/Goose_Bay	ADT	CA	アトランティック夏時間	-3:00
America/Grand_Turk	EST	TC	東部標準時間	-5:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Grand_Turk	EDT	TC	東部夏時間	-4:00
America/Grenada	AST	GD	アトランティック標準時間	-4:00
America/Guadeloupe	AST	GP	アトランティック標準時間	-4:00
America/Guatemala	CST	GT	中部標準時間	-6:00
America/Guayaquil	ECT	EC	エクアドル時間	-5:00
America/Guyana	GYT	GY	ガイアナ時間	-4:00
America/Halifax	AST	CA	アトランティック標準時間	-4:00
America/Halifax	ADT	CA	アトランティック夏時間	-3:00
America/Havana	CST	CU	キューバ標準時間	-5:00
America/Havana	CDT	CU	キューバ夏時間	-4:00
America/Hermosillo	MST	MX	山岳部標準時間	-7:00
America/Indiana/ Indianapolis	EST	US	東部標準時間	-5:00
America/Indiana/ Indianapolis	EDT	US	東部夏時間	-4:00
America/Indiana/Knox	CST	US	中部標準時間	-6:00
America/Indiana/Knox	CDT	US	中部夏時間	-5:00
America/Indiana/Marengo	EST	US	東部標準時間	-5:00
America/Indiana/Marengo	EDT	US	東部夏時間	-4:00
America/Indiana/Petersburg	EST	US	東部標準時間	-5:00
America/Indiana/Petersburg	EDT	US	東部夏時間	-4:00
America/Indiana/Tell_City	CST	US	中部標準時間	-6:00
America/Indiana/Tell_City	CDT	US	中部夏時間	-5:00
America/Indiana/Vevay	EST	US	東部標準時間	-5:00
America/Indiana/Vevay	EDT	US	東部夏時間	-4:00
America/Indiana/Vincennes	EST	US	東部標準時間	-5:00
America/Indiana/Vincennes	EDT	US	東部夏時間	-4:00
America/Indiana/Winamac	EST	US	東部標準時間	-5:00
America/Indiana/Winamac	EDT	US	東部夏時間	-4:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Indianapolis	EST	US	東部標準時間	-5:00
America/Indianapolis	EDT	US	東部夏時間	-4:00
America/Inuvik	MST	CA	山岳部標準時間	-7:00
America/Inuvik	MDT	CA	山岳部夏時間	-6:00
America/Iqaluit	EST	CA	東部標準時間	-5:00
America/Iqaluit	EDT	CA	東部夏時間	-4:00
America/Jamaica	EST	JM	東部標準時間	-5:00
America/Jujuy	ART	AR	アルゼンチン標準時間	-3:00
America/Juneau	AKST	US	アラスカ標準時間	-9:00
America/Juneau	AKDT	US	アラスカ夏時間	-8:00
America/Kentucky/Louisville	EST	US	東部標準時間	-5:00
America/Kentucky/Louisville	EDT	US	東部夏時間	-4:00
America/Kentucky/Monticello	EST	US	東部標準時間	-5:00
America/Kentucky/Monticello	EDT	US	東部夏時間	-4:00
America/Knox_IN	CST	US	中部標準時間	-6:00
America/Knox_IN	CDT	US	中部夏時間	-5:00
America/Kralendijk	AST	BQ	アトランティック標準時間	-4:00
America/La_Paz	BOT	BO	ボリビア時間	-4:00
America/Lima	PET	PE	ペルー標準時間	-5:00
America/Los_Angeles	PST	US	太平洋標準時間	-8:00
America/Los_Angeles	PDT	US	太平洋夏時間	-7:00
America/Louisville	EST	US	東部標準時間	-5:00
America/Louisville	EDT	US	東部夏時間	-4:00
America/Lower_Princes	AST	SX	アトランティック標準時間	-4:00
America/Maceio	BRT	BR	ブラジル標準時間	-3:00
America/Managua	CST	NI	中部標準時間	-6:00
America/Manaus	AMT	BR	アマゾン標準時間	-4:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Marigot	AST	MF	アトランティック標準時間	-4:00
America/Martinique	AST	MQ	アトランティック標準時間	-4:00
America/Matamoros	CST	MX	中部標準時間	-6:00
America/Matamoros	CDT	MX	中部夏時間	-5:00
America/Mazatlan	MST	MX	山岳部標準時間	-7:00
America/Mazatlan	MDT	MX	山岳部夏時間	-6:00
America/Mendoza	ART	AR	アルゼンチン標準時間	-3:00
America/Menominee	CST	US	中部標準時間	-6:00
America/Menominee	CDT	US	中部夏時間	-5:00
America/Merida	CST	MX	中部標準時間	-6:00
America/Merida	CDT	MX	中部夏時間	-5:00
America/Metlakatla	MEST	US	太平洋標準時間	-8:00
America/Mexico_City	CST	MX	中部標準時間	-6:00
America/Mexico_City	CDT	MX	中部夏時間	-5:00
America/Miquelon	PMST	PM	サンピエール・ミクロン標準時間	-3:00
America/Miquelon	PMDT	PM	サンピエール・ミクロン夏時間	-2:00
America/Moncton	AST	CA	アトランティック標準時間	-4:00
America/Moncton	ADT	CA	アトランティック夏時間	-3:00
America/Monterrey	CST	MX	中部標準時間	-6:00
America/Monterrey	CDT	MX	中部夏時間	-5:00
America/Montevideo	UYT	UY	ウルグアイ標準時間	-3:00
America/Montevideo	UYST	UY	ウルグアイ夏時間	-2:00
America/Montserrat	AST	MS	アトランティック標準時間	-4:00
America/Nassau	EST	BS	東部標準時間	-5:00
America/Nassau	EDT	BS	東部夏時間	-4:00
America/New_York	EST	US	東部標準時間	-5:00
America/New_York	EDT	US	東部夏時間	-4:00
America/Nipigon	EST	CA	東部標準時間	-5:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Nipigon	EDT	CA	東部夏時間	-4:00
America/Nome	AKST	US	アラスカ標準時間	-9:00
America/Nome	AKDT	US	アラスカ夏時間	-8:00
America/Noronha	FNT	BR	フェルナンド・デ・ノローニャ標準時間	-2:00
America/North_Dakota/Beulah	CST	US	中部標準時間	-6:00
America/North_Dakota/Beulah	CDT	US	中部夏時間	-5:00
America/North_Dakota/Center	CST	US	中部標準時間	-6:00
America/North_Dakota/Center	CDT	US	中部夏時間	-5:00
America/North_Dakota/New_Salem	CST	US	中部標準時間	-6:00
America/North_Dakota/New_Salem	CDT	US	中部夏時間	-5:00
America/Ojinaga	MST	MX	山岳部標準時間	-7:00
America/Ojinaga	MDT	MX	山岳部夏時間	-6:00
America/Panama	EST	PA	東部標準時間	-5:00
America/Pangnirtung	EST	CA	東部標準時間	-5:00
America/Pangnirtung	EDT	CA	東部夏時間	-4:00
America/Paramaribo	SRT	SR	スリナム時間	-3:00
America/Phoenix	MST	US	山岳部標準時間	-7:00
America/Port-au-Prince	EST	HT	東部標準時間	-5:00
America/Port-au-Prince	EDT	HT	東部夏時間	-4:00
America/Port_of_Spain	AST	TT	アトランティック標準時間	-4:00
America/Porto_Acre	ACT	BR	アマゾン標準時間	-5:00
America/Porto_Velho	AMT	BR	アマゾン標準時間	-4:00
America/Puerto_Rico	AST	PR	アトランティック標準時間	-4:00
America/Rainy_River	CST	CA	中部標準時間	-6:00
America/Rainy_River	CDT	CA	中部夏時間	-5:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/Rankin_Inlet	CST	CA	中部標準時間	-6:00
America/Rankin_Inlet	CDT	CA	中部夏時間	-5:00
America/Recife	BRT	BR	ブラジリア標準時間	-3:00
America/Regina	CST	CA	中部標準時間	-6:00
America/Resolute	CST	CA	中部標準時間	-6:00
America/Resolute	CDT	CA	中部夏時間	-5:00
America/Rio_Branco	ACT	BR	アマゾン標準時間	-5:00
America/Rosario	ART	AR	アルゼンチン標準時間	-3:00
America/Santa_Isabel	PST	MX	太平洋標準時間	-8:00
America/Santa_Isabel	PDT	MX	太平洋夏時間	-7:00
America/Santarem	BRT	BR	ブラジリア標準時間	-3:00
America/Santiago	CLT	CL	チリ標準時間	-4:00
America/Santiago	CLST	CL	チリ夏時間	-3:00
America/Santo_Domingo	AST	DO	アトランティック標準時間	-4:00
America/Sao_Paulo	BRT	BR	ブラジリア標準時間	-3:00
America/Sao_Paulo	BRST	BR	ブラジリア夏時間	-2:00
America/Scoresbysund	EGT	GL	東グリーンランド標準時間	-1:00
America/Scoresbysund	EGST	GL	東グリーンランド夏時間	00:00
America/Shiprock	MST	US	山岳部標準時間	-7:00
America/Shiprock	MDT	US	山岳部夏時間	-6:00
America/Sitka	AKST	US	アラスカ標準時間	-9:00
America/Sitka	AKDT	US	アラスカ夏時間	-8:00
America/St_Barthelemy	AST	BL	アトランティック標準時間	-4:00
America/St_Johns	NST	CA	ニューファンドランド標準時間	-1:27
America/St_Johns	NDT	CA	ニューファンドランド夏時間	-1:27
America/St_Kitts	AST	KN	アトランティック標準時間	-4:00
America/St_Lucia	AST	LC	アトランティック標準時間	-4:00
America/St_Thomas	AST	VI	アトランティック標準時間	-4:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
America/St_Vincent	AST	VC	アトランティック標準時間	-4:00
America/Swift_Current	CST	CA	中部標準時間	-6:00
America/Tegucigalpa	CST	HN	中部標準時間	-6:00
America/Thule	AST	GL	アトランティック標準時間	-4:00
America/Thule	ADT	GL	アトランティック夏時間	-3:00
America/Thunder_Bay	EST	CA	東部標準時間	-5:00
America/Thunder_Bay	EDT	CA	東部夏時間	-4:00
America/Tijuana	PST	MX	太平洋標準時間	-8:00
America/Tijuana	PDT	MX	太平洋夏時間	-7:00
America/Toronto	EST	CA	東部標準時間	-5:00
America/Toronto	EDT	CA	東部夏時間	-4:00
America/Tortola	AST	VG	アトランティック標準時間	-4:00
America/Vancouver	PST	CA	太平洋標準時間	-8:00
America/Vancouver	PDT	CA	太平洋夏時間	-7:00
America/Virgin	AST	TT	アトランティック標準時間	-4:00
America/Whitehorse	PST	CA	太平洋標準時間	-8:00
America/Whitehorse	PDT	CA	太平洋夏時間	-7:00
America/Winnipeg	CST	CA	中部標準時間	-6:00
America/Winnipeg	CDT	CA	中部夏時間	-5:00
America/Yakutat	AKST	US	アラスカ標準時間	-9:00
America/Yakutat	AKDT	US	アラスカ夏時間	-8:00
America/Yellowknife	MST	CA	山岳部標準時間	-7:00
America/Yellowknife	MDT	CA	山岳部夏時間	-6:00

エリア: Antarctica (南極)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Antarctica/Casey	WST	AQ	オーストラリア西標準時間	8:00
Antarctica/Davis	DAVT	AQ	デービス時間	7:00
Antarctica/DumontDUrville	DDUT	AQ	デュモン・デュルヴィル時間	10:00
Antarctica/Macquarie	MIST	AU	マッコリー島時間	11:00
Antarctica/Mawson	MAWT	AQ	モーソン時間	5:00
Antarctica/McMurdo	NZST	AQ	ニュージーランド標準時間	12:00
Antarctica/McMurdo	NZDT	AQ	ニュージーランド夏時間	13:00
Antarctica/Palmer	CLT	AQ	チリ標準時間	-4:00
Antarctica/Palmer	CLST	AQ	チリ夏時間	-3:00
Antarctica/Rothera	ROTT	AQ	ロゼラ時間	-3:00
Antarctica/South_Pole	NZST	NZ	ニュージーランド標準時間	12:00
Antarctica/South_Pole	NZDT	NZ	ニュージーランド夏時間	13:00
Antarctica/Syowa	SYOT	AQ	昭和時間	3:00
Antarctica/Vostok	VOST	AQ	ボストーク時間	6:00

エリア: Asia (アジア)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Asia/Aden	AST	YE	アラビア標準時間	3:00
Asia/Almaty	ALMT	KZ	東カザフスタン時間	6:00
Asia/Amman	EET	JO	東ヨーロッパ標準時間	2:00
Asia/Amman	EEST	JO	東ヨーロッパ夏時間	3:00
Asia/Anadyr	ANAT	RU	マガダン標準時間	12:00
Asia/Aqtau	AQTT	KZ	西カザフスタン時間	5:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Asia/Aqtobe	AQTT	KZ	西カザフスタン時間	5:00
Asia/Ashgabat	TMT	TM	トルクメニアン標準時間	5:00
Asia/Ashkhabad	TMT	TM	トルクメニアン標準時間	5:00
Asia/Baghdad	AST	IQ	アラビア標準時間	3:00
Asia/Bahrain	AST	BH	アラビア標準時間	3:00
Asia/Baku	AZT	AZ	アゼルバイジャン標準時間	4:00
Asia/Baku	AZST	AZ	アゼルバイジャン夏時間	5:00
Asia/Bangkok	ICT	TH	インドシナ時間	7:00
Asia/Beijing	CST	CN	中国標準時間	8:00
Asia/Beirut	EET	LB	東ヨーロッパ標準時間	2:00
Asia/Beirut	EEST	LB	東ヨーロッパ夏時間	3:00
Asia/Bishkek	KGT	KG	キルギスタン時間	6:00
Asia/Brunei	BNT	BN	ブルネイ・ダルッサラム時間	8:00
Asia/Calcutta	IST	IN	インド標準時間	5:30
Asia/Choibalsan	CHOT	MN	チョイバルサン標準時間	8:00
Asia/Chongqing	CST	CN	中国標準時間	8:00
Asia/Chungking	CST	CN	中国標準時間	8:00
Asia/Colombo	IST	LK	インド標準時間	5:30
Asia/Dacca	BDT	BD	バングラデシュ標準時間	6:00
Asia/Damascus	EET	SY	東ヨーロッパ標準時間	2:00
Asia/Damascus	EEST	SY	東ヨーロッパ夏時間	3:00
Asia/Dhaka	BDT	BD	バングラデシュ標準時間	6:00
Asia/Dili	TLT	TL	東ティモール時間	9:00
Asia/Dubai	GST	AE	(ペルシア)湾標準時間	4:00
Asia/Dushanbe	TJT	TJ	タジキスタン時間	5:00
Asia/Gaza	EET	PS	東ヨーロッパ標準時間	2:00
Asia/Gaza	EEST	PS	東ヨーロッパ夏時間	3:00
Asia/Harbin	CST	CN	中国標準時間	8:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Asia/Hebron	EET	PS	東ヨーロッパ標準時間	2:00
Asia/Hebron	EEST	PS	東ヨーロッパ夏時間	3:00
Asia/Ho_Chi_Minh	ICT	VN	インドシナ時間	7:00
Asia/Hong_Kong	HKT	HK	香港標準時間	8:00
Asia/Hovd	HOVT	MN	ホブド標準時間	7:00
Asia/Irkutsk	IRKT	RU	イルクーツク標準時間	9:00
Asia/Istanbul	EET	TR	東ヨーロッパ標準時間	2:00
Asia/Istanbul	EEST	TR	東ヨーロッパ夏時間	3:00
Asia/Jakarta	WIB	ID	西インドネシア時間	7:00
Asia/Jayapura	WIT	ID	東インドネシア時間	9:00
Asia/Jerusalem	IST	IL	イスラエル標準時間	2:00
Asia/Jerusalem	IDT	IL	イスラエル夏時間	3:00
Asia/Kabul	AFT	AF	アフガニスタン時間	4:30
Asia/Kamchatka	PETT	RU	マガダン標準時間	12:00
Asia/Karachi	PKT	PK	パキスタン標準時間	5:00
Asia/Kashgar	CST	CN	中国標準時間	8:00
Asia/Kathmandu	NPT	NP	ネパール時間	5:45
Asia/Katmandu	NPT	NP	ネパール時間	5:45
Asia/Khandyga	YAKT	RU	ヤクーツク標準時間	10:00
Asia/Kolkata	IST	IN	インド標準時間	5:30
Asia/Krasnoyarsk	KRAT	RU	クラスノヤルスク標準時間	8:00
Asia/Kuala_Lumpur	MYT	MY	マレーシア時間	8:00
Asia/Kuching	MYT	MY	マレーシア時間	8:00
Asia/Kuwait	AST	KW	アラビア標準時間	3:00
Asia/Macao	CST	MO	中国標準時間	8:00
Asia/Macau	CST	MO	中国標準時間	8:00
Asia/Magadan	MAGT	RU	マガダン標準時間	12:00
Asia/Makassar	WITA	ID	中央インドネシア時間	8:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Asia/Manila	PHT	PH	フィリピン標準時間	8:00
Asia/Muscat	GST	OM	(ペルシア)湾標準時間	4:00
Asia/Nicosia	EET	CY	東ヨーロッパ標準時間	2:00
Asia/Nicosia	EEST	CY	東ヨーロッパ夏時間	3:00
Asia/Novokuznetsk	NOVT	RU	ノヴォシビルスク標準時間	7:00
Asia/Novosibirsk	NOVT	RU	ノヴォシビルスク標準時間	7:00
Asia/Omsk	OMST	RU	オムスク標準時間	7:00
Asia/Oral	ORAT	KZ	西カザフスタン時間	5:00
Asia/Osaka	JST	JP	日本標準時間	9:00
Asia/Phnom_Penh	ICT	KH	インドシナ時間	7:00
Asia/Pontianak	WIB	ID	西インドネシア時間	7:00
Asia/Pyongyang	KST	KP	韓国標準時間	9:00
Asia/Qatar	AST	QA	アラビア標準時間	3:00
Asia/Qyzylorda	QYZT	KZ	東カザフスタン時間	6:00
Asia/Rangoon	MMT	MM	ミャンマー時間	6:30
Asia/Riyadh	AST	SA	アラビア標準時間	3:00
Asia/Saigon	ICT	VN	インドシナ時間	7:00
Asia/Sakhalin	SAKT	RU	サハリン標準時間	11:00
Asia/Samarkand	UZT	UZ	ウズベキスタン標準時間	5:00
Asia/Sapporo	JST	JP	日本標準時間	9:00
Asia/Seoul	KST	KR	韓国標準時間	9:00
Asia/Shanghai	CST	CN	中国標準時間	8:00
Asia/Singapore	SGT	SG	シンガポール標準時間	8:00
Asia/Taipei	CST	TW	台北標準時間	8:00
Asia/Tashkent	UZT	UZ	ウズベキスタン標準時間	5:00
Asia/Tbilisi	GET	GE	グルジア標準時間	4:00
Asia/Tehran	IRST	IR	イラン標準時間	3:30
Asia/Tel_Aviv	IST	IL	イスラエル標準時間	2:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Asia/Tel_Aviv	IDT	IL	イスラエル夏時間	3:00
Asia/Thimbu	BTT	BT	ブータン時間	6:00
Asia/Thimphu	BTT	BT	ブータン時間	6:00
Asia/Tokyo	JST	JP	日本標準時間	9:00
Asia/Ujung_Pandang	WITA	ID	中央インドネシア時間	8:00
Asia/Ulaanbaatar	ULAT	MN	ウランバートル標準時間	8:00
Asia/Ulan_Bator	ULAT	MN	ウランバートル標準時間	8:00
Asia/Urumqi	CST	CN	中国標準時間	8:00
Asia/Ust-Nera	VLAT	RU	ウラジオストック標準時間	11:00
Asia/Vientiane	ICT	LA	インドシナ時間	7:00
Asia/Vladivostok	VLAT	RU	ウラジオストック標準時間	11:00
Asia/Yakutsk	YAKT	RU	ヤクーツク標準時間	10:00
Asia/Yekaterinburg	YEKT	RU	エカテリンブルク標準時間	6:00
Asia/Yerevan	AMT	AM	アルメニア標準時間	4:00

エリア: Atlantic (大西洋)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Atlantic/Azores	AZOT	PT	アゾレス標準時間	-1:00
Atlantic/Azores	AZOST	PT	アゾレス夏時間	00:00
Atlantic/Bermuda	AST	BM	アトランティック標準時間	-4:00
Atlantic/Bermuda	ADT	BM	アトランティック夏時間	-3:00
Atlantic/Canary	WET	ES	西ヨーロッパ標準時間	00:00
Atlantic/Canary	WEST	ES	西ヨーロッパ夏時間	1:00
Atlantic/Cape_Verde	CVT	CV	カーボベルデ標準時間	-1:00
Atlantic/Faeroe	WET	FO	西ヨーロッパ標準時間	00:00
Atlantic/Faeroe	WEST	FO	西ヨーロッパ夏時間	1:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Atlantic/Faroe	WET	FO	西ヨーロッパ標準時間	00:00
Atlantic/Faroe	WEST	FO	西ヨーロッパ夏時間	1:00
Atlantic/Jan_Mayen	CET	NO	中央ヨーロッパ標準時間	1:00
Atlantic/Jan_Mayen	CEST	NO	中央ヨーロッパ夏時間	2:00
Atlantic/Madeira	WET	PT	西ヨーロッパ標準時間	00:00
Atlantic/Madeira	WEST	PT	西ヨーロッパ夏時間	1:00
Atlantic/Reykjavik	GMT	IS	グリニッジ標準時間	00:00
Atlantic/South_Georgia	GST	GS	サウスジョージア時間	-2:00
Atlantic/St_Helena	GMT	SH	グリニッジ標準時間	00:00
Atlantic/Stanley	FKST	FK	フォークランド諸島標準時間	-3:00

エリア: Australia (オーストラリア)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Australia/ACT	EST	AU	オーストラリア東部標準時間	10:00
Australia/ACT	EDT	AU	オーストラリア東部夏時間	11:00
Australia/Adelaide	CST	AU	オーストラリア中部標準時間	09:30
Australia/Adelaide	CDT	AU	オーストラリア中部夏時間	10:30
Australia/Brisbane	EST	AU	オーストラリア東部標準時間	10:00
Australia/Broken_Hill	CST	AU	オーストラリア中部標準時間	09:30
Australia/Broken_Hill	CDT	AU	オーストラリア中部夏時間	10:30
Australia/Canberra	EST	AU	オーストラリア東部標準時間	10:00
Australia/Canberra	EDT	AU	オーストラリア東部夏時間	11:00
Australia/Currie	EST	AU	オーストラリア東部標準時間	10:00
Australia/Currie	EDT	AU	オーストラリア東部夏時間	11:00
Australia/Darwin	CST	AU	オーストラリア中部標準時間	09:30
Australia/Eucla	CWST	AU	オーストラリア中西部標準時間	8:45

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Australia/Hobart	EST	AU	オーストラリア東部標準時間	10:00
Australia/Hobart	EDT	AU	オーストラリア東部夏時間	11:00
Australia/LHI	LHST	AU	ロード・ハウ標準時間	10:30
Australia/LHI	LHDT	AU	ロード・ハウ夏時間	11:00
Australia/Lindeman	EST	AU	オーストラリア東部標準時間	10:00
Australia/Lord_Howe	LHST	AU	ロード・ハウ標準時間	10:30
Australia/Lord_Howe	LHDT	AU	ロード・ハウ夏時間	11:00
Australia/Melbourne	EST	AU	オーストラリア東部標準時間	10:00
Australia/Melbourne	EDT	AU	オーストラリア東部夏時間	11:00
Australia/NSW	EST	AU	オーストラリア東部標準時間	10:00
Australia/NSW	EDT	AU	オーストラリア東部夏時間	11:00
Australia/North	CST	AU	オーストラリア中部標準時間	09:30
Australia/Perth	WST	AU	オーストラリア西標準時間	8:00
Australia/Queensland	EST	AU	オーストラリア東部標準時間	10:00
Australia/South	CST	AU	オーストラリア中部標準時間	09:30
Australia/South	CDT	AU	オーストラリア中部夏時間	10:30
Australia/Sydney	EST	AU	オーストラリア東部標準時間	10:00
Australia/Sydney	EDT	AU	オーストラリア東部夏時間	11:00
Australia/Tasmania	EST	AU	オーストラリア東部標準時間	10:00
Australia/Tasmania	EDT	AU	オーストラリア東部夏時間	11:00
Australia/Victoria	EST	AU	オーストラリア東部標準時間	10:00
Australia/Victoria	EDT	AU	オーストラリア東部夏時間	11:00
Australia/West	WST	AU	オーストラリア西標準時間	8:00
Australia/Yancowinna	CST	AU	オーストラリア中部標準時間	09:30
Australia/Yancowinna	CDT	AU	オーストラリア中部夏時間	10:30

エリア: その他

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Arctic/Longyearbyen	CET	SJ	中央ヨーロッパ標準時間	1:00
Arctic/Longyearbyen	CEST	SJ	中央ヨーロッパ夏時間	2:00
Brazil/Acre	ACT	BR	アマゾン標準時間	-5:00
Brazil/DeNoronha	FNT	BR	フェルナンド・デ・ノローニャ標準時間	-2:00
Brazil/East	BRT	BR	ブラジリア標準時間	-3:00
Brazil/East	BRST	BR	ブラジリア夏時間	-2:00
Brazil/West	AMT	BR	アマゾン標準時間	-4:00
CST6CDT	CST		中部標準時間	-6:00
CST6CDT	CDT		中部夏時間	-5:00
Canada/Atlantic	AST	CA	アトランティック標準時間	-4:00
Canada/Atlantic	ADT	CA	アトランティック夏時間	-3:00
Canada/Central	CST	CA	中部標準時間	-6:00
Canada/Central	CDT	CA	中部夏時間	-5:00
Canada/East-Saskatchewan	CST	CA	中部標準時間	-6:00
Canada/Eastern	EST	CA	東部標準時間	-5:00
Canada/Eastern	EDT	CA	東部夏時間	-4:00
Canada/Mountain	MST	CA	山岳部標準時間	-7:00
Canada/Mountain	MDT	CA	山岳部夏時間	-6:00
Canada/Newfoundland	NST	CA	ニューファンドランド標準時間	-1:27
Canada/Newfoundland	NDT	CA	ニューファンドランド夏時間	-1:27
Canada/Pacific	PST	CA	太平洋標準時間	-8:00
Canada/Pacific	PDT	CA	太平洋夏時間	-7:00
Canada/Saskatchewan	CST	CA	中部標準時間	-6:00
Canada/Yukon	PST	CA	太平洋標準時間	-8:00
Canada/Yukon	PDT	CA	太平洋夏時間	-7:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Chile/Continental	CLT	CL	チリ標準時間	-4:00
Chile/Continental	CLST	CL	チリ夏時間	-3:00
Chile/EasterIsland	EAST	CL	イースター島標準時間	-6:00
Chile/EasterIsland	EASST	CL	イースター島夏時間	-5:00
Cuba	CST	CU	キューバ標準時間	-5:00
Cuba	CDT	CU	キューバ夏時間	-4:00
EST5EDT	EST		東部標準時間	-5:00
EST5EDT	EDT		東部夏時間	-4:00
Egypt	EET	EG	東ヨーロッパ標準時間	3:00
	EET	EG	東ヨーロッパ夏時間	0.00
Eire	GMT	IE	グリニッジ標準時間	00:00
Eire	IST	IE		1:00
Etc/GMT	GMT		グリニッジ標準時間	00:00
Etc/GMT+0	GMT		グリニッジ標準時間	00:00
Etc/GMT+1	GMT-01		GMT より 1 時間遅い	-1:00
Etc/GMT+10	GMT-10		GMT より 10 時間遅い	-10:00
Etc/GMT+11	GMT-11		GMT より 11 時間遅い	-11:00
Etc/GMT+12	GMT-12		GMT より 12 時間遅い	-12:00
Etc/GMT+2	GMT-02		GMT より 2 時間遅い	-2:00
Etc/GMT+3	GMT-03		GMT より 3 時間遅い	-3:00
Etc/GMT+4	GMT-04		GMT より 4 時間遅い	-4:00
Etc/GMT+5	GMT-05		GMT より 5 時間遅い	-5:00
Etc/GMT+6	GMT-06		GMT より 6 時間遅い	-6:00
Etc/GMT+7	GMT-07		GMT より 7 時間遅い	-7:00
Etc/GMT+8	GMT-08		GMT より 8 時間遅い	-8:00
Etc/GMT+9	GMT-09		GMT より 9 時間遅い	-9:00
Etc/GMT-0	GMT		グリニッジ標準時間	00:00
Etc/GMT-1	GMT+01		GMT より 1 時間早い	1:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Etc/GMT-10	GMT+10		GMT より 10 時間早い	10:00
Etc/GMT-11	GMT+11		GMT より 11 時間早い	11:00
Etc/GMT-12	GMT+12		GMT より 12 時間早い	12:00
Etc/GMT-13	GMT+13		GMT より 13 時間早い	13:00
Etc/GMT-14	GMT+14		GMT より 14 時間早い	14:00
Etc/GMT-2	GMT+02		GMT より 2 時間早い	2:00
Etc/GMT-3	GMT+03		GMT より 3 時間早い	3:00
Etc/GMT-4	GMT+04		GMT より 4 時間早い	4:00
Etc/GMT-5	GMT+05		GMT より 5 時間早い	5:00
Etc/GMT-6	GMT+06		GMT より 6 時間早い	6:00
Etc/GMT-7	GMT+07		GMT より 7 時間早い	7:00
Etc/GMT-8	GMT+08		GMT より 8 時間早い	8:00
Etc/GMT-9	GMT+09		GMT より 9 時間早い	9:00
Etc/GMT0	GMT		グリニッジ標準時間	00:00
Etc/Greenwich	GMT		グリニッジ標準時間	00:00
Etc/UCT	UCT		グリニッジ標準時間	00:00
Etc/UTC	UTC		グリニッジ標準時間	00:00
Etc/Universal	UTC		グリニッジ標準時間	00:00
Etc/Zulu	UTC		グリニッジ標準時間	00:00
GB	GMT	GB	グリニッジ標準時間	00:00
GB	BST	GB		1:00
GB-Eire	GMT	GB	グリニッジ標準時間	00:00
GB-Eire	BST	GB		1:00
GMT	GMT		グリニッジ標準時間	00:00
GMT+0	GMT		グリニッジ標準時間	00:00
GMT-0	GMT		グリニッジ標準時間	00:00
GMT0	GMT		グリニッジ標準時間	00:00
Greenwich	GMT		グリニッジ標準時間	00:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Hongkong	HKT	HK	香港標準時間	8:00
Iceland	GMT	IS	グリニッジ標準時間	00:00
Indian/Antananarivo	EAT	MG	東アフリカ時間	3:00
Indian/Chagos	IOT	IO	インド洋時間	6:00
Indian/Christmas	CXT	CX	クリスマス島時間	7:00
Indian/Cocos	CCT	CC	ココス島時間	6:30
Indian/Comoro	EAT	KM	東アフリカ時間	3:00
Indian/Kerguelen	TFT	TF	フランス領極南時間	5:00
Indian/Mahe	SCT	SC	セーシェル時間	4:00
Indian/Maldives	MVT	MV	モルディブ時間	5:00
Indian/Mauritius	MUT	MU	モーリシャス標準時間	4:00
Indian/Mayotte	EAT	YT	東アフリカ時間	3:00
Indian/Reunion	RET	RE	レユニオン時間	4:00
Iran	IRST	IR	イラン標準時間	3:30
Israel	IST	IL	イスラエル標準時間	2:00
Israel	IDT	IL	イスラエル夏時間	3:00
Jamaica	EST	JM	東部標準時間	-5:00
Japan	JST	JP	日本標準時間	9:00
Kwajalein	MHT	MH	マーシャル諸島時間	12:00
Libya	EET	LY	中央ヨーロッパ標準時間	2:00
MST7MDT	MST		山岳部標準時間	-7:00
MST7MDT	MDT		山岳部夏時間	-6:00
Mexico/BajaNorte	PST	MX	太平洋標準時間	-8:00
Mexico/BajaNorte	PDT	MX	太平洋夏時間	-7:00
Mexico/BajaSur	MST	MX	山岳部標準時間	-7:00
Mexico/BajaSur	MDT	MX	山岳部夏時間	-6:00
Mexico/General	CST	MX	中部標準時間	-6:00
Mexico/General	CDT	MX	中部夏時間	-5:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
NZ	NZST	NZ	ニュージーランド標準時間	12:00
NZ	NZDT	NZ	ニュージーランド夏時間	13:00
NZ-CHAT	CHAST	NZ	チャタム標準時間	12:45
NZ-CHAT	CHADT	NZ	チャタム夏時間	13:45
Navajo	MST	US	山岳部標準時間	-7:00
Navajo	MDT	US	山岳部夏時間	-6:00
PRC	CST	CN	中国標準時間	8:00
PST8PDT	PST		太平洋標準時間	-8:00
PST8PDT	PDT		太平洋夏時間	-7:00
Poland	CET	PL	中央ヨーロッパ標準時間	1:00
Poland	CEST	PL	中央ヨーロッパ夏時間	2:00
Portugal	WET	PT	西ヨーロッパ標準時間	00:00
Portugal	WEST	PT	西ヨーロッパ夏時間	1:00
ROC	CST	TW	台北標準時間	8:00
ROK	KST	KR	韓国標準時間	9:00
Singapore	SGT	SG	シンガポール標準時間	8:00
Turkey	EET	TR	東ヨーロッパ標準時間	2:00
Turkey	EEST	TR	東ヨーロッパ夏時間	3:00
UCT	UCT		グリニッジ標準時間	00:00
US/Alaska	AKST	US	アラスカ標準時間	-9:00
US/Alaska	AKDT	US	アラスカ夏時間	-8:00
US/Aleutian	HAST	US	ハワイ・アリューシャン標準時間	-10:00
US/Aleutian	HADT	US	ハワイ・アリューシャン夏時間	-9:00
US/Arizona	MST	US	山岳部標準時間	-7:00
US/Central	CST	US	中部標準時間	-6:00
US/Central	CDT	US	中部夏時間	-5:00
US/East-Indiana	EST	US	東部標準時間	-5:00
US/East-Indiana	EDT	US	東部夏時間	-4:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
US/Eastern	EST	US	東部標準時間	-5:00
US/Eastern	EDT	US	東部夏時間	-4:00
US/Hawaii	HST	US	ハワイ・アリューシャン標準時間	-10:00
US/Indiana-Starke	CST	US	中部標準時間	-6:00
US/Indiana-Starke	CDT	US	中部夏時間	-5:00
US/Michigan	EST	US	東部標準時間	-5:00
US/Michigan	EDT	US	東部夏時間	-4:00
US/Mountain	MST	US	山岳部標準時間	-7:00
US/Mountain	MDT	US	山岳部夏時間	-6:00
US/Pacific	PST	US	太平洋標準時間	-8:00
US/Pacific	PDT	US	太平洋夏時間	-7:00
US/Pacific-New	PST	US	太平洋標準時間	-8:00
US/Pacific-New	PDT	US	太平洋夏時間	-7:00
US/Samoa	SST	AS	サモア標準時間	-11:00
UTC	UTC		グリニッジ標準時間	00:00
Universal	UTC		グリニッジ標準時間	00:00
W-SU	MSK	RU	モスクワ標準時間	4:00
Zulu	UTC		グリニッジ標準時間	00:00

エリア: Europe (ヨーロッパ)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Europe/Amsterdam	CET	NL	中央ヨーロッパ標準時間	1:00
Europe/Amsterdam	CEST	NL	中央ヨーロッパ夏時間	2:00
Europe/Andorra	CET	AD	中央ヨーロッパ標準時間	1:00
Europe/Andorra	CEST	AD	中央ヨーロッパ夏時間	2:00
Europe/Athens	EET	GR	東ヨーロッパ標準時間	2:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Europe/Athens	EEST	GR	東ヨーロッパ夏時間	3:00
Europe/Belfast	GMT	GB	グリニッジ標準時間	00:00
Europe/Belfast	BST	GB		1:00
Europe/Belgrade	CET	RS	中央ヨーロッパ標準時間	1:00
Europe/Belgrade	CEST	RS	中央ヨーロッパ夏時間	2:00
Europe/Berlin	CET	DE	中央ヨーロッパ標準時間	1:00
Europe/Berlin	CEST	DE	中央ヨーロッパ夏時間	2:00
Europe/Bratislava	CET	SK	中央ヨーロッパ標準時間	1:00
Europe/Bratislava	CEST	SK	中央ヨーロッパ夏時間	2:00
Europe/Brussels	CET	BE	中央ヨーロッパ標準時間	1:00
Europe/Brussels	CEST	BE	中央ヨーロッパ夏時間	2:00
Europe/Bucharest	EET	RO	東ヨーロッパ標準時間	2:00
Europe/Bucharest	EEST	RO	東ヨーロッパ夏時間	3:00
Europe/Budapest	CET	HU	中央ヨーロッパ標準時間	1:00
Europe/Budapest	CEST	HU	中央ヨーロッパ夏時間	2:00
Europe/Busingen	CET	DE	中央ヨーロッパ標準時間	1:00
Europe/Busingen	CEST	DE	中央ヨーロッパ夏時間	2:00
Europe/Chisinau	EET	MD	東ヨーロッパ標準時間	2:00
Europe/Chisinau	EEST	MD	東ヨーロッパ夏時間	3:00
Europe/Copenhagen	CET	DK	中央ヨーロッパ標準時間	1:00
Europe/Copenhagen	CEST	DK	中央ヨーロッパ夏時間	2:00
Europe/Dublin	GMT	IE	グリニッジ標準時間	00:00
Europe/Dublin	IST	IE		1:00
Europe/Gibraltar	CET	GI	中央ヨーロッパ標準時間	1:00
Europe/Gibraltar	CEST	GI	中央ヨーロッパ夏時間	2:00
Europe/Guernsey	GMT	GG	グリニッジ標準時間	00:00
Europe/Guernsey	BST	GG		1:00
Europe/Helsinki	EET	FI	東ヨーロッパ標準時間	2:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Europe/Helsinki	EEST	FI	東ヨーロッパ夏時間	3:00
Europe/Isle_of_Man	GMT	IM	グリニッジ標準時間	00:00
Europe/Isle_of_Man	BST	IM		1:00
Europe/Istanbul	EET	TR	東ヨーロッパ標準時間	2:00
Europe/Istanbul	EEST	TR	東ヨーロッパ夏時間	3:00
Europe/Jersey	GMT	JE	グリニッジ標準時間	00:00
Europe/Jersey	BST	JE		1:00
Europe/Kaliningrad	FET	RU	東ヨーロッパ標準時間	3:00
Europe/Kiev	EET	UA	東ヨーロッパ標準時間	2:00
Europe/Kiev	EEST	UA	東ヨーロッパ夏時間	3:00
Europe/Lisbon	WET	PT	西ヨーロッパ標準時間	00:00
Europe/Lisbon	WEST	PT	西ヨーロッパ夏時間	1:00
Europe/Ljubljana	CET	SI	中央ヨーロッパ標準時間	1:00
Europe/Ljubljana	CEST	SI	中央ヨーロッパ夏時間	2:00
Europe/London	GMT	GB	グリニッジ標準時間	00:00
Europe/London	BST	GB		1:00
Europe/Luxembourg	CET	LU	中央ヨーロッパ標準時間	1:00
Europe/Luxembourg	CEST	LU	中央ヨーロッパ夏時間	2:00
Europe/Madrid	CET	ES	中央ヨーロッパ標準時間	1:00
Europe/Madrid	CEST	ES	中央ヨーロッパ夏時間	2:00
Europe/Malta	CET	MT	中央ヨーロッパ標準時間	1:00
Europe/Malta	CEST	MT	中央ヨーロッパ夏時間	2:00
Europe/Mariehamn	EET	AX	東ヨーロッパ標準時間	2:00
Europe/Mariehamn	EEST	AX	東ヨーロッパ夏時間	3:00
Europe/Minsk	FET	BY	東ヨーロッパ標準時間	3:00
Europe/Monaco	CET	MC	中央ヨーロッパ標準時間	1:00
Europe/Monaco	CEST	MC	中央ヨーロッパ夏時間	2:00
Europe/Moscow	MSK	RU	モスクワ標準時間	4:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Europe/Nicosia	EET	CY	東ヨーロッパ標準時間	2:00
Europe/Nicosia	EEST	CY	東ヨーロッパ夏時間	3:00
Europe/Oslo	CET	NO	中央ヨーロッパ標準時間	1:00
Europe/Oslo	CEST	NO	中央ヨーロッパ夏時間	2:00
Europe/Paris	CET	FR	中央ヨーロッパ標準時間	1:00
Europe/Paris	CEST	FR	中央ヨーロッパ夏時間	2:00
Europe/Podgorica	CET	ME	中央ヨーロッパ標準時間	1:00
Europe/Podgorica	CEST	ME	中央ヨーロッパ夏時間	2:00
Europe/Prague	CET	CZ	中央ヨーロッパ標準時間	1:00
Europe/Prague	CEST	CZ	中央ヨーロッパ夏時間	2:00
Europe/Riga	EET	LV	東ヨーロッパ標準時間	2:00
Europe/Riga	EEST	LV	東ヨーロッパ夏時間	3:00
Europe/Rome	CET	IT	中央ヨーロッパ標準時間	1:00
Europe/Rome	CEST	IT	中央ヨーロッパ夏時間	2:00
Europe/Samara	SAMT	RU	モスクワ標準時間	4:00
Europe/San_Marino	CET	SM	中央ヨーロッパ標準時間	1:00
Europe/San_Marino	CEST	SM	中央ヨーロッパ夏時間	2:00
Europe/Sarajevo	CET	BA	中央ヨーロッパ標準時間	1:00
Europe/Sarajevo	CEST	BA	中央ヨーロッパ夏時間	2:00
Europe/Simferopol	MSK	UA	東ヨーロッパ標準時間	4:00
Europe/Skopje	CET	MK	中央ヨーロッパ標準時間	1:00
Europe/Skopje	CEST	MK	中央ヨーロッパ夏時間	2:00
Europe/Sofia	EET	BG	東ヨーロッパ標準時間	2:00
Europe/Sofia	EEST	BG	東ヨーロッパ夏時間	3:00
Europe/Stockholm	CET	SE	中央ヨーロッパ標準時間	1:00
Europe/Stockholm	CEST	SE	中央ヨーロッパ夏時間	2:00
Europe/Tallinn	EET	EE	東ヨーロッパ標準時間	2:00
Europe/Tallinn	EEST	EE	東ヨーロッパ夏時間	3:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Europe/Tirane	CET	AL	中央ヨーロッパ標準時間	1:00
Europe/Tirane	CEST	AL	中央ヨーロッパ夏時間	2:00
Europe/Tiraspol	EET	MD	東ヨーロッパ標準時間	2:00
Europe/Tiraspol	EEST	MD	東ヨーロッパ夏時間	3:00
Europe/Uzhgorod	EET	UA	東ヨーロッパ標準時間	2:00
Europe/Uzhgorod	EEST	UA	東ヨーロッパ夏時間	3:00
Europe/Vaduz	CET	LI	中央ヨーロッパ標準時間	1:00
Europe/Vaduz	CEST	LI	中央ヨーロッパ夏時間	2:00
Europe/Vatican	CET	VA	中央ヨーロッパ標準時間	1:00
Europe/Vatican	CEST	VA	中央ヨーロッパ夏時間	2:00
Europe/Vienna	CET	AT	中央ヨーロッパ標準時間	1:00
Europe/Vienna	CEST	AT	中央ヨーロッパ夏時間	2:00
Europe/Vilnius	EET	LT	東ヨーロッパ標準時間	2:00
Europe/Vilnius	EEST	LT	東ヨーロッパ夏時間	3:00
Europe/Volgograd	VOLT	RU	ヴォルゴグラード標準時間	4:00
Europe/Warsaw	CET	PL	中央ヨーロッパ標準時間	1:00
Europe/Warsaw	CEST	PL	中央ヨーロッパ夏時間	2:00
Europe/Zagreb	CET	HR	中央ヨーロッパ標準時間	1:00
Europe/Zagreb	CEST	HR	中央ヨーロッパ夏時間	2:00
Europe/Zaporozhye	EET	UA	東ヨーロッパ標準時間	2:00
Europe/Zaporozhye	EEST	UA	東ヨーロッパ夏時間	3:00
Europe/Zurich	CET	CH	中央ヨーロッパ標準時間	1:00
Europe/Zurich	CEST	CH	中央ヨーロッパ夏時間	2:00

エリア: Pacific (太平洋)

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Pacific/Apia	WST	WS	サモア標準時間	13:00
Pacific/Apia	WSDT	WS	サモア夏時間	14:00
Pacific/Auckland	NZST	NZ	ニュージーランド標準時間	12:00
Pacific/Auckland	NZDT	NZ	ニュージーランド夏時間	13:00
Pacific/Chatham	CHAST	NZ	チャタム標準時間	12:45
Pacific/Chatham	CHADT	NZ	チャタム夏時間	13:45
Pacific/Chuuk	CHUT	FM	チューク時間	10:00
Pacific/Easter	EAST	CL	イースター島標準時間	-6:00
Pacific/Easter	EASST	CL	イースター島夏時間	-5:00
Pacific/Efate	VUT	VU	バヌアツ標準時間	11:00
Pacific/Enderbury	PHOT	KI	フェニックス諸島時間	13:00
Pacific/Fakaofu	TKT	TK	トケラウ時間	13:00
Pacific/Fiji	FJT	FJ	フィジー標準時間	12:00
	FJST	FJ	フィジー夏時間	12:00
Pacific/Funafuti	TVT	TV	ツバル時間	12:00
Pacific/Galapagos	GALT	EC	ガラパゴス時間	-6:00
Pacific/Gambier	GAMT	PF	ガンビア時間	-9:00
Pacific/Guadalcanal	SBT	SB	ソロモン諸島時間	11:00
Pacific/Guam	CHST	GU	チャモロ標準時間	10:00
Pacific/Honolulu	HST	US	ハワイ・アリューシャン標準時間	-10:00
Pacific/Johnston	HST	UM	ハワイ・アリューシャン標準時間	-10:00
Pacific/Kiritimati	LINT	KI	ライン諸島時間	14:00
Pacific/Kosrae	KOST	FM	コスラエ時間	11:00
Pacific/Kwajalein	MHT	MH	マーシャル諸島時間	12:00
Pacific/Majuro	MHT	MH	マーシャル諸島時間	12:00

タイムゾーン情報	タイムゾーン名	リージョン	タイムゾーンの説明	タイムゾーンのオフセット
Pacific/Marquesas	MART	PF	マルキーズ時間	-6:27
Pacific/Midway	SST	UM	サモア標準時間	-11:00
Pacific/Nauru	NRT	NR	ナウル時間	12:00
Pacific/Niue	NUT	NU	ニウエ時間	-11:00
Pacific/Norfolk	NFT	NF	ノーフォーク島時間	11:30
Pacific/Noumea	NCT	NC	ニューカレドニア標準時間	11:00
Pacific/Pago_Pago	SST	AS	サモア標準時間	-11:00
Pacific/Palau	PWT	PW	パラオ時間	9:00
Pacific/Pitcairn	PST	PN	ピトケアン時間	-8:00
Pacific/Pohnpei	PONT	FM	ポンペイ(ポナベ)時間	11:00
Pacific/Ponape	PONT	FM	ポンペイ(ポナベ)時間	11:00
Pacific/Port_Moresby	PGT	PG	パプアニューギニア時間	10:00
Pacific/Rarotonga	CKT	CK	クック諸島標準時間	-10:00
Pacific/Saipan	CHST	MP	チャモロ標準時間	10:00
Pacific/Samoa	SST	AS	サモア標準時間	-11:00
Pacific/Tahiti	TAHT	PF	タヒチ時間	-10:00
Pacific/Tarawa	GILT	KI	ギルバート諸島時間	12:00
Pacific/Tongatapu	TOT	TO	トンガ標準時間	13:00
Pacific/Truk	CHUT	FM	チューク時間	10:00
Pacific/Wake	WAKT	UM	ウェーク島時間	12:00
Pacific/Wallis	WFT	WF	ウォリス・フツナ時間	12:00
Pacific/Yap	CHUT	FM	チューク時間	10:00

推奨資料

このタイトルに関連した推奨される参考資料のリストを次に示します。

- [SAS Viya Data Management and Utility Procedures Guide](#)
- [SAS Viya Data Set Options: Reference](#)
- [SAS Viya 出力形式と入力形式: リファレンス](#)
- [SAS Viya 関数と CALL ルーチン: リファレンス](#)
- [SAS Viya システムオプション: リファレンス](#)
- [SAS Viya ステートメント: リファレンス](#)
- [SAS Encoding - Understanding the Details](#)

SAS 刊行物の一覧については、sas.com/store/books から入手できます。必要な書籍についての質問は SAS 担当者までお寄せください:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
電話: 1-800-727-0025
ファクシミリ: 1-919-677-4444
メール: sasbook@sas.com
Web アドレス: sas.com/store/books

用語集

1 バイト文字セット (SBCS)

1 バイトのコンピュータメモリを使用して 1 文字を表現するエンコーディング。1 バイト文字セットの例として、Latin 1 があります。 [関連項目: 2 バイト文字セット, マルチバイト文字セット](#)。

2 バイト文字セット (DBCS)

多くの文字が 2 バイトのメモリを占めるため可変幅エンコーディングを必要とする文字セット。実際は 1 バイトしか必要としない DBCS 文字もあるため、DBCS という用語(従来、日本語、韓国語および中国語などの言語に適用)は、多少誤解を招きやすいと言えます。 [関連項目: 1 バイト文字セット, マルチバイト文字セット](#)。

ANSI

[参照項目: 米国規格協会\(American National Standards Institute\)](#)。

ASCII (情報交換用米国標準コード(American Standard Code for Information Interchange))

基本的な 128 文字のセットを提供し、多様なコンピュータシステムを支援する 7 ビットエンコーディング規格。ASCII では、英文字の大文字と小文字、発音記号、数字(0-9)、コントロール文字をエンコードします。この 128 文字のセットは、他のほとんどのエンコーディングにも含まれています。 [関連項目: EBCDIC, エンコーディング](#)。

BOM

[参照項目: バイトオーダーマーク](#)。

CCS

[参照項目: コード化された文字セット](#)。

CEDA

[参照項目: クロス環境データアクセス](#)。

CJK

中国語、日本語および韓国語ではすべて漢字が使用されるため、まとめてグループと呼ばれます。このグループは翻訳と情報処理に関連します。ベトナム語では歴史的に漢字が使用されるため、ベトナム語がこのグループ(CJKV)に含まれることがあります。

DBCS

[参照項目: 2 バイト文字セット](#)。

EBCDIC (拡張 2 進化 10 進コード)

IBM メインフレームやミッドレンジコンピュータのデータを表現する 1 バイトやマルチバイトのエンコーディングファミリー。 [関連項目: ASCII, エンコーディング](#)。

EUC

[参照項目: 拡張 UNIX コード](#)。

I18N

[参照項目: 国際化](#)。

ICU (Unicode 用国際化コンポーネント)

ソフトウェアアプリケーション用の Unicode とグローバル化をサポートする C/C++ と Java のライブラリを含んでいるオープンソースのプロジェクト。

ISO

[参照項目: 国際標準化機構](#)。

ISO 646 ファミリ

ISO 646 規格に定義された、7 ビットエンコーディングの体系。エンコーディングには、116 の変更不可能な ASCII コード位置と、各国語文字で置き換え可能な 12 の変更可能なコード位置の両方が含まれます。たとえば、コード位置 23 (16 進値) は変更可能な文字のために予約されています。この位置は、米国では番号記号(#)、英国では記号(£)に割り当てられます。ISO 646 の各国版は大部分が現在使用されていません。

ISO 8859 ファミリ

ISO 8859 規格に定義された、16 の部からなる 8 ビットエンコーディングの体系。それぞれのエンコーディングに、そのサポート言語で使用される 128 個の ASCII 文字と 128 個の拡張文字の両方が含まれます。たとえば、Latin-1 とも呼ばれる ISO 8859-1 は、ISO 8859 ファミリ内でよく使用されるエンコーディングであり、ASCII 文字と西ヨーロッパ言語で使用される文字を含みます。 [関連項目: 国際標準化機構](#)。

L10N

[参照項目: 地域化](#)。

locale

地理的な地域の言語や地域規則および文化を反映させた設定。地域規則にはペーパーサイズや日付、時間、数字の出力規則、および国や地域の通貨記号が含まれます。ロケールの値の例として、French_Canada、Portuguese_Brazil、Chinese_Singapore などがあります。

MBCS

[参照項目: マルチバイト文字セット](#)。

NLS

[参照項目: 各国語サポート](#)。

SBCS

[参照項目: 1 バイト文字セット](#)。

Unicode

世界のほとんどの表記体系に含まれている文字や記号の交換、処理、表示をサポートするための業界標準である 16 ビットエンコーディング。

Unicode 用国際化コンポーネント

参照項目: [ICU](#).

アクセント記号付き文字

発音を変えるアクセント記号を付けられて修飾された文字。たとえば、"ñ"はチルダ(~)と文字"n"の組み合わせです。

エンコーディング

コード化した文字セットをコード値にマッピングすること。

エンコーディング方法

エンコードされた文字体系を生成するために、制定された業界ルールをコード化された文字に適用すること。そのようなルールでは、特定の文字の数値表現およびエンコーディングでのそのコード位置を保存するために必要とされるビット数が規定されます。ISO 2022 と UTF-8 は、エンコーディング方法の一例です。

エンコード

特定の文字エンコーディング体系におけるデータの表記。たとえば、ASCII では、文字"A"は 16 進表現の 41 として表記されます。

表語文字

発音ではなく、語や形態素を表す視覚的な記号。中国の表語文字の例として、"mountain"という語を表す山があります。

拡張 2 進化 10 進コード

参照項目: [EBCDIC](#).

拡張 UNIX コード (EUC)

7 ビットコードの列に含まれる最大 94 文字を使用して主に中国語、日本語および韓国語の記述体系をエンコーディングするために使用されるマルチバイトエンコーディング体系。

各国語サポート (NLS)

ソフトウェアを各国の市場向けに正しく動作させるために必要な機能の集合。

各国語文字(ナショナルキャラクタ)

国または国グループで言語を表記する際の言語固有の文字。たとえば、"ñ"はチルダを付けた文字(ñ)はスペイン語の各国語文字です。

グラフィック文字

ISO 規格では、人が読める形式で書き込まれ、出力され、またはそうでなければ表示されることを意図される、エンコードされた文字。グラフィック文字は、1 つ以上のグリフと関連しています。

クロス環境データアクセス (CEDA)

ディレクトリベース動作環境で作成された SAS データファイルを、別のディレクトリベース環境の SAS セッションで読み込めるようにする SAS の機能。 [関連項目: データ表記](#).

コード化された文字セット (CCS)

抽象文字範囲と数値セットのマッピング。ISO Latin-1 コード化された文字セットでは、西ヨーロッパのアルファベットと記号とその数値表現が提供されます。たとえば、文字"À"は 16 進表現の C4 として表記されます。

コードページ

16進値と各文字を関連付ける文字セットの表現。コードページという用語は、EBCDICベースであるIBMのメインフレームシステムに由来しますが、Microsoft、SAP、Oracle Corporationなど多くのベンダがこの用語を使っています。ベンダはしばしば、独自のコードページ番号を文字エンコーディングに割り当てます。ほかによく知られた名前があったとしてもです(たとえばUTF-8文字エンコーディングに対するコードページ番号は、IBMでは1208、Microsoftでは65001、SAPでは4110です)。[関連項目: エンコーディング, コードページ.](#)

国際化 (I18N)

1つの言語またはロケールに基づくという前提なしで行われる、ソフトウェア製品の設計処理。地域化が容易になります。国際化により、文字列の並べ替えと日付、時間、数、通貨のフォーマティングのルールを含む、国際的な規則がサポートされます。また、一貫したユーザーエクスペリエンスを製品のどの言語版でも可能にします。[関連項目: 各国語サポート.](#)

国際標準化機構 (ISO)

規格の策定を推進し、国家間での製品、サービスおよび情報の共有を促進するための関連活動を後援する組織。

コントロール文字

文字セット内でコードポイントにより表現される非出力文字で、その文字自身は書き込まれる記号を表現していません。

照合順序

テキストデータの並べ替えや比較を決定するルールのセット。

情報交換用米国標準コード(American Standard Code for Information Interchange)

[参照項目: ASCII.](#)

双方向テキスト

アラビア語やヘブライ語などの記述体系のテキストで、通常、数字や他の言語の埋め込みテキストを除き、文字の向きが右から左になります。数字や他の言語のテキストでは、左から右の向きになります。

地域化 (L10N)

ソフトウェアを特定の地理文化的地域(ロケール)に適合させるプロセス。ユーザーインターフェイス、システムメッセージおよびドキュメントの翻訳が地域化プロセスの大部分を占めます。[関連項目: 国際化.](#)

データ表記

特定の動作環境でデータを保存するときの形式。動作環境が異なると、使用される規格や規則も異なります。たとえば、浮動小数点数を保存する場合はIEEEやIBM 390、文字エンコーディングの場合はASCIIまたはEBCDIC、メモリ内のバイトオーダーではビッグエンディアンまたはリトルエンディアン、ワード配置では4バイト境界または8バイト境界、データ型の長さでは16ビット、32ビット、64ビットが使用されます。

特殊文字

英数字、アンダースコア(_)、空白以外の文字。たとえば、アスタリスク(*)などがあります。

トランスコーディング

1つのエンコーディングから別のエンコーディングに SAS ファイルの内容を変換する処理。トランスコーディングが必要になるのは、UNIX の Latin 1 エンコーディングから IBM メインフレームのドイツ語 EBCDIC エンコーディングにデータを転送する場合のように、セッションエンコーディングとファイルエンコーディングが異なる場合です。 [関連項目: エンコーディング, 変換テーブル](#).

入力方式

対話型データ入力、一般にはキーボードからのデータ入力のために設計された文字変換のタイプ。この最も単純な入力方式は、ASCII 文字を別のアルファベットにマッピングすることで機能します。別の方式は、文字の列を単一の文字に変換する合成を使用することです。タイ語や韓国語などの言語ではどちらも使用できます。

バイトオーダーマーク (BOM)

テキストファイルまたはストリームの中で、後続の Unicode テキストのバイトオーダーを示す Unicode 文字。BOM は、複数の Unicode エンコーディングのうちどのエンコーディングとしてテキストストリームがエンコードされているかも示しています。UTF-16 では、バイトオーダーマークのコードポイントは、U+FEFF(16 進数)です。

ハングル

韓国語の文字体系の名前で、24 の子音と母音の文字から構成され、音節ごとにブロック単位で整列されています。

米国規格協会(American National Standards Institute) (ANSI)

米国の製品、サービス、プロセス、システム、人材の自発的な合意基準の策定の調整を計る組織。ANSI は国際規格を作成するために、ISO (国際標準化機構)と協力して活動しています。

変換テーブル

1つのエンコーディングから別のエンコーディングにデータをマッピングするために使用する SAS カタログエントリ。ロケール値とエンコーディングプロパティを制御する SAS 言語要素によって適切な変換テーブルが呼び出されます。変換テーブルは動作環境に固有です。たとえば、Windows Latin 2 エンコーディングを ISO Latin 2 エンコーディングにマッピングする変換テーブルがあります。 [関連項目: エンコーディング, トランスコーディング](#).

マルチバイト文字セット (MBCS)

多くの文字が 2 バイト以上のメモリを占めるため可変幅エンコーディングを必要とする文字セット。DBCS と MBCS は相互に交換可能な場合がありますが、日本語、韓国語、中国語のような言語の文字セットを記述する場合は MBCS の方が正確です。 [関連項目: 1 バイト文字セット, 2 バイト文字セット](#).

文字

意味を持つ、記述体系の最小構成要素(英文字、数字、表意文字など)。

文字エンコーディング

抽象文字範囲と数値セットのマッピング。文字エンコーディングは、計算、データ保存およびテキストデータの転送で使用されます。文字エンコーディングには、各国語文字、特殊文字、0 から 9 の数字、コントロール文字などがあります。

キーワード

\$

\$CPTDW 出力形式 88
 \$CPTWD 出力形式 89
 \$EBCDICw.出力形式 90
 \$UCS2BE 出力形式 243
 \$UCS2BE 入力形式 511
 \$UCS2B 出力形式 242
 \$UCS2B 入力形式 510
 \$UCS2LE 出力形式 246
 \$UCS2LE 入力形式 513
 \$UCS2L 出力形式 244
 \$UCS2L 入力形式 512
 \$UCS2XE 出力形式 248
 \$UCS2XE 入力形式 516
 \$UCS2X 出力形式 247
 \$UCS2X 入力形式 514
 \$UCS4BE 出力形式 251
 \$UCS4B 出力形式 249
 \$UCS4B 入力形式 517
 \$UCS4LE 出力形式 253
 \$UCS4L 出力形式 252
 \$UCS4L 入力形式 518
 \$UCS4XE 入力形式 520
 \$UCS4X 出力形式 254, 256
 \$UCS4X 入力形式 519
 \$UESCE 出力形式 258
 \$UESCE 入力形式 522
 \$UESC 出力形式 257
 \$UESC 入力形式 521
 \$UNCRE 出力形式 260
 \$UNCRE 入力形式 524
 \$UNCR 出力形式 259
 \$UNCR 入力形式 523
 \$UPARENE 出力形式 262
 \$UPARENE 入力形式 527
 \$UPAREN 出力形式 261
 \$UPAREN 入力形式 525
 \$UTF8XE 出力形式 264
 \$UTF8X 出力形式 263
 \$UTF8X 入力形式 529

%

%KCMPRES 534
 %KINDEX マクロ関数 534
 %KLEFT マクロ関数 535
 %KLENGTH マクロ関数 535
 %KLOWCASE 自動呼び出しマクロ 35
 %KSCAN マクロ関数 536
 %KSUBSTR マクロ関数 538
 %KTRIM 自動呼び出しマクロ 36
 %KVERIFY 自動呼び出しマクロ 37
 %QKCMPRES 534
 %QKTRIM 自動呼び出しマクロ 36

1

1 バイト文字セット(SBCS)
 データのトランスコーディングで使用されるエンコーディング値
 613

2

2 バイト文字
 文字式の数 322

8

8859 ISO ファミリ 11

A

ANORM420 関数 307
 ANSI (American National Standards Institute、米国規格協会) 13
 ASCII 11
 EBCDIC 間でのデータ転送 22
 ATTRIB ステートメント 590
 TRANSCODE=オプション 591

B

B8601DXw.出力形式 82
 B8601LXw.出力形式 83
 B8601TXw.出力形式 85

BASECHAR 関数 312
 BESTDOTX 出力形式 86
 BOMFILE システムオプション 546

C

CHARSET=オプション 570
 CPTDW 入力形式 414
 CPTWD 入力形式 415
 CVPBYTES オプション 577

D

DATESTYLE=システムオプション
 デフォルト値 606
 DBCS エンコーディング
 先頭の空白の削除 328
 データのトランスコーディングで使
 用されるエンコーディング値
 613
 半角/全角の空白の削除 349
 末尾の空白の削除 339
 文字データ切り捨て 23
 DFLANG=システムオプション 547
 デフォルト値 606

E

E8601DXw.出力形式 91
 E8601LXw.出力形式 92
 E8601TXw.出力形式 94
 EBCDIC 11
 ASCII 間でのデータ転送 22
 エンコーディングのコードポイント
 差異 13
 文字データの変換 90
 ENCODCOMPAT 関数 313
 ENCODING=システムオプション
 Posix 値 606
 デフォルト設定 606
 ENCODING=データセットオプション
 41
 ENCODING オプション 583
 ENCODING システムオプション 549
 ENCODISVALID 関数 314
 EUROX 出力形式 98
 EUROX 入力形式 418
 EURO 出力形式 95
 EURO 入力形式 416

G

GETLOCENV 関数 315
 GETPXLANGUAGE 関数 316
 GETPXLOCALE 関数 317
 GETPXREGION 関数 318, 319

I

IBw.d 入力形式 501, 502, 503, 505
 INENCODING=オプションおよび
 OUTENCODING=オプション 587
 ISO 8601 日時出力形式
 B8601DX 出力形式, 基本日時表記,
 タイムゾーンオフセット 82
 B8601LX 出力形式, 基本日時表記,
 タイムゾーンオフセット 83
 B8601TX 出力形式, 基本時間表記,
 タイムゾーンオフセット 85
 E8601DX 出力形式, 基本日時表記,
 タイムゾーンオフセット 91
 E8601LX 出力形式, 基本日時表記, タ
 イムゾーンオフセット 92
 E8601TX 出力形式, 基本時間表記, タ
 イムゾーンオフセット 94
 ISO(国際標準化機構) 13
 ISO エンコーディング 11
 8859 ファミリ 11
 Windows ファミリ 11

K

KCOMPARE 関数 319
 KCOMPRESS 関数 321
 KCOUNT 関数 322
 KCVT 関数 322
 KINDEXB 関数 325
 KINDEXCB 関数 327
 KINDEXC 関数 326
 KINDEX 関数 324
 KLEFT 関数 328
 KLENGTH 関数 330
 KLOWCASE 関数 331
 KPROPCASE 関数 332
 KPROPCHAR 関数 335
 KPROPDATA 関数 336
 KREVERSE 関数 338
 KRIGHT 関数 339
 KSCAN 関数 340
 KSTRCAT 関数 341
 KSTRIP 関数 342
 KSUBSTRB 関数 346
 KSUBSTRN 関数 345
 KSUBSTR 関数 344
 KTRANSLATE 関数 348
 KTRIM 関数 349
 KTRUNCATE 関数 350
 KUPCASE 関数 351
 KUPCASE マクロ関数 540
 KUPDATEB 関数 354
 KUPDATES 関数 356
 KUPDATE 関数 352
 KVERIFYB 関数 359
 KVERIFY 関数 358

K 関数 279

L

LOCALE=システムオプション
値 599
LOCALEDATA システムオプション
551
LOCALE システムオプション 550
LOGLANGCHG システムオプション
552
LOGLANGENG システムオプション
553
LSWLANG システムオプション 554

M

MAPEBCDIC2ASCII システムオプション
555
META 宣言 570
MINGUO 出力形式 100
MINGUO 入力形式 419

N

NENGO 出力形式 102
NENGO 入力形式 421
NLBEST 出力形式 103
NLDATEL 出力形式 106
NLDATEMDL 出力形式 109
NLDATEMDM 出力形式 110, 132
NLDATEMDS 出力形式 111
NLDATEMD 出力形式 108
NLDATEMN 出力形式 112
NLDATEMYL 出力形式 118
NLDATEM 出力形式 107
NLDATES 出力形式 113
NLDATEWN 出力形式 115
NLDATEW 出力形式 114
NLDATEW 入力形式 423
NLDATEYMM 出力形式 118
NLDATEYMS 出力形式 119
NLDATEYM 出力形式 116
NLDATEYQL 出力形式 121
NLDATEYQM 出力形式 122
NLDATEYQS 出力形式 123
NLDATEYQ 出力形式 120
NLDATEYR 出力形式 124
NLDATEYW 出力形式 125
NLDATE 関数 360
NLDATE 出力形式 105
NLDATE 入力形式 422
NLDATMAP 出力形式 127
NLDATMAP 入力形式 425
NLDATMDT 出力形式 128
NLDATML 出力形式 129

NLDATMMDL 出力形式 131
NLDATMMDS 出力形式 133
NLDATMMD 出力形式 130
NLDATMMN 出力形式 134
NLDATMM 出力形式 130
NLDATMS 出力形式 135
NLDATMTM 出力形式 136
NLDATMTZ 出力形式
日時値 137
NLDATMWN 出力形式 139
NLDATMWZ 出力形式
日時値 140
NLDATMW 出力形式 138
NLDATMW 入力形式 426
NLDATMYML 出力形式 142
NLDATMYMM 出力形式 143
NLDATMYMS 出力形式 143
NLDATMYM 出力形式 141
NLDATMYQL 出力形式 145
NLDATMYQM 出力形式 146
NLDATMYQS 出力形式 147
NLDATMYQ 出力形式 144
NLDATMYR 出力形式 148
NLDATMYW 出力形式 149
NLDATMZ 出力形式
日時値 150
NLDATM 出力形式 126
NLDATM 入力形式 424
NLDECSEPARATOR システムオプション
556
NLMNIAED 出力形式 151
NLMNIAED 入力形式 427, 428
NLMNIAUD 出力形式 152
NLMNIBGN 出力形式 153
NLMNIBGN 入力形式 429
NLMNIBRLw.d 入力形式 430
NLMNIBRL 出力形式 154
NLMNICADw.d 入力形式 431
NLMNICAD 出力形式 155
NLMNICHFW.d 入力形式 432
NLMNICHF 出力形式 156
NLMNICNYw.d 入力形式 433
NLMNICNY 出力形式 157
NLMNICZKw.d 入力形式 434
NLMNICZK 出力形式 158
NLMNIDKKw.d 入力形式 435
NLMNIDKK 出力形式 159
NLMNIEEKw.d 入力形式 436
NLMNIEEK 出力形式 160
NLMNIEGPw.d 入力形式 437
NLMNIEGP 出力形式 161
NLMNIEURw.d 入力形式 438
NLMNIEUR 出力形式 162
NLMNIGBPw.d 入力形式 439
NLMNIGBP 出力形式 163
NLMNIHKDw.d 入力形式 440

- NLMNIHKD 出力形式 164
NLMNIHRKw.d 入力形式 441
NLMNIHRK 出力形式 165
NLMNIHUFw.d 入力形式 442
NLMNIHUF 出力形式 166
NLMNIIDRW.d 入力形式 443
NLMNIIDR 出力形式 167
NLMNIILSw.d 入力形式 444
NLMNIILS 出力形式 168
NLMNIINRW.d 入力形式 445
NLMNIINR 出力形式 169
NLMNIJPYw.d 入力形式 446
NLMNIJPY 出力形式 170
NLMNIKRWw.d 入力形式 447
NLMNIKRW 出力形式 171
NLMNITLW.d 入力形式 448
NLMNITL 出力形式 172
NLMNILVLW.d 入力形式 449
NLMNILVL 出力形式 173
NLMNIMOPw.d 入力形式 450
NLMNIMOP 出力形式 174
NLMNIMXNw.d 入力形式 451
NLMNIMXN 出力形式 175
NLMNIMYRW.d 入力形式 452
NLMNIMYR 出力形式 176
NLMNINOKw.d 入力形式 453
NLMNINOK 出力形式 177
NLMNINZDw.d 入力形式 454
NLMNINZD 出力形式 178
NLMNIPLNw.d 入力形式 455
NLMNIPLN 出力形式 179
NLMNIRUBw.d 入力形式 456
NLMNIRUB 出力形式 180
NLMNISEKw.d 入力形式 457
NLMNISEK 出力形式 181
NLMNISGDw.d 入力形式 458
NLMNISGD 出力形式 182
NLMNITHBw.d 入力形式 459
NLMNITHB 出力形式 183
NLMNITRYw.d 入力形式 460
NLMNITRY 出力形式 184
NLMNITWDw.d 入力形式 461
NLMNITWD 出力形式 185
NLMNIUSDw.d 入力形式 462
NLMNIUSD 出力形式 186
NLMNIZARw.d 入力形式 463
NLMNIZAR 出力形式 187
NLMNLAEDw.d 入力形式 464
NLMNLAED 出力形式 188
NLMNLAUDw.d 入力形式 465
NLMNLAUD 出力形式 189
NLMNLBGNw.d 入力形式 466
NLMNLBGN 出力形式 190
NLMNLBRLw.d 入力形式 467
NLMNLBRL 出力形式 191
NLMNLCADw.d 入力形式 468
NLMNLCAD 出力形式 192
NLMNLCHFw.d 入力形式 469
NLMNLCHF 出力形式 193
NLMNLCNYw.d 入力形式 470
NLMNLCNY 出力形式 194
NLMNLCZKw.d 入力形式 471
NLMNLCZK 出力形式 195
NLMNLDKKw.d 入力形式 472
NLMNLDKK 出力形式 196
NLMNLEEKw.d 入力形式 473
NLMNLEEK 出力形式 197
NLMNLEGPw.d 入力形式 474
NLMNLEGP 出力形式 198
NLMNLEURw.d 入力形式 475
NLMNLEUR 出力形式 199
NLMNLGBPw.d 入力形式 476
NLMNLGBP 出力形式 200
NLMNLHKDw.d 入力形式 477
NLMNLHKD 出力形式 201
NLMNLHRKw.d 入力形式 478
NLMNLHRK 出力形式 202
NLMNLHUFw.d 入力形式 479
NLMNLHUF 出力形式 203
NLMNLIDRW.d 入力形式 480
NLMNLIDR 出力形式 204
NLMNLILSw.d 入力形式 481
NLMNLILS 出力形式 205
NLMNLINRW.d 入力形式 482
NLMNLINR 出力形式 206
NLMNLJPYw.d 入力形式 483
NLMNLJPY 出力形式 207
NLMNLKRWw.d 入力形式 484
NLMNLKRW 出力形式 208
NLMNLLTLw.d 入力形式 485
NLMNLLTL 出力形式 209
NLMNLLVLw.d 入力形式 486
NLMNLLVL 出力形式 210
NLMNLMOPw.d 入力形式 487
NLMNLMOP 出力形式 211
NLMNLMXNw.d 入力形式 488
NLMNLMXN 出力形式 212
NLMNLMYRW.d 入力形式 489
NLMNLMYR 出力形式 213
NLMNLNOKw.d 入力形式 490
NLMNLNOK 出力形式 214
NLMNLNZDw.d 入力形式 491
NLMNLNZD 出力形式 215
NLMNLPLNw.d 入力形式 492
NLMNLPLN 出力形式 216
NLMNLRUBw.d 入力形式 493
NLMNLRUB 出力形式 217
NLMNLSEKw.d 入力形式 494
NLMNLSEK 出力形式 218
NLMNLSGDw.d 入力形式 495
NLMNLSGD 出力形式 219
NLMNLTHBw.d 入力形式 496

NLMNLTHB 出力形式 220
 NLMNLTRYw.d 入力形式 497
 NLMNLTRY 出力形式 221
 NLMNLTWDw.d 入力形式 498
 NLMNLTWD 出力形式 222
 NLMNLUSDw.d 入力形式 499
 NLMNLUSD 出力形式 223
 NLMNLZARw.d 入力形式 500
 NLMNLZAR 出力形式 224
 NLMNYI 出力形式 226
 NLMNY 出力形式 225
 NLNUMI 出力形式 229
 NLNUM 出力形式 227
 NLPCTIw.d 入力形式 507
 NLPCTI 出力形式 232
 NLPCTN 出力形式 233
 NLPCTP 出力形式 234
 NLPCTw.d 入力形式 506
 NLPCT 出力形式 230
 NLPVALUE 出力形式 235
 NLS(各国語サポート) 3
 エンコーディング 9
 トランスコーディング 19
 ロケール 5
 NLSTRMON 出力形式 236
 NLSTRQTR 出力形式 237
 NLSTRWK 出力形式 239
 NLTIMAPw.入力形式 508
 NLTIMAP 出力形式 240
 NLTIMEw.入力形式 509
 NLTIME 関数 366
 NLTIME 出力形式 241
 NLTIME ディスクリプタ 366
 NODATM 関数 363
 N 各国語サポート(NLS)
 出力形式 47

O
 ODSCHARSET オプション 588
 ODSLANGCHG システムオプション
 557
 ODS 出力
 テキストの言語の変更 557
 OPT=オプション, TRANTAB ステート
 メント 594
 OUTREP データセットオプション 44

P
 PAPERSIZE=システムオプション
 デフォルト値 606
 PROC SQL の TRANSCODE=列修飾子
 590

R
 RSASIoTRANSERROR システムオプシ
 ョン 559

S
 SASMSG 関数 370
 SASMSG 関数 367
 SAS 言語要素
 エンコーディング値の使用 613
 SAS セッション
 エンコーディングの動作 17
 デフォルトの文字セットエンコーデ
 ィング 549
 ロケール 550
 ロケールの指定 7
 SAS データセット
 命名 562
 SAS データビュー
 命名 562
 SAS 文字列関数
 国際化の互換性 279
 SBCS エンコーディン
 データのトランスコーディングで使
 用されるエンコーディング値
 613
 SETLOCALE 関数 375
 SO/SI (シフトアウト/シフトイン)
 文字式からの削除 349
 SORTKEY 関数 372
 SORTSEQ=システムオプション 559
 SORT オプション 571
 SORT プロシジャ
 言語固有の照合順序 559
 SQL プロシジャ
 言語固有の照合順序 559

T
 TIMESZONE=システムオプション
 560
 TRANSCODE=オプション
 ATTRIB ステートメント 591
 TRANTAB-ODS オプション 589, 593
 TRANTAB 関数 392
 TRANTAB ステートメント
 UPLOAD プロシジャ 594
 TYPE=オプション, TRANTAB ステート
 メント 594
 TZ=システムオプション 560
 TZID
 TZONID 関数 382
 TZONEDSTOFF 関数 388
 TZONID 関数 382
 TZONENAME 関数 384
 TZONEOFF 関数 385

TZONES2U 関数 386
 TZONESTTNAME 関数 389
 TZONESTTOFF 関数 390
 TZONEU2S 関数 391

U

Unicode 10
 外部ファイルの BOM 接頭辞 546
 データのトランスコーディングで使
 用されるエンコーディング値
 613
 表示単位の長さ 397
 文字単位の長さ 396
 UNICODDEC 関数 394
 UNICODELEN 関数 396
 UNICODEWIDTH 関数 397
 UNICODE 関数 393
 Unicode コンソーシアム 13
 UPLOAD プロシジャ
 TRANTAB テートメント 594
 UTF-16 11
 UTF-32 11
 UTF-8 11

V

VALIDMEMNAME システムオプション
 562
 VALIDVARNAME=システムオプション
 564
 VARTRANSCODE 関数 397
 VTRANSCODEX 関数 400
 VTRANSCODE 関数 399

W

WEEKU 出力形式 265
 WEEKV 出力形式 267
 WEEKW 出力形式 268
 Windows
 ISO エンコーディング 11

X

XMLLENENCODING オプション 593

Y

YEN 出力形式 270
 YYWEEKU 出力形式 271
 YYWEEKV 出力形式 272
 YYWEEKW 出力形式 274

あ

アイテムストア
 命名 562
 アイルランド
 通貨データ 162, 199
 アラブ首長国連邦
 通貨データ 151, 188
 イスラエル
 通貨データ 168, 205
 イタリア
 通貨データ 162, 199
 位置調整
 文字式 328, 339
 引数
 大文字への変換 351
 小文字への変換 331
 指定した引数のトランスコーディン
 グ 400
 長さ 330
 部分文字列の抽出 344
 部分文字列の抽出、バイト位置に基
 づく 346
 インド
 通貨データ 169, 206
 インドネシア
 通貨データ 167, 204
 英国
 通貨データ 163
 エジプト
 通貨データ 161, 198
 エストニア
 通貨データ 160, 197
 エンコーディング
 データセット 16
 リリース別データセットサポート
 17
 エンコーディング 9
 1つのデータタイプから別のデータ
 タイプへの変換 322
 SAS セッションでの動作 17
 外部ファイルの読み込みと書き込み
 18
 混在 43
 出力処理 17
 手動設定 41
 対トランスコーディング 10
 定義 10
 トランスコーディングの互換性 21
 入力処理 17
 東アジア言語 11
 標準組織 13
 エンコーディング値 613
 DBCS 613
 for transcoding data 613
 SBCS 613
 Unicode 613

エンコーディングの手動設定 41
 エンコーディング方法 9, 10
 オーストラリア
 通貨データ 152, 189
 オーストリア
 通貨データ 199
 大文字
 引数の変換 351
 小文字の変更 35
 大文字小文字
 大文字から小文字への変更 35
 オランダ
 通貨データ 162, 199

か

改行文字 22
 外部ファイル
 Unicode ファイルの BOM 接頭辞 546
 エンコーディング 18
 各国語サポート
 参照項目: NLS(各国語サポート)
 カナダ
 通貨データ 155, 192
 韓国
 通貨データ 171, 208
 関数
 K 関数 279
 SAS 文字列関数 279
 カテゴリ別 302
 ギリシャ
 通貨データ 162, 199
 空白
 先頭の DBCS の空白の削除 328
 末尾の DBCS の空白の削除 339
 末尾の削除 36
 グリーンランド
 通貨データ 159, 196, 200
 クロアチア
 通貨データ 165, 202
 言語 5
 国際化対応の日付の出力形式と入力形式 547
 言語切り替え 7
 ODS 出力のテキスト言語の変更 557
 言語コード
 現在の 2 文字のコード 316
 言語照合 16
 言語並べ替えキー 372
 検索
 単語、文字列内の位置 536
 文字式の特定の文字 326
 文字式の文字列 324
 コードページ 9

互換性

SAS 文字列関数 279
 互換性のあるエンコーディング 21
 互換性のないエンコーディング 21
 国際化 4
 国際化対応の日付の出力形式と入力形式
 言語の指定 547
 国際化対応の日付、日時出力形式 47
 国際標準化機構(ISO) 13
 小文字
 引数の変換 331
 大文字の変更 35

さ

時間 6
 時間値
 指定したロケールへの変換 366
 システムオプション
 トランスコーディング 20
 シフトアウト/シフトイン(SO/SI)
 文字式からの削除 349
 出力形式
 NLS 47
 国際化対応の日付/日時出力形式 47
 国際化対応の日付に使用する言語 547
 変数との関連付け 590
 出力処理 17
 照合順序 14
 概要 14
 言語固有 559
 言語照合 16
 異なる順序の結果 15
 バイナリ照合 14
 シンガポール
 通貨データ 182, 219
 スイス
 通貨データ 156, 193
 スウェーデン
 通貨データ 181, 218
 数値 6
 切り捨て 350
 数値データ
 円 270
 台湾の日付値 100
 日本の日付 102
 数値の切り捨て 350
 数値表記
 最善、ロケールに基づく 103
 スペイン
 通貨データ 162, 199
 スロベニア
 通貨データ 162, 199
 セグメント

長いマクロ変数値の保存 539

た

タイ

通貨データ 183, 220

タイムゾーン 25

台湾

通貨データ 185, 222

台湾の日付 100, 419

単語

検索、文字列内の位置を指定 536

地域化 4

地域コード

現在の 2 文字のコード 319

チェコ共和国

通貨データ 158, 195

中国

通貨データ 157, 194

通貨 6

円 270

通貨データ

アイルランド 162, 199

アラブ首長国連邦 151, 188

イスラエル 168, 205

イタリア 162, 199

インド 169, 206

インドネシア 167, 204

英国 163

エジプト 161, 198

エストニア 160, 197

オーストラリア 152, 189

オーストリア 199

オランダ 162, 199

カナダ 155, 192

韓国 171, 208

ギリシャ 162, 199

グリーンランド 159, 196, 200

クロアチア 165, 202

シンガポール 182, 219

スイス 156, 193

スウェーデン 181, 218

スペイン 162, 199

スロベニア 162, 199

タイ 183, 220

台湾 185, 222

チェコ共和国 158, 195

中国 157, 194

デンマーク 159, 196

ドイツ 162, 199

トルコ 184, 221

日本 170, 207

ニュージーランド 178, 215

ノルウェー 177, 214

ハンガリー 166, 203

フィンランド 162, 199

プエルトリコ 186, 223

フェロー諸島 159, 196

ブラジル 154, 191

フランス 162, 199

ブルガリア 153, 190

米国 186, 223

ベルギー 162, 199

ポーランド 179, 216

ポルトガル 162, 199

香港 164, 201

マカオ 174, 211

マルタ 162, 199

マレーシア 176, 213

南アフリカ 187, 224

メキシコ 175, 212

ラトビア 173, 210

リトアニア 172, 209

リヒテンシュタイン 156, 193

ルクセンブルク 162, 199

ロシア 180, 217

データセット

エンコーディング 16

エンコーディングサポート、リリース別 17

混在したエンコーディング 43

特定のエンコーディングの使用 44

トランスコーディングの抑制 43

変数のトランスコーディング属性 397

データセットオプション

トランスコーディング 20

デンマーク

通貨データ 159, 196

ドイツ

通貨データ 162, 199

トランスコーディング 10, 19

EBCDIC と ASCII 間のデータ転送 22

SAS オプション 20

エンコーディング値 613

改行文字 22

回避 22

互換性があるエンコーディングと互換性がないエンコーディング 21

指定した引数に有効 400

指定した変換テーブル 392

指定した文字変数で有効 399

対エンコーディング 10

必要となる理由 20

抑制 43

留意点 21

トランスコーディングエラー 559

トランスコーディング属性

データセット変数 397

トランスポートファイル

変換テーブルの適用 596

- トルコ
 - 通貨データ 184, 221

- な**
- 長いマクロ変数
 - 値の分割保存 539
- 長さ
 - 引数 330
 - 変数との関連付け 590
- 並べ替えキー
 - 言語 372
- 日時値
 - B8601DT 出力形式, ISO 8601 基本表記, タイムゾーンなし 82, 83
 - B8601TX 出力形式, ISO 8601 基本表記, タイムゾーンオフセット 85
 - E8601DT 出力形式, ISO 8601 の基本表記, タイムゾーンなし 91, 92
 - E8601TX 出力形式, ISO 8601 基本表記, タイムゾーンオフセット 94
 - 午前または午後の使用 127
 - 時刻 136
 - 指定したロケールへの変換 363, 366
 - 月名 134
 - 月の名前と月の日 130
 - 月の名前、月の日、年 128
 - 年 148
 - 年と四半期 144
 - 年と週の名前 149
 - 年と月の名前 141
 - 日時 126
 - 曜日 139
 - 曜日と日時 138
- 日時形式ディスクリプタ 363
- 日本
 - 通貨データ 170, 207
- 日本の日付 102, 421
- ニュージーランド
 - 通貨データ 178, 215
- 入力形式
 - 国際化対応の日付に使用する言語 547
 - 変数との関連付け 590
- 入力処理 17
- ノルウェー
 - 通貨データ 177, 214

- は**
- バイトオーダーマーク(BOM)接頭辞
 - Unicode 外部ファイル 546
- バイナリ照合 14
- バイナリ整数値、読み込み 501, 502, 503, 505
- バリエーション文字 14
- ハンガリー
 - 通貨データ 166, 203
- 東アジア言語
 - エンコーディング 11
- 日付 5
 - 日付値 105
- 日付値
 - 指定したロケールへの変換 360
 - 台湾語 100, 419
 - 月名 112
 - 月の名前と月の日 108
 - 年 124
 - 年と四半期 120
 - 年と週 125
 - 年と月の名前 116
 - 日本語 102, 421
 - 日付 105
 - 日付と曜日 114
 - 曜日 115
- 日付形式ディスクリプタ 360
- 標準組織 13
- ファイル参照名
 - 8文字の制約 539
- ファイルのアップロード
 - 変換テーブル 594
- フィンランド
 - 通貨データ 162, 199
- プエルトリコ
 - 通貨データ 186, 223
- フェロー諸島
 - 通貨データ 159, 196
- 不正なデータ 559
- 部分文字列
 - 引数からの抽出 344
 - 引数からの抽出、バイト位置に基づく 346
 - 文字列 538
- ブラジル
 - 通貨データ 154, 191
- フランス
 - 通貨データ 162, 199
- ブルガリア
 - 通貨データ 153, 190
- 米国
 - 通貨データ 186, 223
- ヘブライ語文字 88, 89
- ベルギー
 - 通貨データ 162, 199
- 変換テーブル
 - 指定したテーブルを使用したトランスコーディング 392
 - トランスポートファイルへの適用 596

- 変数
 - 指定した文字変数で有効なトランスコーディング 399
 - 出力形式の関連付け 590
 - トランスコーディング属性 397
 - 長さ、関連付け 590
 - 入力形式の関連付け 590
 - ラベル 590
- 変数名
 - 有効な命名規則 564
- ポーランド
 - 通貨データ 179, 216
- ポルトガル
 - 通貨データ 162, 199
- 香港
 - 通貨データ 164, 201

- ま
- マカオ
 - 通貨データ 174, 211
- マクロ変数
 - 長い値の分割保存 539
- 末尾の空白
 - 削除 36
- 末尾の空白の削除 36
- マルタ
 - 通貨データ 162, 199
- マレーシア
 - 通貨データ 176, 213
- 右揃え
 - 文字式 339
- 南アフリカ
 - 通貨データ 187, 224
- メキシコ
 - 通貨データ 175, 212
- 文字
 - 位置検索 534
- 文字式
 - 2 バイト文字の数 322
 - 圧縮 321
 - 逆順 338
 - 検証 358
 - 更新 352
 - 更新、バイト単位に基づく 354
 - 最初の固有文字の位置 358
 - 削除 349
 - 指定した単語の選択 340
 - 特定の文字の検索 326
 - 特定の文字の置換 348
 - 比較 319
 - 左詰め 328
 - 変換 348
 - 末尾の半角/全角の空白の削除 349
 - 右揃え 339
 - 文字値の内容の削除 352
- 文字値の内容の削除、バイト単位に基づく 354
- 文字値の内容の挿入 352
- 文字値の内容の挿入、バイト単位に基づく 354
- 文字値の内容の置換 352
- 文字値の内容の置換、バイト単位に基づく 354
- 文字列の検索 324
- 連結 341
- 文字式の左詰め 328
- 文字式の変換 348
- 文字セット
 - 指定、出力の META 宣言 570
 - 定義 9
- 文字セットエンコーディング
 - SAS セッション 549
- 文字データ
 - EBCDIC への変換 90
- 文字変
 - 指定した変数で有効なトランスコーディング 399
- 文字列 538
 - 位置指定して単語を検索 536
 - 先頭文字の検索 534
 - 長さ 535
 - 長さを返す 536
 - 文字列の部分文字列 538
- 文字列関数
 - 国際化の互換性 279

- や
- ユーロ
 - 出力形式 95, 98
- 用紙サイズと測定単位 6

- ら
- ラトビア
 - 通貨データ 173, 210
- ラベル, 変数との関連付け 590
- リトアニア
 - 通貨データ 172, 209
- リヒテンシュタイン
 - 通貨データ 156, 193
- リモートアプリケーション
 - 不正なデータ 559
- ルクセンブルク
 - 通貨データ 162, 199
- 連結
 - 文字式 341
- ロケール 5
 - POSIX 値 317
 - SAS 起動時の指定 6
 - SAS セッション 550

SAS セッション中に指定 7
言語切り替え 7
最適な数値表記 103
指定 6
指定したロケールへの時間または日
時値の変換 366

指定したロケールへの日時値の変換
363
指定したロケールへの日付値の変換
360
ロシア
通貨データ 180, 217



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

 support.sas.com/bookstore
for additional books and resources.


THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

