



CNLS の使用によるカスタマイズした各国語変換の設定

Version 5.1

2006-03-14

CNLS の使用によるカスタマイズした各国語変換の設定

Caché Version 5.1 2006-03-14

Copyright © 2006 InterSystems Corporation.

All rights reserved.

このドキュメントは、Sun Microsystems、RenderX Inc.、アドビ システムズ および ワールドワイド・ウェブ・コンソーシアム (www.w3c.org) のツールと情報を使用して、Adobe Portable Document Format (PDF) で作成およびフォーマットされました。主要ドキュメント開発ツールは、InterSystemsが構築したCaché と Javaを使用した特別目的のXML処理アプリケーションです。



Caché 製品とロゴは InterSystems Corporation の登録商標です。



Ensemble 製品とロゴは InterSystems Corporation の登録商標です。



InterSystems という名前とロゴは InterSystems Corporation の登録商標です

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

Caché および InterSystems Caché、Caché SQL、Caché ObjectScript および Caché Object は、インターシステムズ社の商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems ワールドワイド カスタマサポート

Tel: +1 617 621-0700

Fax: +1 617 374-9391

Email: support@InterSystems.com

目次

CNLS の使用によるカスタマイズした各国語変換の設定.....	1
1 NLS 変換	1
2 追記事項	3
2.1 全般的な機能に対するエントリ・ポイント	5
2.2 モーダル変換に対するエントリ・ポイント	8
2.3 I/O 変換の置換属性管理に対するエントリ・ポイント	8

CNLS の使用によるカスタマイズした各国語変換の設定

このドキュメントでは、CNLS ユーティリティ (**cnls.exe**) を使用してカスタマイズされた言語変換の設定、テスト、およびアプリケーションへの適用方法を説明します。

1 NLS 変換

以下は、カスタマイズされた NLS 変換を設定する手順です。

1. **CacheSys¥Bin** ディレクトリの **cnls.exe** アプリケーション (アルファベットのブロック・アイコン) をダブルクリックして、NLS ユーティリティを起動します。
2. [サーバ接続の選択] ウィンドウで、既定の LOCALTCP を選択するか、プルダウン・リストから他のサーバを選択します。[OK] をクリックします。
3. [Caché NLS] ウィンドウが表示されます。[ロケール] タブを選択します。[ロケール] プルダウン・リストを使用して、既存のロケールを表示します。新しく作成するロケールに最も関連性の高いロケールを選択します。右下にある [新規作成] をクリックして、新しいロケールを作成する手順を開始します。

[NLS ロケールを追加] ウィンドウが表示されます。

- ・ 説明：新しいロケールを説明するテキストを入力します。
- ・ コピー元：プルダウン・リストから、テンプレートとして使用する既存ロケールを選択します。例えば、“English, United States, Latin1 (ISO 8859-1)” などを選択します。
- ・ 文字セット：プルダウン・リストから文字セットを選択するか、文字セットを入力します。例えば、“Latin1” などを指定します。既定値は、コピー元のロケールの文字セットです。
- ・ ロケール名：4 文字のコードを指定します。カスタマイズされたロケール名は常に “y” で開始し、8 ビットのアプリケーションでは “8”、Unicode アプリケーションでは “W” で終了します。

これらのすべての指定が完了したら、右下にある [OK] ボタンが使用可能になります。[OK] をクリックします。

新しいロケールの [Caché NLS] ウィンドウが表示されます。ウィンドウ左上の [オプション] メニューを選択し、[ロケールのインストール] をクリックします。別の方法として、NLS の [ロケール

のインポート] オプションを使用して、他の Caché システムからロケールをインポートすることもできます。

メモ：現在のロケールとして他の標準ロケールを選択した場合、NLS ユーティリティの [ロケールのインポート] オプションを使用して、Caché インストール CD からロケール定義を

Cachesys¥Bin ディレクトリにコピーする必要があります。標準ロケールは、**¥nt¥install¥eightbit**、または **¥nt¥install¥unicode** ディレクトリに存在します。ロケール定義は .goq ファイルです。ロケールは、4 文字のロケール名で識別されます。ロケールの **_src.goq** (ソース・コード) バージョンは必ずインポートしてください。例えば、“English, United States, Latin1 (ISO 8859-1)” は **enu8_src.goq** です。[ロケールのインストール] オプションでも、ディスクから上記のファイルをインポートするよう指示されるため、**CacheSys¥Bin** ディレクトリまたは NLS サブディレクトリを選択します。

4. [変換] タブで、右下の [新規作成] を選択して変換テーブルを設定します。[NLS トランスレーション・テーブルを追加] ダイアログ・ボックスの “コピー元” は空欄です。印刷用には、“出力専用” テーブルを使用します。ここでは [テーブル名] の左側は “Latin 1”、[テーブル名] の右側は常に “y” で開始する新規の変換テーブルです。ターミナル入力に対する変換の場合は、“入力専用” テーブルになり、“テーブル名 (From-to)” の左側が新規の変換テーブル、“テーブル名 (From-to)” の右側が “Latin 1” になります。[マッピング・タイプ] は最初は [Single to Single] です。1 つに変換したい複数の文字がある場合、または複数に変換したい複数の文字がある場合は [Multi to Single] または [Multi to Multi] テーブルです。文字は、一番右側のテーブルで変換されます。既定で、10 進数の ASCII 値で表示されます。テーブルは 6 進法、または文字モードで表示されます。これは、左上隅の [ヘルプ] オプションで調整できます。独自の変換を設定する場合は、行の値を変更するか、新規の値を * の隣に追加します。
5. 変換テーブルが一度設定されたら、[ロケール] タブに戻り、新規の変換テーブルで更新します。[トランスレーション名] (真中の欄) で、独自の変換を追加するために * を選択します。ここで “sueprint” など、自由に呼び出しを行うことができます。次に [出力] タブ、または [入力] タブに移動します。ドロップ・ダウン・リストに、ユーザ独自の変換テーブルが表示されるはずなので、これを選択します。ユーザが変換するものによって、[入力] 列と [出力] 列のいずれかで、プレースホルダとして常に既定のテーブルを持つ必要があります。トランスレーション名は、入出力変換を指定するために Caché で使用される名前なので重要です。各トランスレーション名は、出力文字セットと入力文字セットに関連しています。
6. [変換タイプ] (一番下の欄) に移動し、[プログラマ・ウィンドウ] を新規の “トランスレーション名” に、つまり “sueprint” に変更し (真中の欄で指定した名前)、変換をテストします。“トランスレーション名” は、入出力変換を指定するために Caché で使用される名前です。各トランスレーション名は、出力文字セットと入力文字セットに関連しています。
7. 再度、新規変換でロケールをインストール、または検証します。テーブルに対するソースのコンパイルと、共有メモリへのオブジェクトのロードを検証します。ロケールの検証とロケールの更新をインストールします。
8. ターミナル・ウィンドウを新規に開始します。

w \$zm

上記は、ロケールを更新したトランスレーション名をリストにします。

```
>w $zm
>RY\Latin1\K\SUEPRINT
```

この場合、トランスレーション名は "sueprint" です。

2 追記事項

プリンタに変換を設定するには、デバイスがオープンされているときに変換を設定します。

```
u x s zzz=$$SetIO^%NLS("sueprint")
```

以下の構文を使用してテストを行います。

```
>s x="|PRN|"
>o x
>u x s zzz=$$SetIO^%NLS("sueprint") w $C(146),!,$C(147),!
>c x
```

\$ZCVT 関数により、以下の構文を使用して変換をテストすることができます。"ySue" は出力変換テーブルです。

```
>f I=0:1:255 s x=$ZCVT($C(i),"o","ySue") if x'=$c(i) w I," ",$a(x),!
>177 49
 178 50
 180 52
 183 55
 184 56
 231 103
```

これらは、変換された文字です。

Do Dump^%NLSMISC コマンドは、%NLS 設定を一覧表示します。このコマンドの出力例を次に示します。

I/O Table Name	Offset:Type	Source Global
UnicodeLittleIn	4A0030:Struct8	
Out	4A0038:Struct8	
UnicodeBig In	4A0020:Struct8	
Out	4A0028:Struct8	
SAME In	4A0000:Struct8	
Out	4A0008:Struct8	
UTF8 In	4A0AAC:Struct8	%nls("Src","XLT","UTF8","Latin1")
Out	4A0AB8:Struct8	%nls("Src","XLT","Latin1","UTF8")
BIN In	4A0000:Struct8	
Out	4A0008:Struct8	
RAW In	4A0000:Struct8	
Out	4A0008:Struct8	
Ricky Out	4A0098:Struct8	%nls("Src","XLT","Latin1","yAquis")

追記事項

\$X/\$Y Table Name	Offset:Type	Source Global
Latin1	4A0074:Struct8	
Collation Table	Offset:Type	Source Global
0	Enc Dec	Old ANSI
1	Enc Dec	New ANSI
2	Enc Dec	
3	Enc Dec	
4	Enc Dec	
5	Enc Dec	Unicode
128	Enc Dec	Old string
129	Enc Dec	New string
133	Enc Dec	Unicode string
Default	Offset:Type	Source Global
Pattern	4A0040:Struct8	
Ident	4A004C:Struct8	
UpperCase	4A0080:Struct8	
LowerCase	4A008C:Struct8	
TitleCase	4A0080:Struct8	
I/O Default	Offset:Type	Name
Process	490233:ByteStr	Ricky
Direct Term	490233:ByteStr	Ricky
Telnet/LAT	490002:ByteStr	RAW
File	490002:ByteStr	RAW
Magtape	490002:ByteStr	RAW
Network	490002:ByteStr	RAW
DSM-DDP	490002:ByteStr	RAW
DTM-DCP	490002:ByteStr	RAW
SysCalls	490002:ByteStr	RAW
\$X/\$Y Default	Offset:Type	Name
Process	49002D:ByteStr	Latin1
I/O Slots	All Empty	
\$X/\$Y Slots	All Empty	
Process	Offset:Type	Source Global
Pattern	4A0040:Struct8	
Ident	4A004C:Struct8	
UpperCase	4A0080:Struct8	
LowerCase	4A008C:Struct8	
TitleCase	4A0080:Struct8	

Process I/O	Offset:Type	Name
Process	490233:ByteStr	Ricky
Direct Term	490233:ByteStr	Ricky
Telnet/LAT	490002:ByteStr	RAW
File	490002:ByteStr	RAW
Magtape	490002:ByteStr	RAW
Network	490002:ByteStr	RAW
DSM-DDP	490002:ByteStr	RAW
DTM-DCP	490002:ByteStr	RAW
SysCalls	490002:ByteStr	RAW
Process \$X/\$Y	Offset:Type	Name
Process	49002D:ByteStr	Latin1

以下を使用して、ロケール設定を確認します。

```
>d ^%G
>Global ^%SYS("LOCALE","CURRENT") -- NOTE: translation in effect
>^%SYS("LOCALE","CURRENT")=yaq8
```

2.1 全般的な機能に対するエントリー・ポイント

以下は、`%NLS` のエントリー・ポイントの一覧です。各エントリー・ポイントは `EntryPoint(Arg)` の形式で記載され、その呼び出しに使用する完全名は `$$EntryPoint^%NLS(Arg)` です。

- ・ `DefCOL` – システムの既定を返します。
- ・ `DefIO[type]` – システムの既定を返します。type の有効値は、プロセスは 0、M ターミナルは 1 (既定の type)、その他のターミナルは 2、シーケンシャル・ファイルは 3、磁気テープは 4、ネットワーク・デバイスは 5 です。
- ・ `Defldent` – システムの既定を返します。
- ・ `DefLower` – システムの既定を返します。
- ・ `DefPM` – システムの既定を返します。
- ・ `DefTitle` – システムの既定を返します。
- ・ `DefUpper` – システムの既定を返します。
- ・ `DefXY` – システムの既定を返します。
- ・ `Def("C")` – 照合テーブルの名前。
- ・ `Def("E")` – 識別子テーブルの名前。
- ・ `Def("I"[,type])` – I/O 変換テーブルの名前 (type の説明は `DefIO` を参照)。
- ・ `Def("L")` – 小文字変換テーブルの名前。
- ・ `Def("P")` – パターン・マッチ・テーブルの名前。
- ・ `Def("T")` – タイトルの大文字/小文字変換テーブルの名前。

- ・ Def("U") – 大文字変換テーブルの名前。
- ・ Def("X") – \$X/\$Y アクション・テーブルの名前。
- ・ GetCOL – 照合を返します。
- ・ GetCursorFlag – SetCursorFlag の現在の値を取得します。
- ・ GetIO – I/O 変換を返します。
- ・ GetIdent – 識別子を返します。
- ・ GetLower – 小文字変換を返します。
- ・ GetPDefIO[[type]] – プロセスの既定を返します。
- ・ GetPDefXY – プロセスの既定を返します。
- ・ GetPDef("I"[,type]) – I/O 変換テーブルの名前 (type の説明は DefIO を参照)。
- ・ GetPDef("X") – \$X/\$Y アクション・テーブルの名前。
- ・ GetPIO – I/O 変換を返します。
- ・ GetPM – パターン・マッチを返します。
- ・ GetPitch – 現行デバイスのピッチ値を取得します。
- ・ GetTitle – タイトルの大文字/小文字変換を返します。
- ・ GetUpper – 大文字変換を返します。
- ・ GetXY – \$X/\$Y アクションを返します。
- ・ Get("C") – 呼び出しプロセスのテーブル名を指定します。
- ・ Get("E") – 呼び出しプロセスのテーブル名を指定します。
- ・ Get("I") – 現行デバイスのテーブル名を指定します。
- ・ Get("J") – 呼び出しプロセスのテーブル名を指定します。
- ・ Get("L") – 呼び出しプロセスのテーブル名を指定します。
- ・ Get("P") – 呼び出しプロセスのテーブル名を指定します。
- ・ Get("T") – 呼び出しプロセスのテーブル名を指定します。
- ・ Get("U") – 呼び出しプロセスのテーブル名を指定します。
- ・ Get("X") – 現行デバイスのテーブル名を指定します。
- ・ OffIO – 現行デバイスの I/O 変換をオフに設定し、直前の状態を返します。
- ・ OffPIO – 呼び出しプロセスの I/O 変換をオフに設定し、直前の状態を返します。
- ・ OnIO – 現行デバイスの I/O 変換をオンに設定し、直前の状態を返します。
- ・ OnPIO – 呼び出しプロセスの I/O 変換をオンに設定し、直前の状態を返します。

- ・ SetCOL(tblname) – 照合を選択します。
- ・ SetCursorFlag(0) – 現行のデバイスがターミナル上の論理設定と互換性を持つように設定します。
- ・ SetCursorFlag(1) – 現行のデバイスがターミナル上の物理設定と互換性を持つように設定します。
- ・ SetIO(tblname) – I/O 変換を選択します。
- ・ SetIO(tblname) – I/O 変換を選択します。
- ・ SetIdent(tblname) – 識別子を選択します。
- ・ SetLower(tblname) – 小文字変換を選択します。
- ・ SetPDefIO(tblname[,type]) – プロセスの既定を選択します。
- ・ SetPDefXY(tblname) – プロセスの既定を選択します。
- ・ SetPDef("I",tblname[,type]) – I/O 変換テーブルの名前 (type の説明は DefIO を参照)。
- ・ SetPDef("X",tblname) – \$X/\$Y アクション・テーブルの名前。
- ・ SetPIO(tblname) – I/O 変換を選択します。
- ・ SetPM(tblname) – パターン・マッチを選択します。
- ・ SetPitch(value) – 現行デバイスのピッチ値を設定します。
- ・ SetTitle(tblname) – タイトルの大文字/小文字変換を選択します。
- ・ SetUpper(tblname) – 大文字変換を選択します。
- ・ SetXY(tblname) – \$X/\$Y アクションを選択します。
- ・ Set("C",tblname) – 呼び出しプロセスのテーブルを指定します。
- ・ Set("E",tblname) – 呼び出しプロセスのテーブルを指定します。
- ・ Set("I",tblname) – 現行デバイスのテーブルを指定します。
- ・ Set("J",tblname) – 呼び出しプロセスのテーブルを指定します。
- ・ Set("L",tblname) – 呼び出しプロセスのテーブルを指定します。
- ・ Set("P",tblname) – 呼び出しプロセスのテーブルを指定します。
- ・ Set("T",tblname) – 呼び出しプロセスのテーブルを指定します。
- ・ Set("U",tblname) – 呼び出しプロセスのテーブルを指定します。
- ・ Set("X",tblname) – 現行デバイスのテーブルを指定します。
- ・ StatIO – 現行デバイスの I/O 変換状態を返します。
- ・ StatPIO – 呼び出しプロセスの I/O 変換状態を返します。

2.2 モーダル変換に対するエントリー・ポイント

以下の演算は、モーダル変換用としてのみ使用されます。

- ・ `GetInpMode()` – 現行デバイスの入力モード文字列インデックスを取得します。
- ・ `GetInpModeStr(nbr)` – 現行デバイスの入力モード文字列値を取得します。
- ・ `GetOutMode()` – 現行デバイスの出力モード文字列インデックスを取得します。
- ・ `GetOutModeStr(nbr)` – 現行デバイスの出力モード文字列値を取得します。
- ・ `GetPInpMode()` – 呼び出しプロセスの入力モード文字列インデックスを取得します。
- ・ `GetPInpModeStr(nbr)` – 呼び出しプロセスの入力モード文字列値を取得します。
- ・ `GetPOutMode()` – 呼び出しプロセスの出力モード文字列インデックスを取得します。
- ・ `GetPOutModeStr(nbr)` – 呼び出しプロセスの出力モード文字列値を取得します。
- ・ `SetInpMode(nbr)` – 現行デバイスの入力モード文字列インデックスを設定します。
- ・ `SetInpModeStr(nbr,value)` – 現行デバイスの入力モード文字列値を設定します。つまり、`mode(nbr) = value` です。
- ・ `SetOutMode(nbr)` – 現行デバイスの出力モード文字列インデックスを設定します。
- ・ `SetOutModeStr(nbr,value)` – 現行デバイスの出力モード文字列値を設定します。
- ・ `SetPInpMode(nbr)` – 呼び出しプロセスの入力モード文字列インデックスを設定します。
- ・ `SetPInpModeStr(nbr,value)` – 呼び出しプロセスの入力モード文字列値を設定します。
- ・ `SetPOutMode(nbr)` – 呼び出しプロセスの出力モード文字列インデックスを設定します。
- ・ `SetPOutModeStr(nbr,value)` – 呼び出しプロセスの出力モード文字列値を設定します。

2.3 I/O 変換の置換属性管理に対するエントリー・ポイント

これらの関数は、入出力へ変換の置換属性を設定、または取得します。これらの関数は、置換状態を示す次の値を返します。

- ・ 0 は標準（置換なし）を意味します。
- ・ 1 は置換が行われたことを意味します。

一部の関数では、以下のいずれかの引数を指定できます。

- ・ `ReplType` – 有効な変換を持っていない文字または文字列の置換タイプです。有効な値は以下のとおりです。
 - 0 = エラーの生成

- 1 = 置換 (値または文字列)
 - 2 = 無視 (元の文字の使用)
- ・ ReplVal - ReplType が 1 (置換) の場合に使用される値または文字列です。

関数は以下のとおりです。

- ・ GetInpReplStat - 現行デバイスの入力置換状態を取得します。
- ・ GetInpReplType - 現行デバイスの入力置換タイプを取得します。
- ・ GetInpReplVal - 現行デバイスの入力置換値を取得します。
- ・ GetOutReplStat - 現行デバイスの出力置換状態を取得します。
- ・ GetOutReplType - 現行デバイスの出力置換タイプを取得します。
- ・ GetOutReplVal - 現行デバイスの出力置換値を取得します。
- ・ GetPInpReplStat - プロセスの入力置換状態を取得します。
- ・ GetPInpReplType - プロセスの入力置換タイプを取得します。
- ・ GetPInpReplVal - プロセスの入力置換値を取得します。
- ・ GetPOutReplStat - 出力置換状態を取得します。
- ・ GetPOutReplType - プロセスの出力置換タイプを取得します。
- ・ GetPOutReplVal - プロセスの出力置換値を取得します。
- ・ SetInpReplStat(ReplType) - 現行デバイスの入力置換状態を設定します。
- ・ SetInpReplType(ReplType) - 現行デバイスの入力置換タイプを設定します。
- ・ SetInpReplVal(ReplVal) - 現行デバイスの入力置換値を設定します。
- ・ SetOutReplStat(ReplType) - 現行デバイスの出力置換状態を設定します。
- ・ SetOutReplType(ReplType) - 現行デバイスの出力置換タイプを設定します。
- ・ SetOutReplVal(ReplVal) - 現行デバイスの出力置換値を設定します。
- ・ SetPInpReplStat(ReplType) - プロセスの入力置換状態を設定します。
- ・ SetPInpReplType(ReplType) - プロセスの入力置換タイプを設定します。
- ・ SetPInpReplVal(ReplVal) - プロセスの入力置換値を設定します。
- ・ SetPOutReplStat(ReplType) - プロセスの出力置換状態を設定します。
- ・ SetPOutReplType(ReplType) - プロセスの出力置換タイプを設定します。
- ・ SetPOutReplVal(ReplVal) - プロセスの出力置換値を設定します。

“Set” 関数は一般的に、この値が意味を持つとき、以前の設定を返します。

