

Natural for Mainframes

Unicode およびコードページのサポート

バージョン 4.2.5

October 2009

This document applies to Natural バージョン 4.2.5 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © Software AG 1979-2009. All rights reserved.

The name Software AG™, webMethods™, Adabas™, Natural™, ApplinX™, EntireX™ and/or all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

目次








1 Unicode およびコードページのサポート	1
2 はじめに	3
コードページおよび Unicode について	4
Natural での Unicode およびコードページのサポートについて	5
メインフレームプラットフォームでの ICU	6
3 Natural プログラミング言語での Unicode およびコードページのサポート	7
Unicode ベースのデータ用の Natural データフォーマット U	8
ステートメント	9
論理条件基準	14
システム変数	14
ラージ変数およびダイナミック変数	15
セッションパラメータ	15
サンプルプログラム	18
4 Unicode/コードページ環境の設定と管理	19
ICU ライブラリ	20
メインフレームプラットフォーム用の ICU データライブラリのカスタマイズ	21
プロファイルパラメータ	22
エンコード情報	24
エンコード情報を持つ Natural オブジェクトの展開	25
5 開発環境	27
開発環境	28
環境のカスタマイズ	29
エディタ	30
メインフレームでのエディタ、システムコマンド、およびユーティリティのコードページのサポート	31
6 Natural アプリケーションの入力/出力処理	35
Unicode データの表示および入力	36
Natural Web I/O インターフェイスクライアント	37
7 Unicode データストレージ	41
Unicode データ/パラメータアクセス	42
データベース管理システムインターフェイス	42
Windows、UNIX、および OpenVMS プラットフォームでのワークファイルおよび出力ファイル	43
メインフレームプラットフォーム上のワークファイルと出力ファイル	47
8 プラットフォームの相違	49
Windows、UNIX、および OpenVMS プラットフォーム	50
メインフレームプラットフォーム	51
9 既存アプリケーションの移行	61
既存アプリケーションへの Unicode の影響	62
Windows、UNIX、および OpenVMS プラットフォームでの既存オブジェクトの移行	62
メインフレームプラットフォームでの既存オブジェクトの移行	63






既存アプリケーションへの Unicode サポートの追加	65
Natural リモートプロシージャコール (RPC) の移行	66
10 特別な考慮事項と制限事項	67
Windows、UNIX、および OpenVMS プラットフォーム	68
メインフレームプラットフォーム	68
11 双方向言語サポート	69
12 ダブルバイト文字サポート	75
13 よくある質問	77
起動エラー「Invalid code page specified」が表示されるのはなぜですか。	78
"デフォルトコードページ" とは何ですか。	78
どのデフォルトコードページが使用されていますか。	78
すべての Natural ソースを UTF-8 フォーマットで保存する必要がありますか。	78
どうすれば Natural コードで UTF-8 エンコードを処理できますか。	79
一部の文字が正しく表示されないのはなぜですか。	79
Natural ソースを編集するときにエラーが発生するのはなぜですか。	79
Natural ソースを保存するときにエラーが発生するのはなぜですか。	79
Natural ソースのエンコードはどうすればわかりますか。	80
Natural ソースのエンコードはどうすれば変更できますか。	80
どうすれば既存の Natural ソースを UTF-8 フォーマットに変換できますか （Windows、UNIX、および OpenVMS のみ）。	80
文字を変換できない場合、どの置換文字が使用されますか。	81
以前の Natural バージョンで Natural 4.2 ソースを使用できますか。	81
以前の Natural バージョンで UTF-8 ソースを使用できますか。	81
UTF-8 フォーマットのソースをカタログするときに変換エラーが発生するのは なぜですか。	81
端末エミュレーションを経由して U フォーマットを表示するときに、UNIX ま たは OpenVMS でガーベッジが表示されるのはなぜですか。	82
現在の SPoD クライアントと古い SPoD サーバーを使用できますか。	82
現在の SPoD サーバーと古い SPoD クライアントを使用できますか。	82
索引	83

1 Unicode およびコードページのサポート

このドキュメントでは、Windows、UNIX、OpenVMS、およびメインフレームプラットフォームでの、Natural による Unicode およびコードページのサポートについて説明します。また、Natural による双方向言語およびダブルバイト文字のサポートについても説明します。

このドキュメントは次の項目で構成されています。

 はじめに	コードページおよび Unicode 標準に関する全般的な情報、および Natural での Unicode およびコードページのサポートに関する全般的な情報
 Natural プログラミング言語での Unicode およびコードページのサポート	Uフォーマットに関する情報、および Unicode およびコードページのサポートを提供するステートメント、論理条件基準、システム変数、ラージ変数およびダイナミック変数、およびセッションパラメータに関する情報
 Unicode/コードページ環境の設定と管理	ICU ライブラリに関する情報、Unicode およびコードページのサポートを提供するプロファイルパラメータに関する情報、およびコードページデータのエンコードに関する情報
 開発環境	環境をカスタマイズする方法および Natural エディタによる Unicode の処理メインフレームでの Natural ソースのコードページのサポートに関する情報（エディタ、システムコマンド、およびユーティリティ）
 Natural アプリケーションの入力/出力処理	Unicode データの表示および入力方法 SPoD およびランタイム環境で使用される Natural Web I/O インターフェイスクライアントに関する情報
 Unicode データストレージ	データベースアクセスに関する情報、および Unicode およびコードページのサポートを提供するワークファイルタイプおよび出力ファイルに関する情報
 プラットフォームの相違	Windows、UNIX、OpenVMS、およびメインフレームプラットフォームでの処理の相違。

 既存アプリケーションの移行	既存アプリケーションへの Unicode の影響。既存オブジェクトの移行、既存アプリケーションへの Unicode サポートの追加、および Natural リモートプロシージャコール (RPC) の移行方法。
 特別な考慮事項と制限事項	さまざまなプラットフォームでの重要な情報および制限
 双方向言語サポート	Natural による双方向言語のサポート方法
 ダブルバイト文字サポート	Natural によるダブルバイト文字セットのサポート方法
 よくある質問	よくある質問への回答

2 はじめに

- コードページおよび Unicode について 4
- Natural での Unicode およびコードページのサポートについて 5
- メインフレームプラットフォームでの ICU 6

コードページおよび Unicode について

従来のコードページは、特定の言語または共通スクリプトを共有する言語グループをサポートする、特定の順序で並べられた、選択された文字コードのリストです。コードページには、最大で256個の文字コードを含めることができます。中国語や日本語など、256文字よりも多い文字セットの場合は、ダブルバイトコード単位処理（DBCS）が使用されます。DBCS コードページは実際にはマルチバイトエンコードであり、1バイトおよび2バイトコードポイントが混在します。

コードページには、同じデータ文字列内に異なる言語を保存できないという固有の短所があります。Unicode は、この制限をなくすために設計されました。データへのアクセスに使用されるプラットフォーム、プログラム、または言語に依存しない標準エンコードが、すべての文字セットに提供されます。Unicode では、すべての文字に一意的な番号が付けられます。

Unicode 標準によって定義された各コード要素に1つの番号が割り当てられます。これらの番号それぞれは「コードポイント」と呼ばれ、テキスト内で参照される場合は、「U」の後に16進形式でリストされます。例えば、コードポイント "U+0041" は、16進数 "0041"（10進数 "65" と同じ）です。これは、Unicode 標準の文字 "A" を表し、「LATIN CAPITAL LETTER A」という名前です。

Unicode 標準では、同じデータをバイト、ワード、またはダブルワード指向フォーマットで転送することを可能にする3つのエンコード形式が定義されています。「コード単位」は、特定のエンコードで文字を表すことができる最小のビット組み合わせです。Unicode 標準では、UTF-8エンコード形式で8ビットのコード単位、UTF-16エンコード形式で16ビットのコード単位、UTF-32エンコード形式で32ビットのコード単位が使用されます。3つのエンコード形式すべてで同じ共通文字レパートリがエンコードされるため、データを欠落させずに相互に効率的に変換できます。

Natural のコンテキストでは、これらのエンコード形式のうちの UTF-16 および UTF-8 の2つが関係します。Natural では、ランタイム時の Unicode 文字列のコーディングに UTF-16 が使用され、ファイル内の Unicode データのコーディングに UTF-8 が使用されます。UTF-16 は、エンディアンに依存する2バイトエンコードです。使用されるエンディアンフォーマットは、プラットフォームによって異なります。UTF-8 は1バイトエンコードです。

Unicode の詳細については、Unicode Consortium の Web サイト <http://www.unicode.org/> を参照してください。



注意: Unicode コードポイントに関する情報を取得するには、Natural for Windows で使用可能な SYSCP ユーティリティを使用します。

Natural での Unicode およびコードページのサポートについて

以前のバージョンの Natural では、コードページの文字のみがサポートされていました。Natural バージョン 4.2 (メインフレーム)、6.2 (Windows および UNIX)、および 6.3 (OpenVMS) から、Unicode がサポートされています。

Unicode のサポートに関して、Natural データフォーマット U が導入されており、固有のステートメント、パラメータ、システム変数などがあります。詳細については、このドキュメントの該当する箇所を参照してください。

現在、既存のデータのほとんどをコードページフォーマットで使用できます。このデータを Unicode に変換する場合は、正しいコードページを使用する必要があります。Natural では、次の複数のレベルで正しいコードページを定義できます。

- Natural でデフォルトコードページが定義されていない場合は、システムコードページが使用されます。

メインフレームでコードページが定義されていない場合 (CP=OFF)、デフォルトコードページは定義されていません。現在の I/O デバイスのコードページに合わせて Natural セッションを調整するために、CP=AUTO が使用されます。

- Natural パラメータ CP が定義されている場合は、デフォルトコードページが使用されます。オペレーティングシステムのコードページは上書きされます。
- ソースなどに対して定義されているオブジェクトコードページによって、このオブジェクトのデフォルトコードページは上書きされます。

1つのアプリケーションで Unicode 文字列とコードページ文字列を使用する場合は、Natural によって、必要に応じて (例えば、データを移動または比較する場合に) 暗黙的に変換されます。明示的な変換は、ステートメント MOVE ENCODED を使用して実行できます。

ほとんどの場合、Unicode のサポートを必要としない既存アプリケーションは、変更なく実行されます。Windows、UNIX、および OpenVMS プラットフォームでは、既存のソースが異なるコードページでエンコードされる場合に、変更が必要な場合があります。詳細については、このドキュメントの「[既存アプリケーションの移行](#)」を参照してください。

既存アプリケーションを実行し、そのアプリケーションを変更しないで Unicode データもサポートすることはできません。Natural データフォーマット U をアプリケーションに導入する必要があります。ほとんどの場合、A フォーマット定義を U フォーマット定義で置き換えるだけでは不十分です。文字列の特定のメモリレイアウトを前提とするすべてのコード (例えば、英数字から数字フォーマットへの REDEFINE) を変更する必要があります。

Unicode 文字は、変数名、オブジェクト名、およびライブラリ名で使用することはできません。

Unicode ベースのデータは、Adabas および DB2 でサポートされています。

Natural では、Unicode の照合および変換について、International Components for Unicode (ICU) ライブラリが使用されます。詳細については、<http://icu.sourceforge.net/userguide/> を参照してください。このドキュメントの「[ICU ライブラリ](#)」も参照してください。

メインフレームプラットフォームでの ICU

ICU は IBM によって配布されています。これは、コードページおよび Unicode のサポート、ソフトウェアの国際化対応 (I18N)、およびグローバリゼーション (G11N) のための、十分に考慮された移植可能な C/C++ ライブラリのセットです。すべてのプラットフォームで、アプリケーションに同じ結果をもたらします。ICU は C/C++ で記述されているため、実行可能な ICU モジュールのサイズは非常に大きく、アセンブラで記述されたコードに比べてパフォーマンスはよくありません。一方、最初から作り直すのは無意味です。そのような複雑な問題については、承認済みのツールを使用するほうが意味があるでしょう。他の多くのソフトウェアベンダも ICU ライブラリを使用しているため、コードページおよび Unicode のサポートに関する Natural の動作がそれらの製品と互換性があることを保証できます。ICU は、新しい Unicode バージョンのサポートや追加のロケールのサポートのために、IBM によって今後も改善される予定です。

現在使用中の ICU バージョンと Unicode 仕様に関する情報は、SYSCP ユーティリティのメインメニューで入手できます。Natural for Mainframes の『ユーティリティ』ドキュメントの「SYSCP の起動と終了」を参照してください。

3 Natural プログラミング言語での Unicode およびコードページのサポート

- Unicode ベースのデータ用の Natural データフォーマット U 8
- ステートメント 9
- 論理条件基準 14
- システム変数 14
- ラージ変数およびダイナミック変数 15
- セッションパラメータ 15
- サンプルプログラム 18

Unicode ベースのデータ用の Natural データフォーマット U

Natural では、フォーマット U および U 定数を使用して Unicode 文字列を指定できます。

■ フォーマット U

フォーマット U を使用して、Unicode 文字列を保持するデータを定義できます。Natural データフォーマット U は、内部的には UTF-16 です。

『プログラミングガイド』の「ユーザー定義変数のフォーマットおよび長さ」も参照してください。

■ U 定数

接頭辞 "U" を使用して、Unicode 定数を定義できます。次に例を示します。

```
U'Äpfel'
```

接頭辞 "UH" は、Unicode 定数を 16 進形式で定義するために使用できます。Unicode 標準で定義されているように、4桁の16進数は1つのUTF-16コード単位を表します。したがって、全体の長さは4の倍数になります。例えば、次の16進形式が必要であるとします。

```
U'Äpfel'
```

"Ä"、"p"、"f"、"e"、および "l" の UTF-16 コード単位

("U+00C4"、"U+0070"、"U+0066"、"U+0065"、および "U+006C") が必要であり、これらを次の16進数の文字列に結合する必要があります。

```
UH'00C4007000660065006C'
```

『プログラミングガイド』の「Unicode 定数」も参照してください。

データフォーマット U は、エンディアンに依存します。フォーマット B と U 間で移動する場合は、これを考慮する必要があります。

U と A の比較

A フォーマットと比較した U フォーマットの利点は、さまざまな言語の文字の任意の組み合わせを保持でき、デフォルトコードページ（システム変数 *CODEPAGE の値）に依存しない点です。さらに、U フォーマットでは、異なるプラットフォーム間でのデータの共有が簡単です。変換（例えば、EBCDIC から ASCII へ）は必要ありません。

一方、U フォーマットデータは、A フォーマットデータよりも多くのメモリが消費されます。ほとんどの文字列をシングルバイトエンコードで表すことができる言語の場合、U フォーマットでは、以前に必要なだったスペースの2倍のスペースが文字列によって占められるようになります。ただし、東アジア言語の場合は、通常、メモリ消費は増加しません。

ステートメント

基本的に、A フォーマットを使用できるステートメントのほとんどで、U フォーマットを使用できます。ただし、ステートメントのオペランドとして Natural オブジェクト名が与えられる場合（CALLNAT ステートメントなど）、Natural オブジェクト名は A フォーマットであるため、U を使用できません。特定のステートメントの詳細については、『ステートメント』ドキュメントを参照してください。

基本的に、A および U フォーマットは、1つのステートメント内で一緒に使用できます。次に例を示します。

```
EXAMINE S FOR P WITH DELIMITER D REPLACE R
```

S は U フォーマット、P、D、および R は A フォーマットです。

上記の例では、変数 P、D、および R がターゲットフォーマット U に一時的に変換されてから、EXAMINE ステートメントが実際に実行されます。Unicode からコードページ、またはその逆の変換では、ICU 機能呼び出す必要があります。この変換には、コンピュータの時間およびメモリが追加で必要になります。この短所は、大きな変数の場合にさらに顕著になります。頻繁な変換を避けるために、1つのステートメント内では1つのフォーマットのみを使用することをお勧めします。上記の例のすべてのオペランドが U フォーマットまたは A フォーマットで指定されている場合は、変換は必要ありません。ただし、U オペランドのみを指定することを選択した場合は、遅くなります。このオペランドタイプでは、その性質によってリソースが多く消費されるためです。この場合、1文字が、A フォーマットで使用される 1 バイトではなく、2 バイトでコード化されます。

変換（特に、U フォーマットから A フォーマットへ）では、ターゲットコードページで文字を表現できないというリスクが常にあります。例えば、Unicode 文字 "U+05D0"（ヘブライ文字のアレフ）をコードページ IBM01140（英語）に変換するとします。この文字はコードページ IBM01140 に含まれていないため、このコードページの置換文字か、またはコードページの定義時に NATCONFIG に指定されたプレースホルダ（メインフレームのみ）が使用されます。パラメータ CPCVERR が "ON" に設定されている場合は、変換エラーを示すエラーメッセージがこの場合に表示されます。いずれの場合にも、元の情報は失われます。

Unicode を使用する場合、特に次のステートメントに影響します。

- MOVE NORMALIZED
- MOVE ENCODED
- EXAMINE
- PARSE XML
- REQUEST DOCUMENT
- DEFINE PRINTER

- CALLNAT (RPC)


MOVE NORMALIZED

Unicode での正規化：バイナリでの等価比較が可能な形式へデータを変換するために、同等のシーケンスの代替表現をテキストデータから削除するプロセス。Unicode 標準では、さまざまな正規化形式が定義されています。Unicode 文字列の正規分解の結果の正規化形式の後に、可能であればプライマリ合成による分解されたすべてのシーケンスに置き換わるものを付けた形式は、「合成済み正規化形式」（NFC）と呼ばれます。

Natural では、Unicode 文字を部分的に切り捨てることなく文字列処理を実行できるように、すべての Unicode データは NFC フォーマットであることが前提となっています。Natural 変換処理では、結果として生じる Unicode 文字列は NFC であることが保証されます。Natural の外部から Unicode データを受け取り、そのデータが NFC フォーマットであることが保証されない場合は、MOVE NORMALIZED ステートメントを適用できます。

例：

文字シーケンス	NFC
ê (U+00EA)	ê (U+00EA)
e (U+0065) + ^ (U+0302)	ê (U+00EA)

 **注意:** NFC フォーマットの 2 つ以上の文字列を連結した場合、NFC フォーマットにならない場合があります。


MOVE ENCODED

MOVE ステートメントを使用して文字列を U から A へ、またはその逆に移動するときに、Unicode とデフォルトコードページ（システム変数 *CODEPAGE の値）の間の暗黙的な変換が実行されます。

さらに、異なるコードページ間の変換、または使用可能なコードページから Unicode への変換、およびその逆への変換に、MOVE ENCODED ステートメントを使用できます。これは、Natural の外部からのデータがデフォルトコードページとは異なるコードページでコード化されている場合に役立ちます。しかし、デフォルトコードページと Unicode 間の変換についても、GIVING 節での潜在的な変換エラーを取得する場合は、このステートメントを使用できます。この場合は、CPCVERR が "ON" に設定されている場合に、MOVE ステートメントはランタイムエラーで停止します。

文字を変換できない場合に、この文字に対して置換文字が使用されるか変換が失敗するかは、CPCVERR パラメータの設定によって異なります。Windows、UNIX、および OpenVMS プラットフォームでは、Unicode からデフォルトコードページ (CP) への変換について ICU によって定義されたデフォルトの置換文字は、プロファイルパラメータ SUBCHAR を使用して変更できます。

このステートメントは、U データから UTF-8 フォーマットへの変換についても使用できます。


 **注意:** デフォルトコードページとは異なるコードページにデータを変換する場合は、このデータを I/O で使用しないことをお勧めします。I/O は、デフォルトコードページでのみ意味があります。

EXAMINE

「grapheme」は、ユーザーが一般に文字と見なすものです。ほとんどの場合、Unicode コードポイントは書記素ですが、書記素が複数の Unicode コードポイントで構成される場合もあります。例えば、1 つの基底文字と 1 つ以上の結合文字のシーケンスは書記素です。

例: "a" (U+0061) + "." (U+0323) + "^" (U+0302) は、次のように表示される 1 つの書記素を定義します。

â

 **注意:** 基底文字／結合文字のシーケンスが正規化される場合、このことはシーケンスが常に合成済み文字によって置き換えられることを意味しません。すべての文字に合成済みフォーマットがあるわけではないためです。

「補助コードポイント」は、"U+10000" と "U+10FFFF" の間の Unicode コードポイントです。補助コードポイントは UTF-16 であり、2 つのコード単位で構成されるサロゲートペアによって表されます。ペアの最初の値は「上位サロゲートコード単位」、2 番目は「下位サロゲートコード単位」です。このような文字は一般的にはまれですが、中国語や日本語の人名の一部などで使用される場合があります。したがって、東アジアの国の行政アプリケーションでは、一般にこれらの文字のサポートが必要です。

EXAMINE とその SUBSTRING オプションなどの文字列操作ステートメントは、UTF-16 コード単位を処理します。コードによって書記素またはサロゲートペアが分離されないようにすることは、ユーザーの責任です。

ただし、EXAMINE ステートメントの節 CHARPOSITION および CHARLENGTH (「構文 3 - Unicode 書記素用の EXAMINE」を参照) を使用して、書記素の UTF-16 コード単位での開始位置および長さを求めることができます。結果の値は、SUBSTRING 呼び出しに使用できます。これらの節を使用して、書記素ごとに文字列をスキャンできます。

例：

```

DEFINE DATA LOCAL
1 #UNICODE-STRING      (U15)
1 #CODE-UNIT-INDEX    (N4)
1 #CODE-UNIT-LEN      (N4)
1 #GRAPHEME-NUMBER    (N4)
END-DEFINE

MOVE U'aṽcṽbṽcṽd' TO #UNICODE-STRING

#GRAPHEME-NUMBER := 1

REPEAT
EXAMINE
    FULL VALUE OF #UNICODE-STRING
    FOR CHARPOSITION #GRAPHEME-NUMBER
    GIVING POSITION IN #CODE-UNIT-INDEX
    GIVING LENGTH IN #CODE-UNIT-LEN

    DISPLAY #UNICODE-STRING #GRAPHEME-NUMBER #CODE-UNIT-INDEX #CODE-UNIT-LEN

    #GRAPHEME-NUMBER := #GRAPHEME-NUMBER + 1
WHILE #CODE-UNIT-INDEX NE 0
END-REPEAT

END
    
```

上記のプログラム例の出力は、次のとおりです。

```

Page      1                                05-12-15  09:33:49

#UNICODE-STRING #GRAPHEME-NUMBER #CODE-UNIT-INDEX #CODE-UNIT-LEN
-----
aṽcṽbṽcṽd           1                1                1
aṽcṽbṽcṽd           2                2                2
aṽcṽbṽcṽd           3                4                1
aṽcṽbṽcṽd           4                5                3
aṽcṽbṽcṽd           5                8                1
aṽcṽbṽcṽd           6                9                3
aṽcṽbṽcṽd           7               12               1
aṽcṽbṽcṽd           8               13               3
aṽcṽbṽcṽd           9                0                0
    
```


PARSE XML

XML 文書では、文書のエンコードに関する情報を XML 文書ヘッダーに含めることができます（例えば、`<?xml version="1.0" encoding="UTF-8" ?>`）。XML 文書にこの情報が含まれている場合、Windows、UNIX、および OpenVMS プラットフォームでの XML 文書の解析には、受け取りフィールドがフォーマット U ではない場合は、XML 文書ヘッダー内で指定されたコードページから Natural のデフォルトコードページ（システム変数 *CODEPAGE の値）への変換が常に含まれます。

メインフレームプラットフォームでは、文書がすでに UTF-16 でエンコードされていない場合、解析される文書は常に内部的に UTF-16 に変換されます。

詳細については、PARSE XML ステートメントの説明を参照してください。

『プログラミングガイド』の「インターネットおよびXML アクセス用のステートメント」も参照してください。

REQUEST DOCUMENT

REQUEST DOCUMENT ステートメントを使用したデータ転送では、通常、コードページ変換は呼び出されません。送信データ/受信データを特定のコードページでエンコードする場合は、REQUEST DOCUMENT ステートメントの DATA ALL 節/RETURN PAGE 節を使用して指定できます。

詳細については、REQUEST DOCUMENT ステートメントの説明を参照してください。

『プログラミングガイド』の「インターネットおよびXML アクセス用のステートメント」も参照してください。

DEFINE PRINTER

メインフレームプラットフォームでは、DEFINE PRINTER ステートメントに CODEPAGE 節があり、出力レポートデータをデフォルトコードページ（システム変数 *CODEPAGE の値）とは異なるコードページに変換できます。Windows、UNIX、および OpenVMS プラットフォームでは、DEFINE PRINTER ステートメントにそのような節はありません。CODEPAGE 節が定義された場合、Windows、UNIX、および OpenVMS プラットフォームでは無視されます。

CALLNAT (RPC)

RPC 経由での Unicode フォーマットでのデータ交換がサポートされています。CALLNAT ステートメントの説明を参照してください。

U データがビッグエンディアンエンコードのプラットフォームからリトルエンディアンエンコードのプラットフォームへ、またはその逆に送信される場合、エンコードは受信プラットフォームのエンコードに準拠するように変更されます。例えば、リトルエンディアンエンコードの U データがビッグエンディアンのプラットフォームに到着した場合、このデータはビッグエンディアン


エンコードに変換されてからプログラムに渡されます。このデータが送り返されるときは、リトルエンディアンエンコードに戻されます。

論理条件基準

論理条件基準では、Unicode のオペランドを英数字およびバイナリのオペランドとともに使用できます。すべてのオペランドが Unicode のオペランド（フォーマット U）というわけではない場合は、2 番目以降のオペランドはすべて最初のオペランドのフォーマットに変換されます。バイナリのオペランド（フォーマット B）が 2 番目以降のオペランドとして指定された場合、バイナリのオペランドの長さは偶数である必要があります。バイナリのオペランドには Unicode コードポイントが含まれていると想定されます。

最初のオペランドが Unicode のオペランド（フォーマット U）であるために、比較が Unicode 比較として実行される場合、ICU 照合アルゴリズムが使用されます。ICU アルゴリズムでは、単純なバイナリ比較は実行されません。そのため、例えば次のような結果になります。

- "U+0000" などの一部のコードポイントは、比較プロセスで無視されます。
- 組み合わせられた文字は、等価の 1 つのコードポイントと等しいとみなされます。例えば、"U+00E4" によって表されるドイツ語の文字 "ä" と、コードポイント "U+0061" および "U+0308" の組み合わせは、ICU によって等しいとみなされます。

 **注意:** 英数字と Unicode のオペランドの比較は、フィールドのシーケンスに応じて、異なる結果となる場合があります。

『プログラミングガイド』の「論理条件基準」も参照してください。

システム変数

このセクションでは、次のトピックについて説明します。

- *CODEPAGE

■ *LOCALE

*CODEPAGE

システム変数 *CODEPAGE は、デフォルトコードページ（Unicode とコードページフォーマット間の変換に使用されているコードページ）の IANA 名を返すために使用されます。

*LOCALE

システム変数 *LOCALE には、現在のロケールの言語および国が含まれています。

ラージ変数およびダイナミック変数

U フォーマットは、ラージ変数およびダイナミック変数に使用できます。ダイナミック U 変数の場合、*LENGTH によって UTF-16 コード単位の数が返されます。

『プログラミングガイド』の「ダイナミック変数およびフィールドについて」も参照してください。

セッションパラメータ

次のセッションパラメータを使用できます。

パラメータ	説明
DL	フォーマット A または U のフィールドの表示長を指定します。『プログラミングガイド』の「出力の表示長-DL パラメータ」も参照してください。
EMU*	Unicode での編集マスク
ICU*	Unicode での挿入文字
LCU*	Unicode での先頭文字
TCU*	Unicode での末尾文字

上記の表でアスタリスク (*) マークが付いたセッションパラメータは、Windows、UNIX、および OpenVMS プラットフォームでのみ使用できます。

DL と AL の比較

Natural が Unicode 対応ではなかった間は、英数字フィールドの長さは、フィールドを表示するために必要な列の数（表示列の数）と常に同じでした。このことは、DBCS コードページを使用する東アジア言語についても該当しました。A フォーマットフィールドは、半分の文字のみを保持できます。例えば、A10 は A5 になります。

例：

```
DEFINE DATA LOCAL
1 #A8 (A8)
END-DEFINE
#A8 := 'computer'
WRITE #A8
#A8 := '電腦系統'
WRITE #A8
END
```

上のコードの出力結果は、次のとおりです。

```
Page      1 ...
computer
電腦系統
```

Uフォーマットフィールドでは、フィールドの長さ并表示列の数は同じではなくなります。Uの文字は、狭い幅（ラテン文字など）か、広い幅（中国語文字など）か、または幅を持たない（結合文字など）場合があります。したがって、Uフィールドが必要とする表示列の数はまったくわかりません。これは、フィールドの内容によって異なります。Naturalでは、画面上に予約する必要がある列の数を自動的に決定できません。最大サイズを想定するとラテン出力で大きな差異が発生し、最小サイズを想定すると中国語出力が完全には表示されない場合があります。したがって、Naturalのプログラマは、フィールドの表示幅を定義する必要があります。これは、DLパラメータを使用して行います。ALパラメータは、この目的には使用できません。このパラメータでは、フィールドの定義された長さを超える部分が切り取られるためです。Uフィールドから文字を切り取るのではなく、次のフィールドの開始位置のみを定義する必要があります。

例：

```
DEFINE DATA LOCAL
1 #U8 (U8)
1 #U4 (U4)
END-DEFINE
#U8 := 'computer'
WRITE #U8
#U4 := U'電腦系統'
WRITE #U4 (DL=8)
END
```

上記のコードの結果は、前と同じ出力になります。

Page 1 ...

computer
電腦系統

Windows では、出力ウィンドウを使用してローカルで、または Natural Web I/O インターフェイスクライアントを使用してリモート開発環境で、DL パラメータに定義された値がフィールドの実際の表示幅よりも小さいフィールド内をスクロールできます。

EMU、ICU、LCU、TCU と EM、IC、LC、TC の比較

パラメータ EMU、ICU、LCU、および TCU (Windows、UNIX、および OpenVMS プラットフォームでのみ使用可能) を使用すると、デフォルトコードページに含まれていない文字を使用できます。文字は、生成されたプログラムに Unicode フォーマットで保存されます。これらのパラメータは、すべてのフィールドフォーマットで使用できます。

パラメータ EM、IC、LC、および TC も、U フォーマットフィールドで使用できます。デフォルトコードページに含まれている文字が他のコードページではエンコードが異なる場合に、これらのパラメータも便利な場合があります。例えば、ユーロ記号 (€) は、"windows-1252" (Latin 1) コードページではコードポイント "0x80" ですが、"windows-1251" (キリル語) コードページではコードポイント "0x88" です。したがって、Unicode パラメータを使用することによって、PC にインストールされているコードページにかかわらず、ユーロ記号は常に正しく表示されるようになります。

EMU の例：

```
DEFINE DATA
LOCAL
  01 EMPLOYEES-VIEW VIEW OF EMPLOYEES
    02 FIRST-NAME
    02 NAME
    02 SALARY (1)
END-DEFINE
*
  READ (6) EMPLOYEES-VIEW
    DISPLAY NAME FIRST-NAME SALARY(1) (EMU=999,999€)
  END-READ
*
END
```

上のコードの出力結果は、次のとおりです。

Page	1	05-12-15	11:45:36
NAME	FIRST-NAME	ANNUAL SALARY	

ADAM	SIMONE	159,980€	
MORENO	HUMBERTO	165,810€	
BLOND	ALEXANDRE	172,000€	
MAIZIERE	ELISABETH	166,900€	
CAUDAL	ALBERT	167,350€	
VERDIE	BERNARD	170,100€	

サンプルプログラム

ライブラリ SYSEXPB に、Natural での Unicode およびコードページのサポートのサンプルプログラムがあります。

- UNICOX01 は、すべての Unicode 文字をリストします。
- UNICOX02 は、Unicode 文字をコードポイントに、またはその逆に変換します。
- CODEPX01 は、すべてのコードページ、そのコードページが Natural でサポートされているかどうか、およびそのコードページが使用するエンコードをリストします。サポートされるすべてのコードページについて、そのコードページの文字をリストするサービスおよびコードページの文字列を 16 進表示に変換したり、その逆に変換したりするサービスを提供します。
- CODEPXL1 は、任意の 1 バイトコードページのすべての文字をリストします。
- CODEPXL2 は、任意の 2 バイトコードページのすべての文字をリストします。
- CODEPXC1 は、任意のコードページの文字列を 16 進表示に、およびその逆に変換します。


4 Unicode／コードページ環境の設定と管理

■ ICU ライブラリ	20
■ メインフレームプラットフォーム用の ICU データライブラリのカスタマイズ	21
■ プロファイルパラメータ	22
■ エンコード情報	24
■ エンコード情報を持つ Natural オブジェクトの展開	25

ICU ライブラリ


Windows、UNIX、および OpenVMS プラットフォーム

ICU ライブラリは、ICU 変換および照合データのフルセットとともに常にインストールされます。コンフィグレーションファイル NATCONVINI の設定値が A フォーマットに適用されます。U フォーマットに対して、対応するチェック（文字が大文字に変換される時など）が ICU ライブラリによって行われます。

 **注意:** ICU バージョンおよびサポートされるコードページに関する情報を取得するには、Natural for Windows で使用可能な SYSCP ユーティリティを使用します。

メインフレームプラットフォーム

関連モジュールを、共有ニュークリアスに静的にリンクするか、または RCA 手法によって動的にロードできます。

 **注意:** 現在使用中の ICU バージョンと Unicode 仕様に関する情報は、SYSCP ユーティリティのメインメニューで入手できます。Natural for Mainframes の『ユーティリティ』ドキュメントの「SYSCP の起動と終了」を参照してください。

次の 3 つのロードモジュールが提供されています。

ロードモジュール	説明
NATICU	コードページおよび Unicode 変換機能および照合サービスを含みます。前者は、1 つのコードページから別のコードページまたは Unicode への変換、およびその逆の変換に必要です。後者は、ロケール ID を考慮した Unicode 文字列の文字列比較に使用されます。 サイズを小さくするために、最も一般的なコードページおよびロケールのみが含まれています。コードページはすでに NATCONFIG で宣言されています。
NATICUCV	NATICU と同じですが、照合サービスはありません。したがって、サイズは大幅に小さくなっています。
NATICUXL	NATICU と同じですが、現在サポートされている ICU バージョンによって提供される使用可能なすべてのコンバータおよびロケールを含みます。約 230 の異なるコードページ（主に EBCDIC コードページ）および 238 のロケールをサポートします。したがって、モジュールサイズは非常に大きくなっています。 NATICUXL が Natural ニュークリアスにリンクされる場合、目的のコードページがコンフィグレーションファイル NATCONFIG で宣言されている必要があります。 NATICUXL は、現在サポートされている ICU バージョン (http://demo.icu-project.org/icu-bin/convexp を参照) によってサポートされるすべてのコードページおよびロケール ID をサポートします。

ロードモジュール	説明
	注意: 技術的な制約により、z/VSE では NATICUXL は提供されていません。

「さまざまな目的のための NATICU モジュール」も参照してください。

その他のテーブル:

テーブル	説明
NATSCTU	Unicode 文字に必要なスキャナテーブル。Natural ニュークリアスによる使用のために、現在サポートされている Unicode バージョンの Unicode 文字のプロパティをマップします。このテーブルは変更できません。
NATCPTAB	オプションのシングルバイトコードページ変換アクセラレータテーブル。このテーブルが存在する場合は、ICU 機能呼び出さずにこのテーブルによって実行されるため、コードページ間の変換が速くなります。詳細については、Natural for Mainframes の『オペレーション』ドキュメントの「変換テーブル」を参照してください。

メインフレームプラットフォーム用の ICU データライブラリのカスタマイズ

ICU は、広範な種類のデータテーブルを使用して、そのサービスの多くを提供しています。例として、コンバータマッピングテーブル、照合ルール、音写ルール、改行イタレールールおよびディクショナリ、その他のロケールデータなどを挙げるすることができます。Natural 用の ICU データライブラリは、必要なデータ項目を含むパッケージとして提供されています。単一のデータ項目ファイルの代わりにパッケージを使用すると、ファイルへのアクセスは初期化中にパッケージをロードするときの 1 回のみとなるため、パフォーマンスが向上します。ただし、データ項目を追加すると、パッケージを再構築することが必要となるため、柔軟性は低くなります。

Natural バージョン 4.2.5 以降、および ICU バージョン 3.8 以降ではそれぞれ、ICU データライブラリをカスタマイズできます。カスタマイズの主要な目的は、既存または新規のコンバータマッピングテーブルを追加することですが、他のデータ項目（照合ルール、改行イタレールール、その他のロケールデータなど）を追加することも可能です。

ICU データライブラリのカスタマイズツールは、ServLine24

(<http://seroline24.softwareag.com/public/>) の Component Downloads エリアから入手可能です。ICU データライブラリのカスタマイズ方法の詳細については、ダウンロードした zip ファイルに含まれる `readme.txt` ファイルを参照してください。

プロファイルパラメータ

このセクションでは、Unicode およびコードページのサポートとともに使用されるプロファイルパラメータについて説明します。すべてのプラットフォームですべてのプロファイルパラメータが使用できるわけではありません。

- **すべてのプラットフォーム**
- Windows、UNIX、および OpenVMS プラットフォーム
- メインフレームプラットフォーム
- メインフレーム用の Natural 開発サーバー

すべてのプラットフォーム

次のプロファイルパラメータは、すべてのプラットフォームで使用できます。

パラメータ	説明
CP	<p>Natural のデフォルトコードページを定義します。このコードページは、単一のオブジェクト（Natural ソースなど）に対して定義されたコードページがない場合に、ランタイム環境および開発環境に対して使用されます。</p> <p>プラットフォームに適したコードページのみを使用できます。つまり、例えば、EBCDIC コードページは、Windows、UNIX、または OpenVMS プラットフォームには定義できません。また、ASCII コードページは、メインフレームプラットフォームには定義できません。メインフレームでは、間違ったコードページが使用された場合に、初期化エラーメッセージが表示されます。</p> <p>注意: Natural for Windows/UNIX バージョン 6.2、Natural for OpenVMS バージョン 6.3、および Natural for Mainframes バージョン 4.2 以降、Natural RPC で使用される既存の CP パラメータは CPRPC に名前が変更されています。</p>
CPCVERR	<p>Unicode からコードページへ、コードページから Unicode へ、または 1 つのコードページから別のコードページへ変換するときに発生する変換エラーが、Natural エラーになるかどうかを指定します。</p> <p>このパラメータは、Natural ソースをソースエリアにロードするとき、または Natural ソースをカタログするときの Natural ソースの変換には関連しません。</p> <p>メインフレームプラットフォームでは、端末エミュレーションの I/O の前に Unicode フィールドがコードページに変換されたとは見なされません。この場合、ICU によって定義された置換文字は、NATCONFIG で定義されたプレースホルダ文字によって置き換えられます。</p>
CPOBJIN	<p>データのバッチ入力ファイルのエンコードに使用するコードページを指定します。このファイルは、Natural プロファイルパラメータ CMOBJIN（Windows、UNIX、および OpenVMS）またはデータセット CMOBJIN（メインフレーム）で定義されています。</p>
CPPRINT	<p>バッチ出力ファイルのエンコードに使用するコードページを指定します。このファイルは、Natural プロファイルパラメータ CMPRINT（Windows、UNIX、および OpenVMS）またはデータセット CMPRINT（メインフレーム）で定義されています。</p>

パラメータ	説明
CPSYNIN	コマンドのバッチ入力ファイルのエンコードに使用するコードページを指定します。このファイルは、Natural プロファイルパラメータ CMSYNIN (Windows、UNIX、および OpenVMS) またはデータセット CMSYNIN (メインフレーム) で定義されています。
SRETAIN	すべての既存のソースを元のエンコードフォーマットで保存する必要があることを指定します。「 環境のカスタマイズ 」も参照してください。

以下の項目も参照してください。

- Natural for Windows、Natural for UNIX、および Natural for OpenVMS については、『オペレーション』ドキュメントの「バッチモードでの Natural」セクションにある「入力および出力ファイル用のコードページ」
- Natural for Mainframes については、『オペレーション』ドキュメントの「バッチモードでの Natural」
- 有効なコードページについては、<http://www.iana.org/assignments/character-sets>

Windows、UNIX、および OpenVMS プラットフォーム

次のプロファイルパラメータは、Windows、UNIX、および OpenVMS プラットフォームでのみ使用できます。

パラメータ	説明
SUTF8	Natural ソースが保存されるときに使用されるデフォルトのフォーマットを指定します。 注意: UNIX および OpenVMS では、このパラメータは SPoD 環境でのみ使用できます。
SUBCHAR	Unicode からデフォルトコードページへの変換用の置換文字を指定します。SUBCHAR が "OFF" の場合は、ICU によって定義されたデフォルトの置換文字が使用されます。 注意: SUBCHAR は、コードページから Unicode への変換、または Unicode からデフォルトコードページとは異なるコードページへの変換には影響しません。
WEBIO	入出力に (Unicode をサポートする) Natural Web I/O インターフェイスクライアントか (Unicode に対応していない) 端末エミュレーションウィンドウのどちらを使用するかを指定します。 ローカルな Windows 環境では、出力ウィンドウ (Unicode 対応) が使用されます。 リモート Windows 環境では、このパラメータの設定に関係なく、Natural Web I/O インターフェイスクライアントが常に使用されます。 注意: メインフレームプラットフォームの場合は、代わりに NDV コンフィグレーションパラメータ TERMINAL_EMULATION が使用されます。下記を参照してください。

メインフレームプラットフォーム

次のプロファイルパラメータ/マクロは、メインフレームプラットフォームでのみ使用できません。

パラメータ	マクロ	説明
CFICU	NTCFICU	さまざまな Unicode 設定に対して Unicode サポートを有効にします。
使用できません。	NTCPAGE	NATCONFIG モジュールで、このマクロはコードページ、および置換文字、ロケール ID、照合テーブルなどのすべての関連情報を定義します。
PRINT	NTPRINT マクロの CP キーワードサブパラメータ	レポートのコードページを定義します。
CMPO	NTCMPO マクロの CPAGE キーワードサブパラメータ	コードページを区別する Natural プログラムを生成します。
OPRB	NTOPRB	使用される Adabas データベースが UES (ユニバーサルエンコーディングサポート) に対応している場合に、ユーザーエンコードを定義する ACODE/WCODE オプションを設定します。

メインフレーム用の Natural 開発サーバー

次の NDV コンフィグレーションパラメータは、メインフレームプラットフォーム用の Natural 開発サーバーでのみ使用できます。

設定	説明
TERMINAL_EMULATION=WEBIO	Natural Web I/O インターフェイスクライアント (Unicode をサポート) が入出力に使用されることを指定します。

エンコード情報

コードページデータのエンコードは、次に示す異なるレベルで指定できます。

- レベル 1- デフォルトコードページ

■ レベル 2 - 単一のオブジェクトのコードページ

レベル 1 - デフォルトコードページ


デフォルトコードページは、CP パラメータを使用して定義できます。Windows、UNIX、および OpenVMS プラットフォームでは、システムコードページを上書きして、すべてのコードページデータに対して有効になります。

レベル 2 - 単一のオブジェクトのコードページ

Natural ソース、バッチ入力 (CPOBJIN、CPSYNIN)、および出力ファイル (CPPRINT) に対してコードページを定義できます。

また、Windows、UNIX、および OpenVMS プラットフォームでは、タイプ ASCII、圧縮 ASCII、Unformatted、および CSV のワークファイルに対してコードページを定義できます。『コンフィグレーションユーティリティ』ドキュメントの「ワークファイル割り当て」を参照してください。

コードページがオブジェクトレベルで定義された場合、このコードページによってデフォルトコードページは上書きされます。

 **注意:** Windows、UNIX、および OpenVMS プラットフォームでは、すべてのオブジェクトに対して正しいコードページが定義されることが重要です。詳細については、「[既存アプリケーションの移行](#)」を参照してください。

エンコード情報を持つ Natural オブジェクトの展開

Windows、UNIX、および OpenVMS プラットフォーム

エンコード情報がすでに定義された Natural オブジェクトを展開する場合、エンコード情報はファイル FILEDIR.SAG に保存されており、Natural の外部からオブジェクトファイルのみを展開した場合はエンコード情報が失われることに注意する必要があります。

Natural オブジェクトを展開する場合、次のようにエンコード情報を保持できます。

- ライブラリ全体をコピーできます。その後、ライブラリのコピーをすべての Windows、UNIX、および OpenVMS プラットフォームに分散できます。この場合は、元のコードページが維持されます。ライブラリを Windows から UNIX または OpenVMS にコピーする場合は、Natural for UNIX または Natural for OpenVMS のネイティブエディタでオブジェクトを開くことができない場合があることに注意する必要があります。これらのエディタでは、デフォルトコードページを持つオブジェクトのみを開くことができます。
- エンコード情報を保持するオブジェクトハンドラを使用できます。この場合は、元のコードページが維持されます。Windows ライブラリが UNIX または OpenVMS でアンロードされる

場合は、Natural for UNIX または Natural for OpenVMS のネイティブエディタでオブジェクトを開くことができない場合があることに注意する必要があります。これらのエディタでは、デフォルトコードページを持つオブジェクトのみを開くことができます。

- Natural スタジオを使用して、オブジェクトをコピーして貼り付けることができます。SPoD 環境で、ターゲット環境がソース環境とは異なるプラットフォーム上にある場合、Natural はターゲット環境のデフォルトコードページを使用してオブジェクトを保存しようとします。これが不可能な場合、オブジェクトは UTF-8 フォーマットで保存されます。UNIX および OpenVMS ターゲットの場合、このことによって、ソースのすべての文字が UNIX または OpenVMS サーバーのデフォルトコードページで使用可能な場合は、Natural for UNIX または Natural for OpenVMS のネイティブエディタでオブジェクトを開けます。

メインフレームプラットフォーム

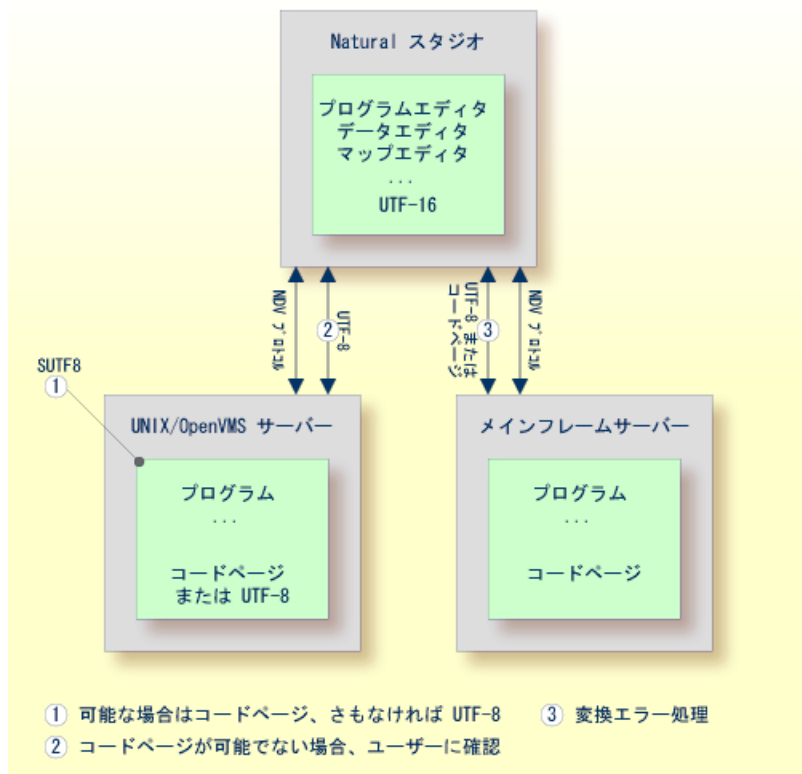
メインフレームプラットフォーム上のオブジェクトの場合、コードページ情報はオブジェクトディレクトリの一部であるため、オブジェクトのコードページ情報を保持するために特別な考慮事項はありません。

5 開発環境

▪ 開発環境	28
▪ 環境のカスタマイズ	29
▪ エディタ	30
▪ メインフレームでのエディタ、システムコマンド、およびユーティリティのコードページのサポート	31

開発環境

Unicode アプリケーションの開発環境は、Natural Single Point of Development (SPoD) です。



SPoD 環境では、Natural 開発サーバー (NDV) 上の Unicode アプリケーションの Natural オブジェクトは、Natural スタジオを使用して変更できます。サーバーによってサポートされている場合、ソースはクライアントとサーバー間で UTF-8 フォーマットで交換されます。


UNIX および OpenVMS 用の NDV サーバーでは、プロファイルパラメータ SUTF8 の設定によって、Natural オブジェクトをサーバーに保存するときを使用されるフォーマットが決まります。これは、ローカルの Windows の場合と同様に処理されます。

メインフレーム用の NDV サーバーでは、プロファイルパラメータ SRETAIN の設定に応じて、デフォルトまたは元のエンコードでオブジェクトが保存されます。

環境のカスタマイズ

Windows、UNIX、および OpenVMS プラットフォームでは、Natural コードを変更する前に、環境に対して正しいデフォルトコードページを定義することが重要です。詳細については、「[既存アプリケーションの移行](#)」を参照してください。

異なる言語の文字をソースに保存する場合、Windows、UNIX、または OpenVMS プラットフォームで UTF-8 フォーマットでソースを保存するか、ソースで 16 進 UH 定数を使用する必要があります。プロファイルパラメータ SUTF8 および SRETAIN を使用して、どのフォーマットでソースを保存するかを制御できます。次の表に、いくつかの状況と推奨設定を示します。

 **注意:** UNIX および OpenVMS では、パラメータ SUTF8 は SPoD 環境でのみ使用できません。

状況	設定	効果
ソースは Windows 上にあります。U 定数が必要です。	SUTF8=ON, SRETAIN=OFF	Natural 6.2 以降で保存する場合、すべてのソースは UTF-8 フォーマットで保存されます。新しいソースは UTF-8 フォーマットで作成されます。すべての文字を1つのソースに保存できます。
ソースは Windows、UNIX、または OpenVMS 上にあります。U 定数が必要であり、SPoD が開発に使用されます。	SUTF8=ON, SRETAIN=ON	元のコードページへの変換が不可能になった場合、すべてのソースは UTF-8 フォーマットで保存されます。可能な場合には、ソースのコードページは変更されません。新しいソースは UTF-8 フォーマットで作成されます。すべての文字を1つのソースに保存できます。UTF-8 フォーマットのソースは、SPoD でのみ変更できます。Natural for UNIX または Natural for OpenVMS エディタでは処理できなくなります。
ソースは Windows、UNIX、または OpenVMS 上にあります。U 定数は必要ありません。	SUTF8=OFF, SRETAIN=ON	すべてのソースは元のコードページで保存されます。新しいソースは、サーバーのデフォルトコードページで保存されます。1つのソースに保存できるのは、ソースコードページからの文字のみです。ソースは、引き続き Natural for UNIX または Natural for OpenVMS エディタで処理できます。
ソースは Windows、UNIX、OpenVMS、またはメインフレーム上にあります。U 定数が必要であり、SPoD が開発に使用されます。	SUTF8=OFF, SRETAIN=ON	すべてのソースは元のコードページで保存されます。新しいソースは、サーバーのデフォルトコードページで保存されます。1つのソースに保存できるのは、ソースコードページからの文字のみです。ソースは、引き続き Natural for UNIX、Natural for OpenVMS、および Natural for Mainframes エディタで処理できます。すべての Unicode 定数は、16 進定数 (UH) として定義される必要があります。

パラメータ SUTF8 が "OFF" に設定されており、異なる文字セットの文字を含むが、まだ UTF-8 フォーマットで保存されていないソースを格納する場合、生成プログラムは作成されるが、ソー


スは保存不可能であり、そのためにソースは変更されないままである場合があります。これは、異なる文字セットの文字がコメントまたはU定数で使用される場合に発生します。このため、異なる文字セットの文字を含むソースを作成する場合、およびソースをメインフレームプラットフォームに分散する必要がない場合は、パラメータ SUTF8 を "ON" に設定することをお勧めします。

パラメータ SRETAIN が "OFF" に設定されている場合、すべてのソースはデフォルトコードページで保存されます。この設定には注意が必要です。以前の Natural バージョンで作成されたソースがある場合、この設定によってコードページ情報が不適切になる可能性があるためです。この場合、ソースのエンコード情報は割り当てられておらず、ソースは常にデフォルトコードページ（システム変数 *CODEPAGE の値）で開かれます。デフォルトコードページがソースの正しいエンコードではない場合でも、このように機能することがあります。この場合、言語固有の一部の文字は正しく表示されません。そのようなソースが間違ったコードページで開かれ、SRETAIN が "ON" に設定された状態で保存された場合、ソースに対してエンコードは保存されません。ソースは、Natural が正しいデフォルトコードページで開始された場合に、後で正しく開くことができます。ただし、SRETAIN が "OFF" に設定された状態でソースを保存すると、デフォルトコードページがソースのエンコードとして保存されます。その後、ソースはこのコードページでのみ開かれます。このため、この設定は、すべての Natural ソースがデフォルトコードページでエンコードされていることが確かな場合にのみ使用する必要があります。

『[コンフィグレーションユーティリティ](#)』ドキュメントの「[地域の設定](#)」も参照してください。

エディタ

Natural for Windows エディタは、Unicode に完全に対応しています。SPoD 経由で、メインフレーム、UNIX、および OpenVMS ソースに対しても使用できます。Natural for Mainframes、Natural for UNIX、および Natural for OpenVMS で提供されるエディタは、Unicode 対応ではありません。

 **注意:** Natural for Mainframes で提供されるエディタでは、コードページのサポートが提供されません。「[メインフレームでのエディタ、システムコマンド、およびユーティリティのコードページのサポート](#)」を参照してください。

Natural スタジオ (Natural for Windows) のエディタを使用してソースが開かれるとき、ソースの内容は、対応するコードページから Unicode に変換されてから、エディタにロードされます。このことによって、システムコードページに含まれない文字がソースに含まれている場合でも、すべての文字を正しく表示できることが保証されます。ソースのコードページから Unicode への変換が失敗した場合は、エラーが表示され、エディタは開かれません。この場合、ユーザーはソースの正しいエンコードを定義する必要があります。ソースのエンコードは、[プロパティ] ダイアログボックスで変更できます（『[Natural スタジオの使用](#)』ドキュメントの「[ノードのプロパティ](#)」を参照）。

Windows、UNIX、および OpenVMS ソースの場合、Natural for Windows エディタによって、異なる言語の文字を含むソースを UTF-8 フォーマットで保存できます。メインフレームでは、UTF-8 ソースを保存できません。



注意: UNIX または OpenVMS ソースを UTF-8 フォーマットで、またはデフォルトコードページとは異なるコードページを使用して保存した場合、ソースは Natural for UNIX または Natural for OpenVMS のネイティブエディタで開くことができなくなります。メインフレームソースは、異なるコードページを使用して保存し、Natural for Mainframes のネイティブエディタで編集できます。

プログラムおよびソース内で Unicode 文字列を使用しない場合でも、Unicode 対応エディタには、インストールされているシステムコードページに関係なく、すべてのコードページのソースを記述できるという利点があります。例えば、"windows-1252" (Latin 1) コードページをインストールしている場合に、キリル語文字を含むプログラムを記述し、このプログラムを "windows-1251" (キリル語) コードページで保存できます。[名前をつけて保存] ダイアログボックスで、コードページ "windows-1251" を選択するだけです (『Natural スタジオの使用』ドキュメントの「新しい名前でのオブジェクトの保存」を参照)。

Natural for Windows プログラムエディタを使用して、テキスト定数を 16 進 Unicode 表現に変換できません (Natural for Windows 『エディタ』ドキュメントの「プログラムエディタ」セクションの「16 進形式への変換」を参照)。したがって、UTF-8 ソースが望ましくないプラットフォーム用に開発している場合、Unicode 定数のすべての文字を入力し、定数のすべての文字を選択し、それらを 16 進表示に変換して、Unicode 16 進定数の "UH" 接頭辞を追加できます。さらに、テキスト定数の文字または選択された文字範囲の上にマウスポインタを置くと、対応する 16 進 Unicode 表現がツールヒントに表示されます。

バイトオーダーマーク (BOM) は、データ文字列の先頭の文字コード "U+FEFF" で構成されます。その場所で、主にマークがないプレーンテキストファイルの、バイト順およびエンコード形式を定義する署名として使用できます。Windows では、バイトオーダーマークは一部のエディタ (メモ帳など) によって、UTF-8 ファイルをマークするために使用されます。Natural for Windows エディタでは、UTF-8 バイトオーダーマークはオブジェクトを読み込むときに認識されます。それまでにオブジェクトに他のエンコードが定義されていない場合は、Natural によって UTF-8 と解釈され、オブジェクトが保存されるときに、UTF-8 がそのオブジェクトのエンコードとして保存されます。この場合、バイトオーダーマークは削除されます。

メインフレームでのエディタ、システムコマンド、およびユーティリティのコードページのサポート

以下では次のトピックについて説明します。

- エディタ

■ システムコマンドおよびユーティリティ

エディタ

Natural for Mainframes で提供されるプログラム、マップ、およびデータエリアエディタは、Unicode 対応ではありません。代わりに、ソースはコードページ情報とともに保存されます。プロファイルパラメータ SRETAIN の設定に従って、Natural ソースがエディタにロードされる場合に、コードページ情報を持つソースが、ソースの現在のコードページから現在の Natural セッションのデフォルトコードページ（システム変数 *CODEPAGE の値）に自動的に変換される場合があります。変換できない文字がある場合は、コードポイント変換エラーがウィンドウに表示され、変換できないそれらのコードポイントの代替値が要求されます。このメッセージの表示は、パラメータ CPCVERR の現在の設定とは関係ありません。この場合、ユーザーは、ソースをデフォルトコードページに変換してエディタを開くか、または変換しないでエディタを開くことを決定できます。変換されたソースを保存または格納すると、新しいコードページ情報が保存されます。コードページ情報を持たないソース（以前の Natural バージョンで保存または STOW されたソースなど）は、変換されずにエディタにロードされます。プロファイルパラメータ SRETAIN の設定に従って、ソースの現在のコードページ情報が保存されます。

.I コマンドまたは画面分割を使用してソースを挿入した場合にも、プロファイルパラメータ SRETAIN の設定に従って、必要に応じてソースは変換されます。文字が変換できない場合は、代わりに定義済みの置換文字が挿入されます。

ソースのチェックおよび変換は、プログラムがソースエリアにロードされるのではなく、エディタが開始されるときに実行されます。プログラムが RUN *program-name* によって実行される場合、変換は実行されません。RUN *program-name* が NEXT 画面で入力されたかエディタ画面で入力されたかに応じて、動作は異なります。RUN *program-name* が NEXT 画面で入力された場合は、その後に変換は実行されません。エディタ画面で入力された場合は、プログラムの実行直後にエディタが開始され、変換が実行されます。

保存または STOW された既存の Natural ソースに割り当てられるコードページは、プロファイルパラメータ SRETAIN および CP の値に応じて異なります。以下の表を参照してください。

元のソースコードページ情報	SRETAIN の設定	CP が "OFF" 以外の値に設定されている場合の、SAVE または STOW 実行後のソースコードページ情報	CP が "OFF" 値に設定されている場合の、SAVE または STOW 実行後のソースコードページ情報
コードページ情報を含まないソース	SRETAIN=ON SRETAIN=(ON, EXCEPTNEW)	コードページ情報なし	コードページ情報なし
コードページ情報を含まないソース	SRETAIN=OFF	CP の評価の結果のコードページ	コードページ情報なし
ソースが <i>code page 1</i> でエンコードされている	SRETAIN=ON SRETAIN=(ON, EXCEPTNEW)	元のコードページ (<i>code page 1</i>)	元のコードページ (<i>code page 1</i>)
ソースが <i>code page 1</i> でエンコードされている	SRETAIN=OFF	CP の評価の結果のコードページ	元のコードページ (<i>code page 1</i>)

保存または STOW された新規の Natural ソースに割り当てられるコードページは、プロファイルパラメータ SRETAIN および CP の値に応じて異なります。これを以下の表に示します。

SRETAIN の設定	CP が "OFF" 以外の値に設定されている場合の、SAVE または STOW 実行後のソースコードページ情報	CP が "OFF" 値に設定されている場合の、SAVE または STOW 実行後のソースコードページ情報
SRETAIN=ON	CP の評価の結果のコードページ	コードページ情報なし
SRETAIN=OFF	CP の評価の結果のコードページ	コードページ情報なし
SRETAIN=(ON, EXCEPTNEW)	コードページ情報なし	コードページ情報なし

システムコマンドおよびユーティリティ

LIST

デフォルトでは、システムコマンド LIST によって、ソースがシステムファイル内に変換されずに保存された状態で表示されます。

LIST コマンドの CONVERTED オプションを使用すると、ソースのコードページ情報がある場合、ソースがデフォルトコードページ（システム変数 *CODEPAGE の値）に変換されます。変換不可能なすべての文字は、定義済みの置換文字によって置き換えられます。

LIST DIR

システムコマンド LIST DIRによって、Naturalソースの使用されているコードページ情報がディレクトリウィンドウに示されます。

SCAN

エディタと同様に、システムコマンド SCAN では、ソースが変換されてから実際の SCAN コマンドが実行されます。

オブジェクトハンドラ (SYSOBJH)

オブジェクトハンドラによって、さまざまなコードページ情報を持つソースがアンロードおよびロードされ、元のコードページ情報が保持されます。

転送フォーマットオプション UTF-8 によって、ソースはアンロード時にコードページから UTF-8 フォーマットに変換され、元のコードページの情報はワークファイルに保存されます。対応するロード機能によって、ソースは元のコードページに戻されるか、指定した場合、別のコードページに変換されます。また、このオプションを使用すると、以前の Natural バージョンで保存または STOW されたためにコードページ情報を持たないソースに、コードページ情報を与えることもできます。

内部フォーマットのソースのアンロードおよびロードでは、コードページ情報がある場合は保持されます。

SYSCP ユーティリティ・コードページ管理

SYSCP ユーティリティを使用して、コードページに関する情報を取得し、ソースのコードページ割り当てをチェックまたは変更できます。

6 Natural アプリケーションの入力／出力処理

- Unicode データの表示および入力 36
- Natural Web I/O インターフェイスクライアント 37

Unicode データの表示および入力

Unicode データを表示または入力する場合は、次の点に注意してください。

- ローカル開発環境で Natural for Windows を使用する場合、Natural 出力ウィンドウですべての Unicode 文字を表示および入力できます。
- リモート開発環境で Natural for Windows を使用する場合（SPoD）、すべての Unicode 文字を表示および入力するには、Natural Web I/O インターフェイスクライアント（[下記を参照](#)）が必要です。
- Natural for UNIX、Natural for OpenVMS、Natural for Mainframes、または Natural for Windows でアプリケーションを実行するときは、下記の「[Natural Web I/O インターフェイスクライアント](#)」を参照してください。

注意:

1. Windows で Unicode 対応の出力インターフェイスを使用している場合でも、現在選択しているフォントによってサポートされている Unicode 文字のみが表示されます。
2. Unicode データは 3270 端末では表示できません。

端末エミュレーションまたは 3270/3279 のようなメインフレーム端末を経由して Natural を実行する場合、ページはデフォルトコードページ（システム変数 *CODEPAGE の値）に変換されてから表示されます。デフォルトコードページに含まれていないすべての文字は、置換文字で置き換えられます。同様に、入力はコードページフォーマットでのみ可能であり、Unicode フォーマットに変換されてから U フォーマットフィールドに割り当てられます。置換文字は ICU 変換テーブルによって定義されると考える必要があります。この文字によっては、端末エミュレーションでガーベッジが表示される場合があります。UNIX および OpenVMS プラットフォームでは、プロファイルパラメータ SUBCHAR を設定することによって、この置換文字を変更できます。ただし、デフォルトコードページに含まれていない文字を表示するときは、Natural Web I/O インターフェイスを使用することを強くお勧めします。リモート Windows セッションを実行しているときは、どのような場合にも Natural Web I/O インターフェイスが使用されます。

コードページ指向のメインフレーム端末では、適切なコードページを選択することが重要です。Natural のデフォルトコードページ、端末のコードページ、さらには端末で使用されるフォントによって、特定の文字を正しく表示する機能が決定されます。

Natural Web I/O インターフェイスクライアント

Natural Web I/O インターフェイスクライアントは、Unicode 文字を含む非 GUI 情報を表示するために使用されます。次の環境で使用できます。

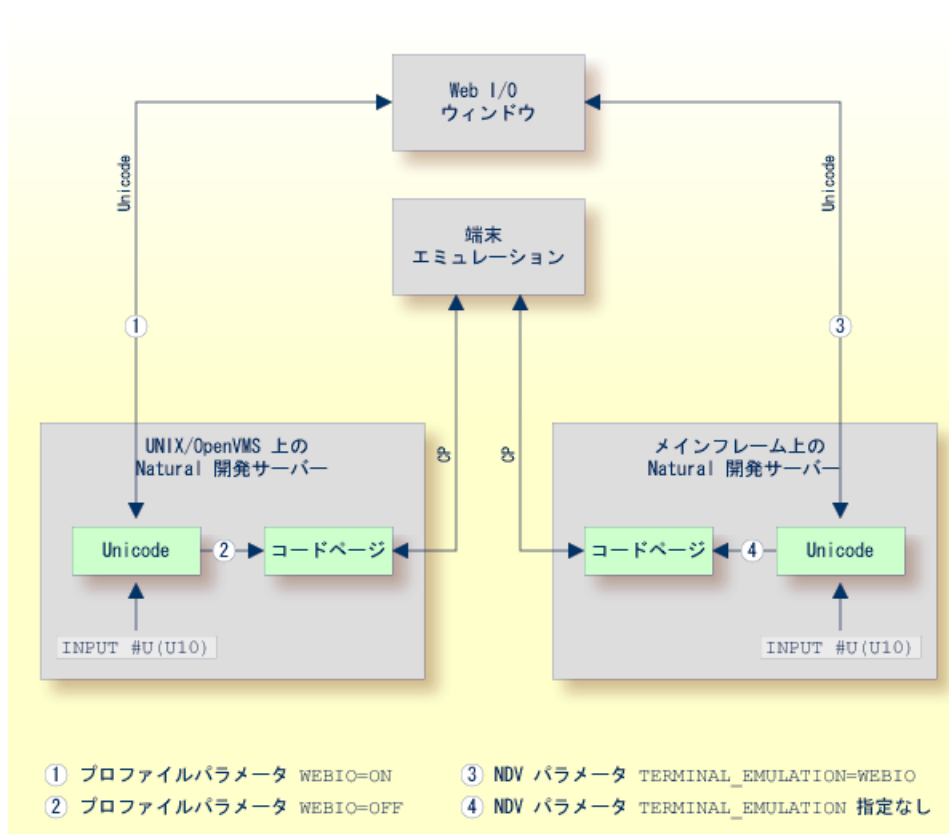
- SPoD 環境
- ランタイム環境

SPoD 環境

Natural Web I/O インターフェイスクライアントは、Natural for Windows を使用していて、リモート開発環境 (SPoD) で Natural Studio を使用している場合に呼び出すことができます。Natural for Windows ドキュメントに含まれる『SPoD を使用したリモート開発』の「Web I/O インターフェイスクライアント」を参照してください。

Natural Web I/O インターフェイスクライアントを使用している場合、リモート UNIX、OpenVMS、またはメインフレーム環境では、Unicode 対応ではない端末エミュレーションウィンドウの代わりに、またはリモート Windows 環境では出力ウィンドウの代わりに、Web I/O ウィンドウが表示されます。

次の図に、UNIX、OpenVMS、およびメインフレームでの、Unicode アプリケーションと Natural 開発サーバー (NDV) の SPoD 環境を示します。



Natural Web I/O インターフェイスクライアントを呼び出せるように、Natural 開発サーバーを次のように設定する必要があります。

■ UNIX と OpenVMS

リモート UNIX または OpenVMS 環境で Natural Web I/O インターフェイスクライアントを使用する場合は、NDV サーバーでプロファイルパラメータ WEBIO を "ON" に設定する必要があります。Natural for UNIX または Natural for OpenVMS ドキュメントの「コンフィグレーションユーティリティ」を参照してください。

■ メインフレーム

リモートメインフレーム環境で Natural Web I/O インターフェイスクライアントを使用する場合は、NDV サーバーで NDV コンフィグレーションパラメータ TERMINAL_EMULATION を "WEBIO" に設定する必要があります。詳細については、『Natural Development Server』ドキュメントの「NDV Configuration Parameters」を参照してください。さらに、Web I/O 端末コンバータモジュール NATWEB が Natural ニュークリアスにリンクされている必要があります。ユーザー画面サイズを決めるには、Natural プロファイルパラメータ TMODEL を使用できます。

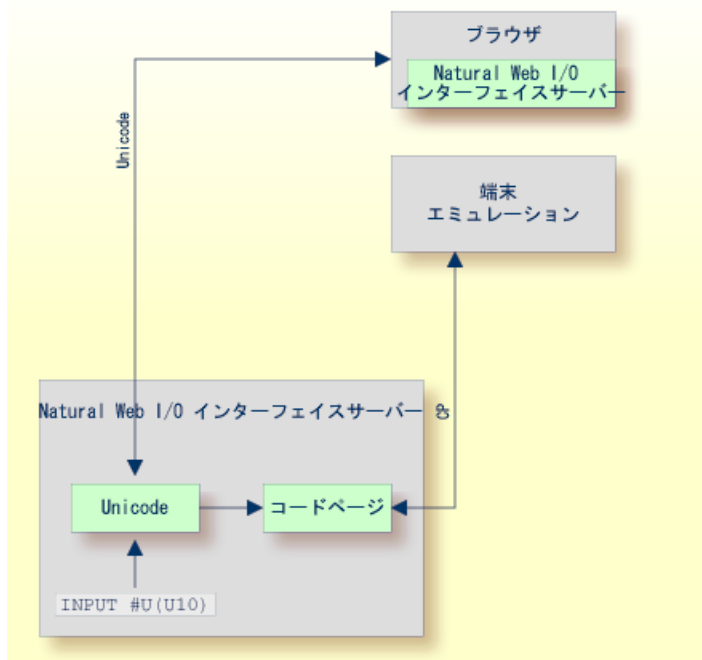
■ Windows

リモート Windows 環境では、プロファイルパラメータ WEBIO の設定に関係なく、Natural Web I/O インターフェイスクライアントが常に使用されます。

ランタイム環境

Natural Web I/O インターフェイスクライアントは、Natural for UNIX、Natural for OpenVMS、Natural for Mainframes、または Natural for Windows でアプリケーションを実行するときに表示されます。このクライアントは、Web／アプリケーションサーバーで実行されます。

次の図に、Unicode アプリケーションのランタイム環境を示します。



Natural では、セッションが Natural Web I/O インターフェイスクライアントから起動されたか、端末エミュレーションから起動されたかを自動的に認識します。

Natural Web I/O インターフェイスクライアントを使用するための前提条件は、次のとおりです。

■ Natural for UNIX および Natural for OpenVMS

デーモンとして実装されている Natural Web I/O インターフェイスサーバーがインストールされ、アクティブになっている必要があります。Natural for UNIX または Natural for OpenVMS ドキュメントの『*Natural Web I/O Interface*』を参照してください。

■ Natural for Mainframes

Natural Web I/O インターフェイスサーバーがインストールされ、設定されている必要があります。Natural for Mainframes ドキュメントの『*Natural Web I/O Interface*』を参照してください。さらに、Web I/O 端末コンバータモジュール NATWEB が Natural ニュークリアスにリンクされている必要があります。ユーザー画面サイズを決めるには、Natural プロファイルパラメータ TMODEL を使用できます。

■ **Natural for Windows**

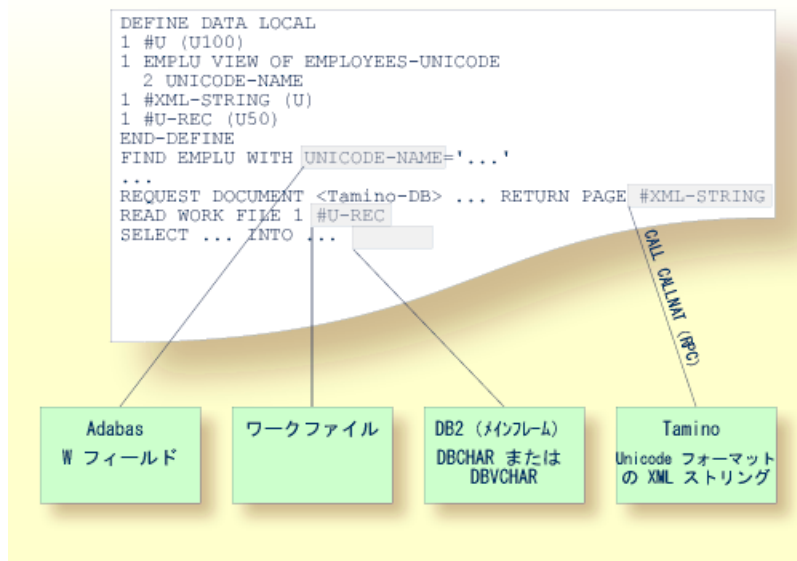
サービスとして実装されている Natural Web I/O インターフェイスサーバーがインストールされ、アクティブになっている必要があります。Natural for Windows ドキュメントの『*Natural Web I/O Interface*』を参照してください。

7 Unicode データストレージ

■ Unicode データ／パラメータアクセス	42
■ データベース管理システムインターフェイス	42
■ Windows、UNIX、および OpenVMS プラットフォームでのワークファイルおよび出力ファイル	43
■ メインフレームプラットフォーム上のワークファイルと出力ファイル	47

Unicode データ / パラメータアクセス

次の図に、Unicode データおよびパラメータがどのようにアクセスされるかを示します。



データベース管理システムインターフェイス

以下では次のトピックについて説明します。

- [Adabas データベースの Unicode データへのアクセス](#)
- [DB2 データベースの Unicode データへのアクセス](#)

Adabas データベースの Unicode データへのアクセス

Natural を使用すると、Adabas データベースのワイド文字フィールド（フォーマット W）にアクセスできます。

データ定義モジュール

Adabas ワイド文字フィールド（W）は、Natural データフォーマット U（Unicode）にマップされます。

アクセスコンフィグレーション

Natural は Adabas からデータを受け取り、共通のエンコードとして UTF-16 を使用してデータを Adabas に送ります。

このエンコードは、OPRB パラメータによって指定され、オープン要求によって Adabas に送信されます。これはワイド文字フィールドに使用され、Adabas ユーザーセッション全体を通して適用されます。

詳細については、『プログラミングガイド』の「Adabas データベースのデータへのアクセス」の「Unicode データ」を参照してください。

DB2 データベースの Unicode データへのアクセス

Natural では、ユーザーは DB2 データベースの CHAR/WCHAR フィールドに Unicode データとしてアクセスできます。

『データベース管理システムインターフェイス』ドキュメントの「Natural for DB2」も参照してください。

Windows、UNIX、および OpenVMS プラットフォームでのワークファイルおよび出力ファイル

以下では次のトピックについて説明します。

- WRITE WORK FILE
- READ WORK FILE
- ワークファイルタイプ転送の特別な考慮事項
- 出力ファイル

WRITE WORK FILE

次の情報は、ステートメント WRITE WORK FILE に適用されます。このステートメントの詳細については、『ステートメント』ドキュメントを参照してください。


コードページデータ

次のワークファイルタイプによって、コードページデータが書き込まれます。


- ASCII および圧縮 ASCII
- Unformatted
- CSV
- Entire Connection

ワークファイルタイプおよびコードページは、コンフィグレーションユーティリティで定義される必要があります。詳細については、『コンフィグレーションユーティリティ』の「ワークファイル設定」を参照してください。

オペランド A (英数字) および U (Unicode) で定義されたすべての Natural データは、指定されたコードページに変換されます。コードページが指定されていない場合、すべてのデータは CP パラメータで定義されたデフォルトコードページに変換されます。

 **注意:** ワークファイルでは、書き込まれるすべての A および U オペランドデータは、コードページフォーマットです。

U オペランドデータをこれらのワークファイルに書き込み、後でこれらのワークファイルからデータを欠落させずに読み取る必要がある場合、UTF-8 をコードページとしてコンフィグレーションユーティリティで定義する必要があります。この場合、すべての A および U オペランドデータは、UTF-8 フォーマットで書き込まれます。ワークファイルもコードページ UTF-8 を使用して設定されている後続の READ WORK FILE ステートメントによって、オペランド U データはデータを欠落させずに読み取られます。

 **注意:**

1. UTF-8 フォーマットで書き込まれたワークファイルデータは、UTF-8 をサポートするテキストエディタ (Windows プラットフォームのメモ帳など) で読み取ることができます。
2. オペランド B (バイナリ) で定義された Natural データは、コンフィグレーションユーティリティで指定されたコードページに変換されません。これらのデータは、Natural に保存されたときのままで書き込まれ、コードページ変換は行われません。

上記のワークファイルタイプのいずれかが指定され、ワークファイルに対してコードページ UTF-8 が定義された場合、ワークファイル属性 BOM (バイトオーダーマークを書き込む) および NOBOM (バイトオーダーマークを書き込まない) が有効になります。これらの属性は、コンフィグレーションユーティリティのワークファイルカテゴリで、DEFINE WORK FILE ステートメントを使用して指定できます。ワークファイルに対してコードページ UTF-8 が定義され、ワークファイル属性 BOM が指定された場合、UTF-8 バイトオーダーマーク (16 進表示: H'EFBBBF') がワークファイルの先頭、ワークファイルデータの前に書き込まれます。

上記のワークファイルタイプ以外のワークファイルタイプがワークファイルの書き込みに使用された場合、またはワークファイルに対して UTF-8 以外のコードページが定義された場合は、ランタイム時に属性 BOM の指定は無視されます。次の表に、ステートメント WRITE WORK FILE および READ WORK FILE の処理でのランタイム時の動作を示します。

コードページおよび属性の設定	WRITE WORK FILE	READ WORK FILE
ワークファイルにコードページ UTF-8 が指定されていません (デフォルト)。 ワークファイル属性 BOM および NOBOM は有効ではありません。	UTF-8 バイトオーダーマークが書き込まれていません。 UTF-8 への変換は行われません。	UTF-8 バイトオーダーマークの確認は行われません。 UTF-8 からの変換は行われません。
ワークファイルにコードページ UTF-8 が指定されています。	UTF-8 バイトオーダーマークが書き込まれています。	UTF-8 バイトオーダーマークを確認してください。

コードページおよび属性の設定	WRITE WORK FILE	READ WORK FILE
ワークファイル属性 BOM が指定されています。	A および U フィールドは UTF-8 に変換されます。	UTF-8 バイトオーダーマークが見つかった場合、そのマークはワークファイルデータから削除されます。フィールドは UTF-8 からデフォルトコードページに変換されます。U フィールドは UTF-8 から Natural の内部ランタイム表現である UTF-16 に変換されます。
ワークファイルにコードページ UTF-8 が指定されています。	UTF-8 バイトオーダーマークが書き込まれていません。	UTF-8 バイトオーダーマークを確認してください。
ワークファイル属性 NOBOM (デフォルト) が指定されています。	A および U フィールドは UTF-8 に変換されます。	UTF-8 バイトオーダーマークが見つかった場合、そのマークはワークファイルデータから削除されます。フィールドは UTF-8 からデフォルトコードページに変換されます。U フィールドは UTF-8 から Natural の内部ランタイム表現である UTF-16 に変換されます。

バイナリデータ

次のワークファイルタイプによって、バイナリデータ（オペランドフォーマット U の UTF-16 など）が書き込まれます。

- SAG
- Portable

オペランド A および U で定義された Natural データは、コードページに変換されません。これらのデータは、ワークファイルにバイナリフォーマットで書き込まれます。U オペランドデータの場合、これは UTF-16 で行われます。

READ WORK FILE

次の情報は、ステートメント READ WORK FILE に適用されます。このステートメントの詳細については、『ステートメント』ドキュメントを参照してください。Natural for Windows、Natural for UNIX、および Natural for OpenVMS ドキュメントで、RECORD オプションについてリストされている制限に注意してください。

コードページデータ

次のワークファイルタイプが使用されるとき、Natural U (Unicode) オペランドに読み取られるワークファイルデータは、指定されたコードページから UTF-16 に変換されます。

- ASCII および圧縮 ASCII
- Unformatted
- CSV

■ Entire Connection

A（英数字）オペランドに読み取られるデータは、必要に応じて、指定されたコードページから、パラメータ CP で定義されたデフォルトコードページに変換されます。

上記のワークファイルタイプのいずれかが指定され、ワークファイルに対してコードページ UTF-8 が定義された場合、READ WORK FILE ステートメントによって、ワークファイルで UTF-8 バイトオーダーマークが自動的に確認されます。ワークファイルの先頭で UTF-8 バイトオーダーマークが見つかった場合、そのマークは削除されます。ワークファイルから読み取られたデータは、UTF-8 からデフォルトコードページに変換されます。

データが別のワークファイルタイプから読み取られた場合、バイトオーダーマークの確認は実行されず、したがってバイトオーダーマークは削除されません。

ステートメント WRITE WORK FILE および READ WORK FILE の処理でのランタイム時の動作の詳細については、[前](#)のセクションの表を参照してください。

バイナリデータ

次のワークファイルタイプが使用されるとき、ワークファイルデータは変換されずに Natural オペランド A および U に読み取られます。つまり、バイナリフォーマットで読み取られます。

■ SAG

■ Portable

ワークファイルタイプ Portable では、オペランドフォーマット U のデータのエンディアン変換がサポートされます。

ワークファイルタイプ転送の特別な考慮事項

オペランドフォーマット U は、一般にワークファイルタイプ転送がサポートされています。Entire Connection によって、選択されたファイルタイプの Unicode の読み取りまたは書き込みができない場合は、ランタイムエラーメッセージが表示されます。

出力ファイル

出力ファイルの Unicode データの処理は、選択された論理デバイス（LPT1～LPT31）の出力方法によって異なり、現在は GUI（Windows のみ）または TTY です。

出力方法に関係なく、データは UTF-16 フォーマットで Natural 出力サービスに渡されます。つまり、フォーマット A フィールドデータはすでに Unicode に変換されています。

GUI 出力方法

この Windows のみの出力方法では、データは Unicode（UTF-16）フォーマットで Windows プリンタドライバに渡されます。これは Windows での標準のデータ出力方法であるため、この

データはドライバによって常に適切に処理されます。したがって、この出力方法は、システムコードページに含まれない文字が使用されている場合に Windows で推奨される出力方法です。

TTY 出力方法

この出力方法では、データは、デフォルトで内部 (UTF-16) フォーマットからシステムコードページに変換されます。ただし、プリンタプロファイルを使用することによって、データが代わりに UTF-8 フォーマットに変換されるか、または任意の外部コードページに追加変換されることを指定できます。これらの代替手段の詳細については、『[コンフィグレーションユーティリティ](#)』の「プリンタプロファイル」を参照してください。

データをシステムコードページに変換するデフォルトの動作の理由は、現在、UTF-8 フォーマットの未加工テキストデータを直接受け取ることが可能なプリンタがないことです。

メインフレームプラットフォーム上のワークファイルと出力ファイル

以下では次のトピックについて説明します。

- [ワークファイル](#)
- [出力ファイル](#)

ワークファイル

ワークファイルの書き込みおよび読み取りを行う場合、Unicode データについて特別な考慮事項はありません。他のすべてのデータタイプと同様に、Unicode データはそのまま書き込みおよび読み取りされ、変換は行われません。

出力ファイル

Unicode データを出力ファイルに送信する場合、1 つまたは 2 つの変換手順が発生します。

最初の手順で、出力ファイルに含まれる Unicode データは、セッションのデフォルトコードページに変換されます。結果として、このデフォルトコードページに含まれないすべての文字は、置換文字で置き換えられます。

この変換された出力行を実際出力アクセスメソッドに渡す前に、論理プリンタに対してコードページが指定されているかどうか追加でチェックされます。これは、`DEFINE PRINTER` ステートメントの `CODEPAGE` オペランドまたは `PRINT` パラメータの `CP` サブパラメータによって実行されている場合があります。そのようなコードページが指定されている場合は、それに従って、Unicode 部分だけではなく出力行全体が 2 番目の手順で変換されます。

変換された出力行がアクセスメソッドに渡されます。つまり、出力アクセスメソッドが Unicode データを受け取ることはありません。

例：

```
DEFINE PRINTER (1) CODEPAGE 'IBM01140'  
WRITE (1) 'HELLO' U'WORLD'  
END
```

8 プラットフォームの相違

- Windows、UNIX、および OpenVMS プラットフォーム 50
- メインフレームプラットフォーム 51

Windows、UNIX、および OpenVMS プラットフォーム

Windows、UNIX、および OpenVMS プラットフォームでは、Natural は内部的に Unicode 対応です。つまり、文字列を含む数多くの構造は、現在 Unicode フォーマットです。例えば、現在 Natural ソースエリアは Unicode フォーマットです。このため、Natural コードを書き込むとき、およびカタログするときに、Natural I/O および Natural 開発環境でランタイム時に Unicode データを処理できます。

最初のバージョンには、いくつかの例外があります。Natural ダイアログ（エディタおよびランタイム）は Unicode 対応ではありません。これらのモジュールは、後のバージョンでは Unicode 対応される予定です。

Natural は内部的に Unicode 対応ですが、既存のすべてのデータには、現在はコードページフォーマットがあります。結果として、Natural バージョン 6.2 以降で使用された場合、このようなデータはすべてコードページフォーマットから Unicode フォーマットに変換されます。例えば、ソースがプログラムエディタで開かれた場合、コードページファイルフォーマットから Unicode ソースエリアフォーマットへの変換が実行されます。U フォーマットを使用しない場合でも、このことには利点があります。インストールされているシステムコードページに関係なく、すべての言語固有の文字が表示されます。ただし、正しいコードページ情報を定義するのはユーザーの責任です。詳細については、「[既存アプリケーションの移行](#)」を参照してください。

Natural オブジェクトをカタログするとき、U 接頭辞を使用して定義されていないすべての定数は、対応するソースのコードページに変換されます。ソースが UTF-8 フォーマットの場合、これらの定数はデフォルトコードページに変換されます。



注意: ほとんどの場合、Unicode データには、コードページデータよりも多くのメモリスペースが必要です。そのため、Natural バージョン 6.2 以降では、Natural パラメータ USIZE を増やす必要がある場合があります。

Windows

Unicode は、ローカルな Natural for Windows 環境で完全にサポートされます。

エディタは Unicode 対応であり、使用できるすべての文字を入力できます。ソースを保存するとき、Natural では最初にソースを元のコードページに変換しようとします。このコードページにない文字がソースに含まれているために失敗した場合、その後の処理はパラメータ SUTF8 の設定によって異なります。SUTF8 が "ON" の場合、ソースは UTF-8 フォーマットで保存されます。SUTF8 が "OFF" の場合、ユーザーは、ソースを元のコードページで保存するか、現在の保存をキャンセルするかを確認されます。ソースを元のコードページで保存することをユーザーが決定した場合、見つからない文字は置換文字で置き換えられます。また、[名前をつけて保存] ダイアログボックスで、コードページを明示的に選択できます。


プログラムエディタは、Unicode 双方向アルゴリズムをサポートするために拡張されています。

出力ウィンドウも、Unicode 対応です。文字がキーボードで入力される場合、A フォーマットフィールドは、デフォルトコードページで使用可能な文字のみを受け入れます。

UNIX と OpenVMS

完全な Unicode サポートは、SPoD および Natural Web I/O インターフェイスでのみ利用できます。SPoD は、Natural ソースへの Unicode 入力に必要です。ローカルな Natural for Windows 環境について前述した内容と同じことが適用されます。Natural Web I/O インターフェイスは、Natural アプリケーションからの Unicode I/O に必要です。

Natural が端末エミュレーションを経由して使用される場合、すべての出力は Unicode からデフォルトコードページに変換されてから表示されます。デフォルトコードページで使用できない文字は、デフォルトコードページの置換文字で置き換えられます。同様の場合の入力は、デフォルトコードページに基づいてのみ可能です。

 **注意:** UTF-8 フォーマットの Natural ソースは、Natural for UNIX または Natural for OpenVMS ネイティブエディタを使用して開くことができません。

メインフレームプラットフォーム

Natural ランタイム環境は、Unicode サポートに対応しています。Unicode 文字は、デフォルトコードページ（システム変数 *CODEPAGE の値）に変換されてから、端末に表示されます。デフォルトコードページに同等の文字がない Unicode 文字は、置換文字によって置き換えられます。

SPoD での Natural Web I/O インターフェイスでは、Unicode 文字は端末エミュレーションによって完全にサポートされています。この場合、U フォーマットフィールドが表示され、Unicode として正しく入力できます。デフォルトコードページの同等の文字に変換されません。Natural Web I/O インターフェイスは、NDV サーバーコンフィグレーションパラメータ `TERMINAL_EMULATION=WEBIO` によってアクティブになります。システム変数 *DEVICE には、"BROWSER" が含まれています。

Natural コンパイラ、エディタ、および Natural システムファイルでは、Unicode でエンコードされたオブジェクトソースはサポートされません。オブジェクトソース内にコード化された Unicode 定数はデフォルトコードページに保存され、カタログ化オブジェクトに Unicode コードポイントが含まれます。デフォルトコードページに同等の文字がない Unicode 定数を定義する唯一の方法は、16 進定義 (UH) を使用することです。

コードページ変換および Unicode サポートでは、ICU ライブラリが提供する機能が使用されます。この機能を提供する ICU モジュールのサイズは、使用する ICU 機能に応じて、3~12 MB です。コードページ変換と Unicode サポートがどちらも必要ない場合は、これらのモジュールを Natural ニュークリアスにリンクする必要はありません。柔軟性を高めるために、Natural セッションの初期化中に、これらのモジュールを Natural ニュークリアスに動的にリンクすることもできます。ICU 機能を使用することで、必要な Natural スレッドサイズは、最大 200 KB のストレージ分増加します。

以下では次のトピックについて説明します。

- **さまざまな目的のための NATICU モジュール**
- セッションモード
- CFICU パラメータ
- 共有 FUSER
- CPAGE コンパイラオプション
- プログラムソース
- NTCPAGE マクロ
- データベースの Unicode およびコードページのサポート
- 変換テーブル
- マルチバイトコードページのサポート

さまざまな目的のための NATICU モジュール

NaturalでUnicodeおよびコードページのサポートを有効にするには、ICUライブラリモジュールをリンクする必要があります。Naturalには、次に示すように、さまざまな目的のためのICUライブラリのさまざまな実装があります。

■ NATICU

この実装を使用する対象は、ほとんどのヨーロッパの国と北米および南米の国です。英語、ドイツ語、フランス語、およびスペイン語の地域用に、コードページおよびロケールIDが削減されています。サポート言語が削減されているため、サイズは比較的小さくなっています。

このモジュールのもう一つの特徴は、照合サービスです。照合サービスは、Unicode文字列を比較するために使用されます。照合サービスでは、アルファベット順が言語によって異なることが考慮されています。世界の言語と表記体系および使用されるさまざまな順序に対応することは、大きな課題です。ただし、ICU照合サービスには、ロケールを区別して文字列を比較するすぐれた方法があります。例えば、文字"Ä"は、ドイツ語ロケールでは"A"と"B"の間にソートされ、スウェーデン語ロケールでは"Z"の後にソートされます。リトアニア語では、文字"y"は"i"と"k"の間にソートされます。照合サービスのICU実装は、Unicode Collation Algorithmに従っており、ISO 14651に準拠します。アルゴリズムは、専門家によって複数の照合で設計および確認されているため、堅牢で包括的です。

NATICUは、次の表のコードページおよびロケールを提供します。

■ NATICUCV

この実装はNATICUと同じですが、照合サービスはありません。照合サービスは、ICU内で非常に大きなパッケージであるためです。NATICUCVを使用する場合、ロケールに依存する照合サービスの比較の代わりに、Unicode文字列に対してバイナリ比較が実行されます。すべてのUnicodeデータが同じコードページから発生しており、正規化されている場合に、このモジュールを使用できます。

NATICUCVは、次の表のコードページおよびロケールを提供します。

■ NATICUXL

この実装を使用する対象は、NATICUの削減されたコードページおよびロケールIDに含まれない、世界のすべての地域です。

NATICUXLには、現在サポートされているICUバージョンによって提供される、すべてのコードページおよびロケールIDが含まれています。サポートされるコードページおよびロケールIDの概要については、ICUのホームページ (<http://demo.icu-project.org/icu-bin/convexp>) を参照してください。

NATICUがデフォルトであり、Naturalニュークリアスにリンクされています。NATICUは、セッションの初期化中にNATICUCVまたはNATICUXLで置き換えることができます。必要なモジュールをセッションパラメータRCA (RCA=NATICUCVまたはRCA=NATICUXL) によってロードします。

NATICU および NATICUCV は、次のコードページおよびロケールを提供します。

コードページ	ロケール
IBM037	de_DE
IBM273	en_US
IBM1025	es_ES
IBM1026	fr_FR
IBM1047	sv_SE
IBM1097	
IBM01140	
IBM01141	
IBM01145	
IBM01146	
IBM01147	
US (IBM01140 のエイリアス)	
DE (IBM01141 のエイリアス)	
ES (IBM01145 のエイリアス)	
EN (IBM01146 のエイリアス)	
FR (IBM01147 のエイリアス)	
IBM-37_P100-1995,SWAPLFNL	
IBM-1047_P100-1995,SWAPLFNL	
IBM-1140_P100-1997,SWAPLFNL	
EBCDIC-XML-US	
EDF03DRV (Siemens コードページ)	
EDF03IRV (Siemens コードページ)	
EDF04DRV (Siemens コードページ)	
EDF04IRV (Siemens コードページ)	
IBM-290 (日本語コードページ SBCS)	
IBM-930 (日本語コードページ SBCS/DBCS)	
IBM-939 (日本語コードページ SBCS/DBCS)	
IBM-1390 (日本語コードページ SBCS/DBCS)	
IBM-1399 (日本語コードページ SBCS/DBCS)	
IBM-932 (日本語コードページ ASCII MBCS)	
IBM-942 (日本語コードページ ASCII MBCS)	

プラットフォームの相違

コードページ	ロケール
IBM-943 (日本語コードページ ASCII MBCS)	
EUC-JP (日本語コードページ ASCII MBCS)	
IBM-420 (RTL コードページ)	
IBM-424 (RTL コードページ)	
IBM-916 (RTL コードページ)	

セッションモード

パラメータ CFICU および CP を使用して、特定の目的に合わせて Natural を調整できます。

設定	説明
CFICU=OFF、CP=OFF	互換モード。Unicode およびコードページのサポートを使用しない既存のアプリケーションを実行するため。レガシー変換テーブルが I/O 変換に使用されます。前のバージョンと比較して、リソースの消費 (CPU 時間およびバッファ使用率) に大きな増加はありません。このモードでは、ICU が Natural ニュークリアスにリンクされている必要はありません。
CFICU=ON、CP=OFF	Unicode およびコードページの変換 (MOVE ENCODED) を使用するが、デフォルトコードページのサポートを使用しない新規アプリケーション用。したがって、システム変数 *CODEPAGE は空です。U フォーマット変数は使用できますが、MOVE A TO U などは使用できません。これにはデフォルトコードページ情報が必要であるためです。使用できるデフォルトコードページがないことを示すエラー NAT3411 が発行されます。
CFICU=ON, CP=value*	完全な Unicode およびコードページのサポートを使用する新規アプリケーション用。
CFICU=OFF, CP=value*	この組み合わせは可能ですが、無意味です。コードページのサポートには、変換のために ICU サービスが必要であるためです。したがって、この場合は CFICU=ON が強制され、セッション初期化メッセージが発行されます。

* value は "OFF" 以外の任意の値です。

CFICU パラメータ

パラメータ CFICU とそのサブパラメータについては、『パラメータリファレンス』で詳細に説明されています。一部のサブパラメータは、パフォーマンスに影響します。

Unicode 文字列の比較に照合サービスが使用される場合、両方の文字列は正規化されているかどうかチェックされます。チェック自体に多くの CPU 時間が消費されます。文字列がすでに正規化されていることが確実な場合は、チェックをオフ (COLNORM=OFF) に切り替えることができます。

Unicode では、同じ文字を 1 つのコードポイントとして、または 2 つ以上のコードポイントの組み合わせとして表すことができます。例えば、ドイツ語の文字 "ä" は、"U+00E4" か、コードポイント "U+0061" および "U+0308" の組み合わせで表すことができます。Unicode から例えば IBM01140 への変換では、組み合わせられた文字は 1 つのコードポイントとして処理され、"a" と

それに続く置換文字が生成されます。これは、コードポイント "U+0308" がターゲットコードページで表現されないためです。CNVNORM=ON では、実際の変換の直前に正規化が実行されます。正規化では、追加の CPU 時間および一時ストレージが消費されます。MOVE ステートメント (MOVE NORMALIZED 以外) に結合文字が含まれないことが確実な場合は、CNVNORM を "OFF" に設定してパフォーマンスを向上させる必要があります。可能なすべての組み合わせが、1 つのコードの Unicode コードポイントで表されます。

Unicode からコードページへの変換、およびその逆の変換のパフォーマンスは高くはありません。その理由は、ICU 実装が C++ で記述されていることと、世界中のほぼすべての Unicode、コードページ、および言語を対象としていることです。ただし、変換を速くするために、変換テーブルによって一部のコードページを Unicode に (およびその逆に) マップできます。アクセラレータテーブルは、CPOPT サブパラメータによってアクティブになります。"ON" に設定した場合は、Natural によって、セッションの初期化中に ICU 変換機能を使用して、2 つのアクセラレータテーブルが自動的に作成されます。サイズが 512 バイトの最初のテーブルは、コードページから Unicode への変換に使用され、サイズが 65535 バイトのもう一つのテーブルは、Unicode からコードページへの変換に使用されます。Natural セッション中、すべての変換は ICU の呼び出しではなく、アクセラレータテーブルによって実行されます。アクセラレータテーブルは、デフォルトコードページ (*CODEPAGE) に対してのみ提供されます。(例えば、MOVE ENCODED ステートメントの) 一時的なコードページでは、モジュール NATCPTAB がリンクされていない場合、アクセラレータテーブルは使用されません。このモジュールがリンクされている場合は、ICU データベースに基づく最大 30 個のアクセラレータテーブルが、パフォーマンスを向上させるために使用されます。

共有 FUSER

Natural ソースは、保存前は Unicode または UTF-8 に変換されないため、以前の Natural バージョンによって読み取ることができます。Natural バージョン 4.2 (以降) のソースの追加コードページ情報は、ソースのヘッダーに保存されます。ソースが以前の Natural バージョンによってアクセスされた場合、ヘッダーのコードページ情報は無視されるだけです。

CPAGE コンパイラオプション

コンパイラオプション CPAGE によって、作成時に使用されたコードページとは異なるコードページで実行可能なオブジェクトが作成されます。つまり、作成時のコードページでコード化されたオブジェクトのすべての英数字定数は、ランタイム時にアクティブなコードページに変換される必要があります。Natural オブジェクトローダーが英数字定数を検出して変換できるように、コンパイラによって追加のテーブルが作成されます。このことにより、使用される英数字定数の数に応じて、生成されるオブジェクトのサイズが大きくなります。ランタイム時の変換では、追加の CPU 時間が消費されます。デフォルトコードページ (システム変数 *CODEPAGE の値) が作成時のコードページと同じ場合、またはセッションにデフォルトコードページがない場合 (CP=OFF) は、変換は行われません。パラメータ CPCVERR の設定とは関係なく、変換エラーは無視されます。コンパイラオプション CPAGE が "OFF" に設定されている場合は、ランタイム時に変換は行われず、英数字定数はそのまま処理されます。

プラットフォームの相違

次のサンプルプログラムは、コードページ IBM01141 (German) でカタログされ、デフォルトコードページ IBM01140 (us) で実行されます。文字 "Ä"、"Ö"、および "Ü" は両方のコードページで定義されていますが、コードポイントは異なります。

例 1 - CPAGE=OFF :

```
OPTIONS CPAGE=OFF
WRITE *CODEPAGE 'ÄÖÜ'
END
```

コードページ IBM01140 (us) を使用した出力 :

```
Page      1
IBM01140                                     φ\!
```

例 2 - CPAGE=ON :

```
OPTIONS CPAGE=ON
WRITE *CODEPAGE 'ÄÖÜ'
END
```

コードページ IBM01140 (us) を使用した出力 :

```
Page      1
IBM01140                                     ÄÖÜ
```

プログラムソース

メインフレーム上の Natural ソースは、Unicode フォーマットではなく、現在の Natural セッションのデフォルトコードページで保存されています。コードページの名前は、ソースのディレクトリに保存されています。したがって、以前の Natural バージョンと比較して、ソースのサイズは変わりません。しかし、ソースのコードページが Natural セッションのデフォルトコードページと同じかどうか、エディタによるチェックが必要です。コードページが異なる場合は、ソースをデフォルトコードページに変換する必要がありますが、変換エラーが発生する可能性があります。デフォルトコードページでマップされないソースの文字がある場合、エディタにウィンドウが表示され、失敗した文字を手動で変換できます。例えば、コード IBM01140 で作成されたソースに、次の行が含まれています。

```
WRITE '100 €'
```

このソースをコードページ IBM037 で実行されている Natural で再度編集した場合、文字 "€" がコードページ IBM037 でマップされないため、変換エラーが発生します。

変換は、ソースがロードされるのではなく、エディタが開始されるときに行われることに注意してください。

NTCPAGE マクロ

コードページ名の最も一般的な標準は、IANA 名です。そのため、システム変数 *CODEPAGE には、デフォルトコードページの IANA 名が含まれています。IBM の場合、コードページはその Coded Character Set ID (CCSID) で修飾されています。Siemens の場合、Coded Character Set Name (CCSN) が最も一般的です。現在、Adabas では Entire Conversion Service 定義 (ADA ECS) が使用されています。マクロ NTCPAGE を使用すると、これらの異なる名前を明確な IANA 名に割り当てることができます。NTCPAGE は、Natural コンフィグレーションファイル (NATCONFIG) の一部です。

CP パラメータは、Natural コンフィグレーションファイルに定義されている値のみを受け入れます。CP パラメータで入力されるのが IANA 名か、CCSID/CCSN か、エイリアス名かは重要ではありません。エイリアス名には、より意味のある名前をコードページに割り当てるために使用するユーザー定義名が可能です。いずれの場合にも、*CODEPAGE には、選択されたコードページの IANA 名が含まれます。

また、コードページに対してプレースホルダ文字を定義できます。これにより、そのコードページのデフォルトの置換文字が上書きされます。デフォルトの置換文字は、通常は非表示文字です (EBCDIC コードページの H'3F' など)。プレースホルダ文字は、非表示文字が端末に送信されるのを防ぐために使用できます。

例：

```
NTCPAGE IANA=IBM01140,CCSID=1140,ECS=1140,ALIAS='US',PHC=003F
```

値 "IBM01140"、"1140"、または "US" を CP パラメータを使用して入力して、コードページをアクティブにすることができます。*CODEPAGE には、名前 IBM01140 が含まれます。コードページの置換文字は、引用符 (?) である "U+003F" によって置き換えられます。

使用できるコードページの数、使用される ICU 実装によって異なります。

Natural バージョン 4.2.5 以降および ICU バージョン 3.8 以降ではそれぞれ、該当する NTCPAGE エントリを省略できます。その代わりに、現在使用されているデータパッケージ内に定義されているすべてのコードページを Natural で使用できます。NTCPAGE エントリが必要となるのは、代替エイリアス名またはプレースホルダ文字が必要な場合のみです。

データベースの Unicode およびコードページのサポート

Adabas は、幅広いコードページをサポートし、世界中の言語を扱うことができます。文字エンコードおよびデータ変換は、Adabas の内部で処理されます。このとき、ストレージ（ファイルエンコード）とユーザーへの表示（ユーザーエンコード）用のデフォルトエンコードとして Unicode が使用されます。コードページまたは Unicode のサポート（CFICU=ON）がセッションに対して強制される場合、Natural アプリケーションはユーザーエンコードを指定し、それをこのセッションが開かれるときに（OP コマンド）Adabas ニュークリアスに伝える必要があります。ADALNK モジュールにより、呼び出し元の設定に応じて Adabas バッファデータが変換されます。メインフレームでは、ICU ではなく Entire Conversion Services（ECS）が変換に使用されます。オープンシステムでは、ICU が Adabas によって使用されます。したがって、NATCONFIG の関連する NTCPAGE エントリに ECS 名が定義されている必要があります。データベースが開かれるときに、Natural ニュークリアスによって OP コマンドが送信されます。メインフレームバージョンの Adabas では、レコードバッファ内のすべての A フィールドの ACODE 設定およびすべての W フィールドの WCODE 設定（UTF-16 を意味する 4095）とともに送信されます。オープンシステムバージョンの Adabas では、レコードバッファ内の WCHARSET 設定とともに送信されます。ACODE/WCODE オプションは、このデータベースの OPRB パラメータに定義されている必要があります。

Adabas 変換の詳細については、OP コマンドの説明および Adabas のメインフレームドキュメントで提供されている UES（ユニバーサルエンコーディングサポート）エンコードの情報を参照してください。

変換テーブル

Natural では、文字変換および文字タイプ定義にさまざまなテーブルが使用されます。テーブルの内容は、Natural セッションの開始時にセッションパラメータ（TAB、UTAB1、UTAB2、および SCTAB）を使用して変更できます。

Natural がコードページサポートを使用して実行されている（つまり、パラメータ CP が "ON"、"AUTO"、またはコードページの名前に設定されている）場合、ユーザーはテーブルを変更できません。この場合、前述のセッションパラメータが考慮されないことをユーザーに通知するために、次の Natural 起動メッセージが発行されます。

CFICU=ON が設定されているため、文字変換パラメータ *table-name* は無視されました。

Natural により、デフォルトコードページ（システム変数 *CODEPAGE の値）の要件に合わせてテーブルは自動的に調整されます。『オペレーション』ドキュメントの「変換テーブル」も参照してください。

マルチバイトコードページのサポート

Natural では、EBCDIC および DBCS に基づく日本語コードページである IBM-939 などのマルチバイトコードページ (MBCS) がサポートされます。マルチバイトコードページは、CP パラメータを使用して選択できます。CP を "AUTO" (サポートされている場合) またはコードページの名前に設定します。Natural がマルチバイトコードページを使用して実行されている場合、Unicode に基づく内部 I/O バッファが使用されます。つまり、I/O ステートメントによって内部 I/O バッファに書き込まれるすべてのデータは、Unicode に変換されます。Unicode およびマルチバイトコードページの要件のために、I/O バッファのサイズは従来の I/O と比較して大きくなります。これは、Unicode 文字には EBCDIC 文字の 2 倍のスペースが必要であり、フィールドを説明するために拡張属性が必要であるためです。

IBM-1140 などのシングルバイトコードページ (SBCS) の場合は、リソースを保持するために従来の EBCDIC ベースの I/O がまだ使用されています。

9 既存アプリケーションの移行

- 既存アプリケーションへの Unicode の影響 62
- Windows、UNIX、および OpenVMS プラットフォームでの既存オブジェクトの移行 62
- メインフレームプラットフォームでの既存オブジェクトの移行 63
- 既存アプリケーションへの Unicode サポートの追加 65
- Natural リモートプロシージャコール (RPC) の移行 66

既存アプリケーションへの Unicode の影響

Windows、UNIX、および OpenVMS プラットフォーム

Windows、UNIX、および OpenVMS プラットフォームでは、Natural は内部的に Unicode 対応です。つまり、文字列を含む数多くの構造は、現在 Unicode フォーマットです。例えば、現在 Natural ソースエリアは Unicode フォーマットです。このため、コードページフォーマットでのみ使用できるデータは、内部的に Unicode フォーマットに変換されます。このことは、例えば、Natural ソースや Natural ライブラリ名およびオブジェクト名に適用されます。ただし、コードページから Unicode への変換、およびその逆の変換は、正しいコードページが変換に使用された場合にのみ正常に実行されます。アプリケーションの変更ではなく再カタログのみが実行される場合でも、カタログのためにオブジェクトが Natural ソースエリアにロードされるため、コードページ情報が重要です。すべてのオブジェクトがシステムコードページでコード化されている場合、変更は必要ありません。オブジェクトがシステムコードページでコード化されていない場合は、追加情報について「[Windows、UNIX、および OpenVMS プラットフォームでの既存オブジェクトの移行](#)」を参照してください。

ほとんどの場合、内部 Unicode 構造は、より多くのメモリを必要とします。プロファイルパラメータ USIZE に小さい値を定義している場合は、この値を大きくする必要がある場合があります。

メインフレームプラットフォーム

既存アプリケーションへの Unicode の影響はありません。内部構造は変更されておらず、A フォーマットフィールドの変換は強制されません。つまり、既存の Natural アプリケーションは何もしなくても実行されます。パラメータ CFICU および CP は "OFF" に設定されている必要があります。ICU ロードモジュールのいずれか (NATICU、NATICUCV、または NATICUXL) をリンクする必要性も、この場合は重複になります。潜在的な Unicode フィールドをサポートするために属性が拡張されているため、I/O バッファのみが大幅に増やされています。CP が "OFF" に設定されると、システム変数 *CODEPAGE はクリアされ、既知の変換テーブル (標準テーブルや代替テーブルなど) の使用は I/O 変換のため継続されます。

Windows、UNIX、および OpenVMS プラットフォームでの既存オブジェクトの移行

Natural は、コードページ情報を複数のレベルで定義できるように拡張されています。

- Natural プロファイルパラメータ CP は、デフォルトの Natural コードページを定義します。
- 一部のオブジェクト (Natural ソース、Natural バッチ入力/出力ファイル、タイプ ASCII、圧縮 ASCII、Unformatted、および CSV のワークファイル) に対して、オブジェクト固有のコードページを定義できます。

オブジェクト固有のコードページもデフォルトコードページも定義されていない場合は、Naturalによってオペレーティングシステムのコードページが使用されます。

正しいコードページを自動的に識別することはできないため、必要なコードページ情報をユーザーが定義することが重要です。次のような状況が考えられます。

ステータス	作業	対処
すべてのデータがオペレーティングシステムのコードページで使用可能。	作業不要	対処不要。
すべてのデータを1つのコードページで保存。ただし、このコードページはオペレーティングシステムのコードページとは異なる。	簡単	Natural プロファイルパラメータ CP を正しいコードページに設定する必要があります。
データは異なるコードページで使用可能。	ソースとコードページの数に依存	すべてのNatural オブジェクトに正しいコードページを定義する必要があります。 <ul style="list-style-type: none"> ■ ソース 影響を受けるオブジェクトが少数のみの場合は、[プロパティ] ダイアログボックスを使用してコードページを変更します。複数のオブジェクト（ライブラリ全体など）が影響を受ける場合は、FTOUCH ユーティリティを使用してコードページを変更します。 ■ バッチファイル Natural プロファイルパラメータ CPOBJIN、CPSYNIN、および CPPRINT を正しいコードページに設定します。 ■ ワークファイル コンフィグレーションユーティリティでワークファイルの正しいコードページを設定します。
1つのオブジェクト（ソースなど）に異なるコードページが混在。	多大	オブジェクトを UTF-8 フォーマットで再書き込みする必要があります。

メインフレームプラットフォームでの既存オブジェクトの移行

Natural は、コードページ情報を複数のレベルで定義できるように拡張されています。

- Natural プロファイルパラメータ CP は、デフォルトの Natural コードページを定義します。
- 一部のオブジェクト（Natural ソース、Natural バッチ入力/出力ファイル、出力レポート、Adabas ファイル）に対して、オブジェクト固有のコードページを定義できます。

既存アプリケーションの移行

オブジェクト固有のコードページもデフォルトコードページも定義されていない場合（CP=OFFが適用されている場合）、Naturalによってデータは変換されません。

正しいコードページを自動的に識別することはできないため、必要なコードページ情報をユーザーが定義することが重要です。次のような状況が考えられます。

ステータス	作業	対処
すべてのデータがオペレーティングシステムのコードページで使用可能。	作業不要	対処不要。
すべてのデータを1つのコードページで保存。ただし、このコードページはオペレーティングシステムのコードページとは異なる。	簡単	Natural プロファイルパラメータ CP を正しいコードページに設定する必要があります。このコードページがI/Oデバイスによってサポートされていることを確認します。CP=AUTOによって、NaturalはI/Oデバイスのコードページを使用して実行されます。
データは異なるコードページで使用可能。	ソースとコードページの数に依存	すべてのNaturalオブジェクトに正しいコードページを定義する必要があります。 <ul style="list-style-type: none"> ■ ソース セッションの各オブジェクトを正しいコードページを使用して保存します。 ■ バッチファイル Natural プロファイルパラメータ CPOBJIN、CPSYNIN、および CPPRINT を正しいコードページに設定します。 ■ Adabas ファイル（ECS 対応） Natural プロファイルパラメータ OPRB を ACODE オプションを使用して設定します。
1つのオブジェクト（ソースなど）に異なるコードページが混在。	多大	オブジェクトを適切なコードページフォーマットで再書き込みする必要があります。

以前の Natural バージョンで保存または STOW されたソースには、コードページ情報がありません。ディレクトリのコードページフィールドは空です。

Natural ソースは Unicode フォーマットで保存されないため、ソースを、セッションに適用されるデフォルトコードページ（システム変数 *CODEPAGE の値）に変換する必要があります。コードページのサポートが無効の場合（CP=OFF）、ソースのコードページ情報は無視され、変換は実行されません。英数字定数は、ソースエリアにロードされるときに、デフォルトコードページに合わせて調整される必要があります。

Natural ソースは Unicode フォーマットで保存されないため、英数字定数を、オブジェクトの起動時にデフォルトコードページに合わせて調整する必要があります。これは、CPAGE コンパイラオプションを使用して実行できます。CPAGE が "ON" に設定されている場合、追加テーブルがオブジェクト内に生成されます。このテーブルは Natural ローダーによって使用され、すべての英数字定数がデフォルトコードページ（システム変数 *CODEPAGE の値）に変換されます。英

数字定数の量に応じて、追加テーブルで結果のオブジェクトのサイズが大きくなり、変換によって追加の CPU 時間が消費されます。

従属オブジェクト（プログラムやプログラムによって使用されるローカルデータエリアなど）によって同じコードページが使用されることが重要です。複数の従属オブジェクトで異なるコードページが使用されている場合、使用されている文字（例えば、"#"）が、使用されているコードページの同じコードポイントにマップされるようにする必要があります。次のオブジェクトおよびデータには、オブジェクト固有のコードページもデータ固有のコードページも関連付けられていません。

- データ定義モジュール（DDM）
- Predict ルール
- Predict XRef データ

このようなデータをオブジェクト固有のコードページが定義されているオブジェクトで使用する場合、またはこのようなオブジェクトにより生成する場合は、注意が必要です。アプリケーション自体をコードページ対応にする必要はないが、処理対象のデータに関してアプリケーションをコードページセンシティブにする必要がある場合は、プロファイルパラメータ SRETAIN を値 "(ON,EXCEPTNEW)" とともに使用することを考慮してください。

既存アプリケーションへの Unicode サポートの追加

Uフォーマットに基づく新しいソースコードで既存アプリケーションを拡張することは、簡単です。（Aフォーマットと比較して）Uフォーマットについては次のルールを考慮する必要があります。

- U以外のフォーマットへのUのREDEFINEを避ける必要があります。文字が分割される場合があるためです。
- Uフォーマットは、エンディアンに依存します。フォーマットBとU間で移動する場合は、これを考慮する必要があります。
- パフォーマンス上の理由から、DEFINE DATAでUを整理します（UNIXおよびOpenVMSでのパフォーマンスの向上）。
- UをAに移動すると文字が失われる場合があることに注意します。

既存フィールドをAフォーマットからUフォーマットに変更する場合は、次のルールを考慮する必要があります。

- 文字列の特定のエンコードを前提とするコードは、変更する必要があります（Bフィールドとの比較など）。
- フィールドのすべてのREDEFINEステートメントについて、その有効性をチェックする必要があります。
- NへのREDEFINEは不可能です。つまり、予期した結果を得られません。

- データベースフィールドは Unicode に移行する必要があります（データベースでサポートされている場合）。
- フィールドの長さを変更する必要がある場合があります。A フィールドに DBCS 文字が含まれている場合、U フィールドには半分の長さが必要です。

Natural リモートプロシージャコール (RPC) の移行

プロファイルパラメータ CP の名前が CPRPC に変更されました。以前の Natural バージョンでは、CP は、トランスポートプロトコル ACI (EntireX Broker) が使用される場合に、Entire Conversion Service (ECS) によって使用され、Natural リモートプロシージャコールにのみ適用されるコードページの名前を指定するために使用されました。

バージョン 6.2 (Windows および UNIX)、バージョン 6.3 (OpenVMS)、およびバージョン 4.2 (メインフレーム) 以降は、Natural データのデフォルトコードページを定義する新しい CP パラメータを使用できます。Natural RPC を使用しており、以前は CP パラメータを動的に使用していた場合は、このパラメータを CPRPC に変更する必要があります。

Windows、UNIX、および OpenVMS プラットフォーム

以前のバージョンのパラメータファイルを使用する場合、何も変更する必要はありません。コンフィグレーションユーティリティによって、CP は CPRPC に自動的に移行されます。

メインフレームプラットフォーム

パラメータ CP は、Natural データのデフォルトコードページの名前を指定するため、またはユーザー端末からコードページ名を自動的に取得するために、ソースモジュール NATCONFIG 内のパラメータマクロ NTCPAGE とともに使用されます。

パラメータ CPRPC は、プロファイルパラメータ RPC および対応するマクロ NTRPC とともに使用されます。

10 特別な考慮事項と制限事項

- Windows、UNIX、および OpenVMS プラットフォーム 68
- メインフレームプラットフォーム 68

Windows、UNIX、および OpenVMS プラットフォーム

- Natural for Windows で提供されるダイアログエディタおよびダイアログベースのランタイムは、Unicode 対応ではありません。
- Natural for UNIX および Natural for OpenVMS で提供されるエディタは、Unicode 対応ではありません。
- 250 文字を超えるフィールドに対して DL パラメータが指定された場合、最大 250 文字がフィールドに表示されます。
- Natural のソース行の長さは、250 バイトを超えることはできません。Unicode フォーマット上で機能するプログラムエディタでは、UTF-16 コード単位の数が 250 を超えないことのみがチェックされます。ただし、ソースのエンコードによっては、エンコードを UTF-16 からソースのエンコードに変換するときに、行の長さが増える場合があります。例えば、UTF-8 エンコードでは、中国語 1 文字に最大で 4 バイトが必要です。この場合、エラーが表示され、変更は保存されません。
- UNIX および OpenVMS の場合、Unicode はランタイムに Natural Web I/O インターフェイスでのみサポートされます。アプリケーションが端末エミュレーションまたは xterm で実行され、Unicode 文字列が表示される場合、不正な結果となる場合があります。
- コードページと Unicode の間でいくつかの変換を実行する必要があるため、以前の Natural バージョンと比較してパフォーマンスは低下しています。

メインフレームプラットフォーム

- Natural for Mainframes で提供されるエディタは、Unicode 対応ではありません。
- I/O バッファのサイズは、Unicode フィールドへの準備のために増やされています。
- 完全な Unicode I/O は、ランタイムに Natural Web I/O インターフェイスでのみサポートされます。アプリケーションが端末エミュレーションで実行され、Unicode 文字列が表示される場合、一部の Unicode 文字が正しく表示されない場合があります。

11 双方向言語サポート

アラビア語やヘブライ語などの一部の言語は右から左 (RTL) に記述されますが、英語やドイツ語などの大部分の言語は左から右 (LTR) に記述されます。左から右と右から左の両方が含まれるテキストは、双方向テキストと呼ばれます。

Natural は、双方向言語を基本的にサポートします。Windows では、このサポートは、Natural デフォルトコードページと Windows システムコードページの両方が双方向コードページとして定義されている場合に有効です。Natural で特定のコードページが定義されていない場合は、双方向の Windows システムコードページが定義されていれば十分です。UNIX および OpenVMS では、双方向言語のサポートは、Natural デフォルトコードページが双方向コードページである場合に有効です。メインフレームでは、双方向言語のサポートは自動的にアクティブはなりません。このため、ユーザーは常に、以下の説明に従ってすべての必要なパラメータを (例えば、PM=I のように) 指定する必要があります。

Natural プログラムの出力は、プロファイルパラメータ PM、端末コマンド %V、およびセッションパラメータ PM を使用して制御できます。

メインフレーム、UNIX、および OpenVMS では、プロファイルパラメータ D0 (表示順序) を追加で使用して、元々は双方向データではなくインバース (右から左) 出力モードをサポートする端末用に作成されているアプリケーションをサポートすることができます。これらのアプリケーションにより、アプリケーションコード内の双方向データの表示順序が作成されます。パラメータ D0 を使用すると、これらのアプリケーションは、双方向データをサポートする I/O デバイスでも動作できるようになります。このことは、例えば、Natural Web I/O インターフェイスを使用してブラウザで実行されるアプリケーションなどが当てはまります。

プロファイルパラメータ PM は、デフォルトの画面方向を定義します。PM が "R" (リセット) に設定されている場合、デフォルトの画面方向は左から右です。PM が "I" (逆) に設定されている場合、デフォルトの画面方向は右から左です。英数字以外のフィールド、システム変数、および PF キー行 (メインフレームのみ) はいずれも、Natural によって自動的にインバートされるため、画面方向が右から左であれば、右から左に正しく表示されます。UNIX および OpenVMS では、PF キー行はインバートされず、常に左から右へ表示されます。

端末コマンド %V は、画面方向を変更するために使用できます。画面方向が右から左の場合、現在のウィンドウのレイアウトはミラーリングされます。つまり、すべてのウィンドウコンポーネントまたはフィールドの基点は右上隅になります。画面方向は、%VON を使用して右から左に変更され、%VOFF を使用して左から右に戻されます。

セッションパラメータ PM では、フィールドの方向が反対になります。「フィールドの方向を反対にした場合」の結果は、PM パラメータが使用されるステートメントおよびプラットフォームによって異なります。PM パラメータが MOVE ステートメントで使用された場合、フィールドの内容は単純に反対になります。つまり、最初の文字が最後の文字になるなどです。結果はフィールドの文字に依存しません。末尾の空白が削除されてから、フィールドは反対にされます。

例えば、次のようなプログラムがあります。

```
DEFINE DATA LOCAL
1 TEST1 (A10)
1 TEST2 (A10)
END-DEFINE
TEST1 := 'program'

MOVE TEST1 (PM=I) TO TEST2
INPUT TEST1 (AD=0) TEST2 (AD=0)

END
```

次の出力を生成します。

```
TEST1 program TEST2 margorp
```

"margorp" は "program" が反対になったものです。

PM パラメータが INPUT や DISPLAY などの IO ステートメントに対して使用される場合、その結果はより複雑です。この場合、フィールド方向は画面方向に基づきます。

- 画面方向が左から右であり、PM=I がフィールドに適用される場合、フィールド方向は右から左に変わります。
- 画面方向が右から左であり、PM=I がフィールドに適用される場合、フィールド方向は左から右に変わります。

Windows およびブラウザ端末 (Natural Web I/O インターフェイス) では、「フィールド方向を反対にすること」は、フィールドの文字が単純に反対になることを意味しません。この場合、複雑な双方向アルゴリズムが適用されます。詳細については、Microsoft Windows のドキュメントを参照してください。ただし、(文字型の) 他の端末では、フィールドの文字は再ソートされません。単純に反対になるだけです。

次の例では、変数 TEST に割り当てられた文字は、次の順序で入力されています。

a b c ㄱ ㄴ ㄷ 1 2 3

次に、Windows のプログラム例を示します。プログラムエディタで入力するときに、定数の文字はすでに再ソートされています。

```
DEFINE DATA LOCAL
1 TEST (A20)
END-DEFINE
TEST := 'abc 123 ㄱㄴㄷ'

SET CONTROL 'voff'

INPUT TEST (AD=0) /
      TEST (AD=0 PM=I)

SET CONTROL 'von'

INPUT TEST (AD=0) /
      TEST (AD=0 PM=I)
END
```

このプログラムでは、Windows 上に次の 2 つの画面が生成されます。

```
TEST abc 123 ㄱㄴㄷ
TEST          123 ㄱㄴㄷ abc
```

および

```
          123 ㄱㄴㄷ abc TEST
abc 123 ㄱㄴㄷ          TEST
```

次に、UNIX、OpenVMS、およびメインフレームのプログラム例を示します。文字が同じ順序で入力された場合、文字は単純にキーイング順に表示されるため、プログラムは次のように表示されます。

```
DEFINE DATA LOCAL
1 TEST (A20)
END-DEFINE
TEST := 'abc ㄱㄴㄷ 123'

SET CONTROL 'voff'

INPUT TEST (AD=0) /
      TEST (AD=0 PM=I)
```

双方向言語サポート

```
SET CONTROL 'von'  
  
INPUT TEST (AD=0) /  
      TEST (AD=0 PM=I)  
END
```

UNIX および OpenVMS では、このプログラムにより次の2つの画面が生成されます。

```
TEST abc ㄩㄨㄩ 123  
TEST      321 ㄩㄨㄩ cba
```

および

```
321 ㄩㄨㄩ cba TSET  
abc ㄩㄨㄩ 123      TSET
```

メインフレームでは、このプログラムにより2つの同じ画面が生成されます（メインフレームでは、ステートメント SET CONTROL 'voff' と SET CONTROL 'von' は英数字フィールドには適用されません）。どちらの画面も、次のように表示されます。

```
TEST abc ㄩㄨㄩ 123  
TEST      321 ㄩㄨㄩ cba
```

Windows、UNIX、および OpenVMS では、マップエディタには、**[Reverse Map]** コマンドがあり、双方向フィールドを持つマップを簡単に処理できます。このコマンドによって、現在のマップの表示方向が変更されます。フィールドの位置は変わらず、表示のみが変更されます。Windows では、このコマンドは現在のマップにのみ影響します。UNIX および OpenVMS では、以降のすべてのマップが反対に表示されるように、フラグが設定されます。次の **[Reverse Map]** コマンドによって、元の状況が復元されます。

Windows では、ダイアログの出力を同様に制御できます。ダイアログ自体とほとんどのダイアログコントロールの両方に RTL 属性があります。ダイアログの RTL 属性をオンにすると、ダイアログの画面方向は右から左になります。他のコントロールの RTL 属性をオンにすると、これらのコントロールの方向は右から左になります。

プロファイルパラメータ PM によって、新しいダイアログの RTL 属性のデフォルト設定が定義されます。PM が "R"（リセット）に設定されている場合、RTL 属性はデフォルトでオンになりません。PM が "I"（逆）に設定されている場合、RTL 属性はデフォルトでオンになります。ダイアログの新しく作成されるコントロールの RTL 属性のデフォルト設定は、ダイアログの RTL 属性の設定から派生します。

ダイアログにすでにコントロールがあるときにダイアログの RTL 属性が変更された場合は、コントロールの RTL 属性も変更するかどうかを確認するダイアログが表示されます。

Windows で双方向言語を使用している場合は、"GUI" が推奨される出力方法です。出力方法 "GUI" を使用すると、出力されるページに、画面上に表示されるウィンドウと同じレイアウトが表示されます。フィールド文字のソートは同一です。出力方法 "TTY" を使用すると、ほとんど

の場合、出力されるレイアウトは画面ウィンドウのレイアウトとは異なります。これは、フィールド文字が論理順に出力されるためです。方向が右から左へのフィールドの場合、すべての文字が単純に反対になります。つまり、最初の文字が最後の文字になるなどです。

12 ダブルバイト文字サポート

ほとんどの東アジア言語では、コードページ文字列内の言語固有の文字（Natural フォーマット A）は2バイト（いわゆるダブルバイト文字セット）で表され、ASCII 文字（メインフレーム上の EBCDIC）は1バイトで表されます。そのため、コードページ文字列は、1バイトと2バイトという異なる長さの文字で構成されます。

Natural は、ダブルバイト文字セットを基本的にサポートします。Windows では、このサポートは、Natural デフォルトコードページと Windows システムコードページの両方がダブルバイトコードページとして定義されている場合に有効です。Natural で特定のコードページが定義されていない場合は、ダブルバイトの Windows システムコードページが定義されていれば十分です。UNIX および OpenVMS では、ダブルバイト文字セットのサポートは、Natural デフォルトコードページがダブルバイトコードページである場合に有効です。メインフレームでは、プロファイルパラメータ CP が EBCDIC MBCS コードページ（IBM-942 など）に設定されている必要があります。

ダブルバイト文字セットのサポートが有効な場合、Natural では、すべての文字列操作について、ダブルバイト文字セットが1つのユニットとして処理されることが保証されます。このことは、文字列の意味を保持するために不可欠です。

フォーマット A の変数の操作の後（例えば、SUBSTRING オプションを使用してサブストリングを抽出した後）でダブルバイト文字セットの先頭または末尾の1バイトが残された場合、このバイトは空白文字で置き換えられます。

次の例では、コードページ Shift_JIS が選択されています。変数 #A には、4文字で構成された文字列が含まれています。最初と最後の文字は、コードページ Shift_JIS でバイトシーケンス H'8282' によって表されるダブルバイト文字セット "FULL WIDTH LATIN SMALL LETTER B" です。2番目と3番目の文字は、1バイト H'61' によって表されるシングルバイト文字 "LATIN SMALL LETTER A" です。したがって、文字列全体の16進表示は H'828261618282' です。

ダブルバイト文字サポート

```
DEFINE DATA LOCAL
  1  #A  (A10)
END-DEFINE

#A := ' b aa b '

WRITE #A #A (EM=H(6))
EXAMINE #A FOR PATTERN ' B ' REPLACE 'a'
WRITE #A #A (EM=H(6))

END
```

ダブルバイト文字セットがサポートされない場合、上記のプログラムの出力は次のとおりです。

```
Page          1                      07-02-07    17:22:09

b aa b      828261618282
B a b      826161828220
```

これは、文字 "b" (コードページ Shift_JIS の H'8282') が 1 つの単位として扱われなかった結果です。この文字の末尾のバイトおよびその次の文字 "a" (H'61') が、誤ってダブルバイト文字セット "B" (コードページ Shift_JIS の H'8261') として解釈されています。

ダブルバイト文字セットがサポートされる場合、プログラムの出力は予期したとおりになります。

```
Page          1                      07-02-07    17:22:09

b aa b      828261618282
b aa b      828261618282
```


13 よくある質問

-
- 起動エラー「Invalid code page specified」が表示されるのはなぜですか。 78
 - "デフォルトコードページ"とは何ですか。 78
 - どのデフォルトコードページが使用されていますか。 78
 - すべての Natural ソースを UTF-8 フォーマットで保存する必要がありますか。 78
 - どうすれば Natural コードで UTF-8 エンコードを処理できますか。 79
 - 一部の文字が正しく表示されないのはなぜですか。 79
 - Natural ソースを編集するときにエラーが発生するのはなぜですか。 79
 - Natural ソースを保存するときにエラーが発生するのはなぜですか。 79
 - Natural ソースのエンコードはどうすればわかりますか。 80
 - Natural ソースのエンコードはどうすれば変更できますか。 80
 - どうすれば既存の Natural ソースを UTF-8 フォーマットに変換できますか (Windows、UNIX、および OpenVMS のみ) 。 80
 - 文字を変換できない場合、どの置換文字が使用されますか。 81
 - 以前の Natural バージョンで Natural 4.2 ソースを使用できますか。 81
 - 以前の Natural バージョンで UTF-8 ソースを使用できますか。 81
 - UTF-8 フォーマットのソースをカタログするときに変換エラーが発生するのはなぜですか。 81
 - 端末エミュレーションを経由して U フォーマットを表示するときに、UNIX または OpenVMS でガーベッジが表示されるのはなぜですか。 82
 - 現在の SPoD クライアントと古い SPoD サーバーを使用できますか。 82
 - 現在の SPoD サーバーと古い SPoD クライアントを使用できますか。 82

起動エラー「Invalid code page specified」が表示されるのはなぜですか。

プロファイルパラメータ CP で定義したコードページが、存在しないか（有効な ICU コードページについては <http://demo.icu-project.org/icu-bin/convexp>、適切な IANA 名については <http://www.iana.org/assignments/character-sets> を参照）、またはプラットフォームで無効なデフォルトコードページです（例えば、EBCDIC コードページは Windows、UNIX、または OpenVMS プラットフォームでは使用できません）。

メインフレームプラットフォームでは、コードページは NTCPAGE によって Natural コンフィグレーションファイルに指定されている必要があります（「[NTCPAGE マクロ](#)」も参照）。ここに入力されていないコードページは、ICU 実装で使用可能であっても、無効として拒否されます。NATCONFIG で指定されているのと同じ IANA 名、CCSID/CCSN、またはエイリアス名が使用されているかどうかを確認してください。

"デフォルトコードページ" とは何ですか。

デフォルトコードページとは、プロファイルパラメータ CP の評価の結果のコードページです。CP が入力されていない場合（Windows、UNIX、および OpenVMS）、現在のオペレーティングシステムコードページがデフォルトコードページとなります。

どのデフォルトコードページが使用されていますか。

コードページから Unicode への変換、およびその逆の変換のために Natural によって使用されているデフォルトコードページは、システム変数 *CODEPAGE の内容を表示することによって知ることができます。

すべての Natural ソースを UTF-8 フォーマットで保存する必要がありますか。

使用する文字およびソースを保存するプラットフォームによります。Unicode 定数を使用する場合、文字のすべての組み合わせを保存できるのは UTF-8 のみです。ただし、16 進 UH 定数を定義して、それをコードページソースに保存することもできます。16 進定数の短所は、定数のすべての文字の UTF-16 エンコードを知る必要があることです。メインフレームでは、ソースの UTF-8 フォーマットは使用できません。UNIX および OpenVMS では、UTF-8 ソースは SPoD 経由でのみ処理できます。UNIX または OpenVMS 上でローカルに処理することはできません。

どうすれば Natural コードで UTF-8 エンコードを処理できますか。

UTF-8 から UTF-16 への変換のために `MOVE ENCODED` ステートメントを使用します。A フォーマット変数に対してコードページ "UTF-8" を使用する必要があります。

一部の文字が正しく表示されないのはなぜですか。

正しいコードページを使用しているかどうかを確認してください。コードページが正しい場合は、選択したフォントによって、表示する文字がサポートされているかどうかを確認してください。

Natural ソースを編集するときにエラーが発生するのはなぜですか。

ソースに対して定義されているコードページが正しくありません。ソースの内容を Unicode に変換するときに、変換エラーが発生します。Unicode に正常に変換されるように、ソースのエンコードを変更してください。

メインフレームプラットフォームでは、ソースは作成時のコードページで保存されます。保存されたソースのコードページから現在の Natural セッションのコードページにソースが変換されなかった場合、変換エラーが発生します。変換を避けるために、ソースのコードページを使用して Natural を開始できます。または、エディタが開始されたときに表示されるウィンドウで、変換不可能な文字を調整できます。

Natural ソースを保存するときにエラーが発生するのはなぜですか。

ソースの読み取りに使用されたコードページに変換されない文字をソースに入力しました。これらの文字を誤って入力したか、本当にソースに保存するかを確認してください。最初の場合には、誤った文字を削除してソースを保存します。2つ目の場合には、ソースを UTF-8 フォーマットで保存するか、または文字が U 定数に含まれている場合は、代わりに UH 定数を使用します。

ソースのコードページに含まれていない文字を入力していない場合は、プロファイルパラメータ `SRETAIN` が "OFF" に設定されているかどうかを確認します。この設定の場合、ソースはデフォルトコードページで保存されます。当該のソースが以前に別のコードページで保存された場合は、変換エラーが発生する場合があります。

SPoD 経由でメインフレーム環境に接続している場合、メインフレームのソースは、SPoD 環境では変換されて Unicode で編集されます。保存される場合は、Natural サーバーのコードペー

ジに変換される必要があります。Unicode 文字が Natural サーバーセッションのコードページでマップされない場合は、変換エラーが発生する場合があります。

SPoD を使用しないネイティブ Natural for Mainframes 環境の場合、変換は実行されないため、ソースを保存するときにエラーは発生しません。ソースは、現在の Natural セッションのコードページ情報を使用して保存されます。

Natural ソースのエンコードはどうすればわかりますか。

Natural スタジオでは、ソースノードの [プロパティ] ダイアログボックスを呼び出します。 [一般] ページに、ソースのエンコードが表示されます。 [エンコード] テキストボックスが空の場合、ソースに対して特定のエンコードは保存されていません。つまり、ソースを読み取るときにデフォルトのエンコードが使用されます。

Natural スタジオのリストビューウィンドウにも、リストされているすべてのオブジェクトのエンコードが表示されます。

メインフレームプラットフォームでは、コードページ情報は Natural ソースディレクトリの一部です。Natural for Mainframes で LIST DIR コマンドを使用して、ディレクトリを表示します。

Natural ソースのエンコードはどうすれば変更できますか。

Natural スタジオでは、ソースノードの [プロパティ] ダイアログボックスを呼び出します。 [一般] ページに、ソースのエンコードが表示されます。これが正しいエンコードではない場合は、 [変更] ボタンを選択して変更できます。使用できるコードページのリストが表示され、ソースに対して正しいエンコードを選択できます。

メインフレームプラットフォームでは、CP パラメータを使用して、目的のコードページで Natural セッションを開始する必要があります。パラメータ SRETAIN を "OFF" に設定し、ソースを編集して保存します。これで、ソースのコードページ情報は変更されます。または、SYSCP ユーティリティを使用して、ソースのコードページ割り当てを確認または変更することもできます。

どうすれば既存の Natural ソースを UTF-8 フォーマットに変換できますか (Windows、UNIX、および OpenVMS のみ) 。

正しいコードページを使用して Natural エディタでソースを開きます。 [名前をつけて保存] でソースを保存し、 [名前をつけて保存] ダイアログボックスでエンコードとして UTF-8 を選択します。

文字を変換できない場合、どの置換文字が使用されますか。

変換の方向によって異なります。コードページの文字をUnicodeに変換できない場合は、Unicode置換文字 "U+FFFD" が使用されます。Unicodeの文字をコードページに変換できない場合は、このコードページに対してICUによって定義されている置換文字が使用されます。

Natural for Mainframes では、コードページの置換文字または、コンフィグレーションファイルで指定されている場合は、プレースホルダ文字が使用されます。

Unicodeからデフォルトコードページへの変換の場合、Windows、UNIX、およびOpenVMSプラットフォームでは、プロファイルパラメータ SUBCHAR を設定することによって、置換文字を変更できます。

以前の Natural バージョンで Natural 4.2 ソースを使用できますか。

Natural for Mainframes では、以前の Natural バージョンで Natural 4.2 ソースを使用できます。ソースのレイアウトは変更されておらず、以前のバージョンでソースにアクセスした場合、Natural 4.2 ソースの追加のコードページ情報は単純に無視されます。

以前の Natural バージョンで UTF-8 ソースを使用できますか。

できません。以前の Natural バージョンでは、コードページ情報は認識されません。UTF-8 ソースは、現在のシステムコードページとして解釈されます。

UTF-8 フォーマットのソースをカタログするときに変換エラーが発生するのはなぜですか。

コードポイントを変換できないため、UTF-8 フォーマットの Natural ソースはカタログできません (Windows、UNIX、および OpenVMS のみ)。

UTF-8 フォーマットのソース内のすべての A 定数は、生成されるプログラムに保存するときに、デフォルトコードページに変換されます。デフォルトコードページに含まれない文字を A 定数から削除するか、A 定数の代わりに U 定数を使用してください。

端末エミュレーションを経由してUフォーマットを表示するときに、UNIXまたはOpenVMSでガーベッジが表示されるのはなぜですか。

端末エミュレーションに出力が表示される前に、デフォルトコードページに含まれないすべての文字は、コードページの置換文字で置き換えられます。ASCIIコードページの場合、ICU変換テーブルによって定義される置換文字が"0x1A"である場合があります。これは、UNIXまたはOpenVMS端末で制御文字である場合があります。I/OステートメントでUフォーマットを使用する場合は、Natural Web I/O インターフェイスを使用することを強くお勧めします。端末エミュレーションの使用が不可欠な場合は、置換文字 (SUBCHAR) を出力可能な文字 ("?" など) に変更できます。

メインフレームプラットフォームでは、端末エミュレーションを使用できます。NCPAGEマクロで、表示可能なプレースホルダ文字によって置換文字を置き換えることができます。プレースホルダ文字によって、変換不可能な文字があった場合にガーベッジを回避できます。

現在のSPoDクライアントと古いSPoDサーバーを使用できますか。

はい。ただし、SPoDクライアントのコードページをサーバーソースのコードページに設定する必要があります。

「Prerequisites for Natural Single Point of Development」

(http://documentation.softwareag.com/natural/spod_prereq/prereq.htm) を参照してください。

現在のSPoDサーバーと古いSPoDクライアントを使用できますか。

はい。ただし、ソースのエンコードを定義している場合は、お勧めしません。

「Prerequisites for Natural Single Point of Development」

(http://documentation.softwareag.com/natural/spod_prereq/prereq.htm) を参照してください。

索引

U

Unicode サポート, 1

こ

コードページのサポート, 1

