

IBM InfoSphere Global Name Management



開発者ガイド

バージョン 6 リリース 0

IBM InfoSphere Global Name Management



開発者ガイド

バージョン 6 リリース 0

注記

本書および本書で紹介する製品をご使用になる前に、特記事項のセクションに記載されている情報をお読みください。

本書は、IBM InfoSphere Global Name Recognition (製品番号 5724-Q20) バージョン 6.0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM InfoSphere Global Name Management
Developer's Guide
Version 6 Release 0

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 2001, 2013.

目次

前書き	v
-----	---

第 1 章 IBM InfoSphere Global Name Management の概要 1

バージョン 6.0 の新機能	2
バージョン 6.0 - リリース情報	2
バージョン 6.0 - 機能および機能強化	8
製品体系	9
コンポーネント API	10
IBM NameWorks	11
クライアント・アプリケーション	11
Distributed Search	12

第 2 章 名前および名前の突き合わせの概要 13

名前突き合わせの方法	13
名前カテゴリー	14
個人名	14
組織名	18
名前のパーツ	18
解析ツリー	18
解析した名前	19
名前フィールド	20
名前句	22
名前トークン	22
名前リスト	27
名前データ・アーカイブ	27
外部トークン・リスト	27
名前音訳	28
音訳規則ファイル	28
中国語の音訳についての概要	29
日本語の音訳についての概要	32

第 3 章 名前の解析 33

NameWorks を使用した名前の解析	33
名前を個々の部分に解析する	33
NameParser を使用した名前の解析	34
入カストリングのタイプ	34
名前の解析のための NameParser 関数	35
NameParser の句オーバーライド・リスト	35

第 4 章 名前の分析 39

NameWorks を使用した名前の分析	39
NameWorks を使用した名前の国/地域別情報の識別	40
国/地域別情報の識別	40
国/地域別情報コード	40
氏名の国/地域別情報の識別	42
名前フィールドの国/地域別情報の識別	43
組織名の国/地域別情報の識別	43
NameWorks を使用した名前の性別の識別	44

氏名の性別の識別	44
名の性別の識別	45
NameWorks を使用した名前の関連国の識別	45
関連国 (Country of Association)	45
氏名の関連国の識別	46
名と姓の関連国の識別	46
NameWorks を使用した名前変形の生成	47
名前の変形	47
氏名の名前変形のリストの生成	48
名と姓の名前変形のリストの生成	49
コンポーネント API を使った名前の分析	50

第 5 章 名前の検索 51

IBM NameWorks におけるデータ・リストの管理	51
データ・リスト	51
データ・リストへの名前の追加	52
データ・リスト上の名前の更新	53
データ・リストからの名前の削除	54
IBM NameWorks のマイグレーション	55
検索のための名前の準備	58
シナリオ: 名前の検索	59
名前検索のための名前オブジェクトの作成	59
IBM NameWorks を使用した名前の検索	62
検索方針の管理	63
比較パラメーターのオーバーライド	66
検索のための名前の準備	70
IBM NameWorks を使用した名前の分類、名前の比較、および日付の比較	71
データ・リストでの名前の検索	78
固有の名前に関連付けられた名前の補足データの取得	79
NameHunter を使用した名前の検索	79
NameHunter の概要	80
NameHunter API クイック・スタートの例	93
NameHunter のサンプル・アプリケーション	97
比較パラメーターの変更	98
NameHunter の音訳規則セットの構成	127
Distributed Search を使用した名前の検索	128
Name Preprocessor の概要	130
Distributed Search のパフォーマンスおよび構成の概要	143
NameHunter Distributed Search XML インターフェース	153
Enterprise Name Search を使用した名前の検索	166
Enterprise Name Search ユーザー・セキュリティーの管理	166
NameLoader ユーティリティーを使用した名前リストの管理	172

第 6 章 IBM NameWorks の構成 . . . 185

IBM NameWorks 構成ファイルを使用した構成設定の指定	186
構成ファイルの General セクション	188
構成ファイルの Custom Tokens セクション	189
構成ファイルの Datalist セクション	190
Configuration クラスを使用した構成設定の指定追加の音訳規則ファイルを使用するように IBM NameWorks 構成を更新する	199 201

第 7 章 トラブルシューティングとサポート 203

IBM InfoSphere Global Name Management のトラブルシューティング・チェックリスト	203
コンポーネント API の C++ エラー・コード	205
参照データのエラー・コード	206
グローバル・エラー・コード	210
入力エラー・コード	211
内部エラー・コード	214
IBM NameWorks のエラー・コード	215
IBM NameWorks の C++ エラー・コード	216
Distributed Search のエラー・コード	220
Enterprise Name Search のエラー・コード	223
ENS コンソールのエラー・コード	223
ENS 検索のエラー・コード	230
知識ベースの検索	236
ログ・ファイル	237
トレース	237

IBM サポートへのお問い合わせ	238
サポート・アップデートへのサブスクライブ	239

付録. 用語集 243

A	243
C	243
D	244
F	244
G	244
H	244
I	244
J	245
M	245
N	245
O	246
P	246
Q	246
R	247
S	247
T	248

特記事項 249

商標 253

使用条件 255

索引 257

前書き

IBM InfoSphere Global Name Management は、国/地域固有の名前データを活用し、名前の国/地域別情報に関連した規則を構築して、最適なマッチング、管理、解析、およびスコアリング結果を実行します。業界をリードする Global Name Management のテクノロジーを使用すると、複数の地理上の国/地域にまたがって多文化の名前を検索、認識、管理したり、潜在的な脅威を選別したり、身元調査を実行したりできます。

本書について

IBM® InfoSphere® Global Name Management 開発者ガイド および API リファレンスは、提供されている API を使って IBM InfoSphere Global Name Recognition テクノロジーを活用するアプリケーションを作成するうえで役立ちます。

この情報は、読者の便宜を図るためにいくつかの形式で提供されています。(製品インストール DVD および ibm.com からのダウンロードで利用可能な) PDF 形式に加えて、製品の全情報を含むオンライン版が IBM InfoSphere Global Name Management インフォメーション・センターにあります。また、ブラウザー・ベースの製品インフォメーション・センターをローカル・マシンにインストールすることもできます。

対象読者

開発者ガイド および API リファレンスは、ご使用の環境に製品を正常にデプロイしてアプリケーションを作成するうえで役立ちます。

ご意見の送付方法

IBM にお客様のご意見をお寄せください。本書または他の IBM InfoSphere Global Name Management 資料についてコメントがある場合は、インフォメーション・センターの「フィードバック」リンク、または次のフォームを使用することができます。

<http://www.ibm.com/software/data/rcf/>

第 1 章 IBM InfoSphere Global Name Management の概要

IBM InfoSphere Global Name Management 製品には、多文化の名前データ・セットを管理、検索、分析、および比較するためのテクノロジーが含まれています。この操作は、名前の国/地域に関連する文化圏固有の名前データと言語規則を活用することによって行います。

IBM InfoSphere Global Name Management に含まれているコンポーネントを使用すると、以下の操作が可能になります。

- 最も高い確率で予想される、名前の文化圏 (民族的区分) の識別と分類。特定の姓と名が最も多く見られる国の識別を含みます。
- 名の性別 (男性または女性) の相対頻度を認識して報告すること
- 個人名を姓と名の構成要素に解析すること
- 名前の構成要素 (姓と名) の変形を含むリストの生成
- 文化圏固有の検索方針を使った名前の検索と突き合わせ
- 一般的なスペル上の変形や国/地域的な変形によって名前が変化する場合、スペル上は全く異なっても音学的に名前が相互関連している場合、あるいはスペルや入力のエラーで名前が破損している場合であっても、名前を突き合わせること
- 名前のいくつかの構成要素 (姓、名) が欠落している場合や順序が正しくない場合であっても、名前を突き合わせること
- 発音とつづりの両方で名前を突き合わせて、最も近い一致を最初に戻すこと
- 微調整可能なアプリケーション固有の結果を戻すために検索パラメーターを調整すること
- 個人名を組織名から分離すること
- 日付値の比較。これは、日付値 (例えば生年月日) が関連付けられた名前を検索するうえで役立ちます。あるいは複数の日付値の間の差異の計算。

API コンポーネントとサーバー・プロセス

このバンドルには、以下のような API コンポーネントとサーバー・プロセスが含まれています。

- IBM NameWorks (統合された高水準 API)
- NameParser[®]
- NameClassifier[™]
- NameClassifier - Country of Association
- Country of Association
- NameHunter[®]
 - Distributed Search プロセス
- NameGenderizer[®]
- NameVariationGenerator[®]
- NameSifter

- DateCompare

関連概念:

『バージョン 6.0 の新機能』

IBM InfoSphere Global Name Management のこのバージョンには、新しい機能や拡張された製品機能が多数含まれています。

『バージョン 6.0 - リリース情報』

このリリース情報には、IBM InfoSphere Global Name Management バージョン 6.0 に関する情報が含まれています。例えば、インストールに関する注意事項、既知の問題、修正された問題、使用上の注意などです。

バージョン 6.0 の新機能

IBM InfoSphere Global Name Management のこのバージョンには、新しい機能や拡張された製品機能が多数含まれています。

IBM InfoSphere Global Name Management バージョン 6.0 の最新情報については、製品 Web サイト (<http://www-03.ibm.com/software/products/en/infosphere-global-name-management>) を参照してください。

プラットフォームの追加と非推奨

IBM InfoSphere Global Name Management のこのバージョンは、いくつかのプラットフォームのより新しいバージョンをサポートしています。このリリースで非推奨になったプラットフォームもあります。サポートされているプラットフォームおよび開発環境のリストについては、[ibm.com](http://www.ibm.com) の system requirements を参照してください。

関連概念:

1 ページの『第 1 章 IBM InfoSphere Global Name Management の概要』

IBM InfoSphere Global Name Management 製品には、多文化の名前データ・セットを管理、検索、分析、および比較するためのテクノロジーが含まれています。この操作は、名前の国/地域に関連する文化圏固有の名前データと言語規則を活用することによって行います。

バージョン 6.0 - リリース情報

このリリース情報には、IBM InfoSphere Global Name Management バージョン 6.0 に関する情報が含まれています。例えば、インストールに関する注意事項、既知の問題、修正された問題、使用上の注意などです。

注: 最新版のリリース・ノートについては、[ibm.com](http://www.ibm.com) の インフォメーション・センターでオンライン・バージョンをご覧ください。個別の HTML ファイルのリリース・ノートは、インストール・メディアに付属していますし、製品のサポート・ポータルから個別にダウンロードすることもできます。

目次

- 3 ページの『システム要件』
- 3 ページの『パフォーマンス上の考慮事項』

- 『IBM Global Name Management バージョン 6.0 へのアップグレード』
- 『フィックスおよび修正のリスト』
- 『製品を使用時に発生する既知の問題』
- 4 ページの『既知の問題 - Enterprise Name Search』
- 7 ページの『既知の問題に関する最新情報を参照する方法』
- 7 ページの『製品資料』
- 8 ページの『発表』

システム要件

ハードウェアおよびソフトウェアの互換性に関する最新情報については、
<http://www-01.ibm.com/support/docview.wss?&uid=swg27019150> で詳しく説明
 されているシステム要件を参照してください。

パフォーマンス上の考慮事項

Linux for IBM zSeries (s390) を実行するコンピューターでは、大きなデータ・リスト (500 万個以上の名前) に対して検索を実行するときに検索のトークンが 5 つを超えないようにしてください。5 つより多いトークンを検索照会に含めた場合、サイズ制限を超える結果が戻され、トランザクション・タイムアウトが発生して Web サービスが失敗する可能性があります。パフォーマンスに関する詳細情報は、製品のインフォメーション・センターを参照してください。

IBM Global Name Management バージョン 6.0 へのアップグレード

インストール・プログラムにより、新規構成ファイルが *install_path/data/*config.template* に作成されます。*install_path* は、IBM InfoSphere Global Name Management をインストールしたディレクトリーの絶対パスです。既存の構成ファイルは保持され、インストール・プログラムによって上書きされません。

フィックスおよび修正のリスト

次のリストは、バージョン 6.0 に含まれている機能改善や修正された問題のいくつかについて説明します。

- 障害修復 #1
- 障害修復 #2
- 障害修復 ...

製品を使用時に発生する既知の問題

IBM InfoSphere Global Name Management バージョン 6.0 をインストールして使用する前に、以下の情報を確認してください。

NameParser C++ API への変更

NameParser API は、NameWorks 解析 API の動作と整合するように、バージョン 6.0 で単純化されました。NameParser を直接利用する C++ プログラムは、バージョン 6.0 と共に動作するように変更する必要があります。

NameWorks 構成ファイル・エントリーに対する変更および追加事項

ネイティブ文字比較をサポートするために、比較ファイルの新しいサポートがバージョン 6.0 に追加されました。

- NativeTaq=ネイティブ文字 TAQ ファイルへのパス

ネイティブ文字 TAQ データを指定します。

- NativePnVar=ネイティブ文字個人名の変形ファイルへのパス

ネイティブ文字個人名の変形データを指定します。

- NativeOnVar=pネイティブ文字組織名の変形ファイルへのパス

ネイティブ文字組織名の変形データを指定します。

- NativePnReg=ネイティブ文字個人名の正規化モジュールへのパス、文字名

特定の文字タイプに対するネイティブ文字個人名の正規化データを指定します。

- NativeOnReg=ネイティブ文字組織名の正規化モジュールへのパス、文字名

特定のスクリプト・タイプに対するネイティブ文字組織名の正規化データを指定します。

既知の問題 - Enterprise Name Search

Enterprise Name Search をインストールして使用する前に、以下に示す追加の情報と指示に注意してください。

Enterprise Name Search には UTF-8 エンコードが必須

SOAP Web サービスに非 ASCII 文字を含むデータを処理している場合は、ロケール環境変数が UTF-8 エンコードを使用するように設定されていることを確認してください。REST Web サービスには影響しません。システムのデフォルトが UTF-8 エンコードではない場合は、以下を実行します。

Linux/UNIX 環境:

1. 使用する言語と国のロケールの UTF-8 バージョンを見つけてインストールします。このロケールは、ENS セルで使用されるすべてのマシンにインストールされていなければなりません。 **locale -a** コマンドを使用して、マシンにインストールされたロケールのリストを見ることができます。例:

```
jksmith@din:~$ locale -a
C
C.UTF-8
en_AG
en_AG.utf8
en_AU.utf8
en_GB.utf8
en_US.utf8
ja_JP.utf8
POSIX
```

使用する言語と国のロケールの UTF-8 バージョンを見つけてインストールします。例えば、アメリカにいる場合は、リストから「en_US.UTF-8」を選択することができます。

2. ENS WebSphere プロセスを開始する (<ENS Install Home>/bin/start-<profile name>.sh スクリプトを実行する) シェル環境で、ENS を開始する前に、以下の環境が選択された UTF-8 ロケールに設定されていることを確認してください。 Bourne シェル (sh) の例:

```
export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8
```

sh シェルの例 :

```
LC_ALL=en_US.UTF-8
LANG=en_US.UTF-8
export LC_ALL
export LANG
```

Windows 環境:

Windows では、UTF-8 をシステム全体またはユーザー全体のデフォルト文字エンコードに設定することがサポートされないため、ケースごとに UTF-8 を使用するようにアプリケーションを構成する必要があります。ENS の場合、WebSphere と ENS を実行する Java JVM が UTF-8 を使用するように構成する必要があります。IBM J9 Java JVM は WebSphere にバンドルされていて、JVM オプションを指定するための「IBM_JAVA_OPTIONS」という名前の環境変数があります。この変数は、複数の方法で Windows で設定することができます。

- Windows のコマンド・プロンプトから:
 1. 環境変数を設定します: **IBM_JAVA_OPTIONS=-Dfile.encoding=UTF-8**
 2. **start-(ENS Profile Name).bat** スクリプトを実行します。
- 特定の Windows ユーザーに適用範囲を設定します:

この設定は、ユーザーによって実行されるすべての IBM J9 ベースの Java アプリケーションに伝搬します。

1. システムのコントロール・パネルの「詳細設定」タブにある「環境変数...」ボタンをクリックします。Windows 2008 Server では、「スタート」>「コントロール・パネル」>「システムの詳細設定」を選択すると、システムのコントロール・パネルの「詳細設定」タブに直接移動できます。「環境変数」ダイアログが表示されます。
2. 「<現行ユーザー名> のユーザー環境変数」セクションで、「新規...」ボタンをクリックします。
3. 「新しいユーザー変数」ボックスが表示されます。「変数名」フィールドに、IBM_JAVA_OPTIONS と入力します。
4. 「変数値」フィールドに、-Dfile.encoding=UTF-8 と入力します。
5. ボックスの「OK」をクリックします。その後、「環境変数」ダイアログとシステムのコントロール・パネルの両方で「OK」をクリックします。

6. Windows からログアウトしてから、再度ログインします。IBM J9 JVM を使用しているこのユーザーによって開始された Java アプリケーションは、デフォルトで UTF-8 エンコード方式を使用するようになります。

- Windows システム全体に適用範囲を設定します

この設定は、Windows サーバーで開始されるすべての IBM J9 ベースの Java アプリケーションに伝搬します。

1. システムのコントロール・パネルの「詳細設定」タブにある「環境変数...」ボタンをクリックします。Windows 2008 Server では、「スタート」>「コントロール・パネル」>「システムの詳細設定」を選択すると、システムのコントロール・パネルの「詳細設定」タブに直接移動できます。「環境変数」ダイアログが表示されます。
2. 「システム環境変数」セクションで、「新規...」ボタンをクリックします。
3. 「新しいシステム変数」ボックスが表示されます。「変数名」フィールドに、IBM_JAVA_OPTIONS と入力します。
4. 「変数値」フィールドに、-Dfile.encoding=UTF-8 と入力します。
5. ボックスの「OK」をクリックします。その後、「環境変数」ダイアログとシステムのコントロール・パネルの両方で「OK」をクリックします。
6. システムをリブートします。IBM J9 JVM を使用する、このシステムで開始された Java アプリケーションは、デフォルトで UTF-8 エンコード方式を使用するようになります。

注: 上記の IBM_JAVA_OPTIONS 環境変数ソリューションは、Linux や AIX など、IBM J9 JVM を使用して WebSphere を実行する他のプラットフォーム上でも機能します。これは Solaris では機能しません。IBM は IBM J9 JVM をそのオペレーティング・システムでは使用しないためです。

ENS CU ユーティリティを使用した複数のプロファイルの作成

一度に 1 つのプロファイルだけを作成してください。ENS CU ユーティリティを使用した複数のプロファイルの作成は、多数のプロファイルがある場合には特に、非常に長い時間がかかることがあります。追加のプロファイルを作成するたびに、そのプロファイルと既存の各プロファイルとの間に SSL 暗号キーが生成されるため、作成に時間がかかるようになります。それらのキーは、任意の 2 つのプロファイル間の SSL/HTTPS セキュア通信機能をサポートするために必要です。例えば、35 個のプロファイルを作成する場合、その 35 番目のプロファイルは 34 個の暗号キーを作成する必要があります。

ENS 分散環境内のすべてのシステムに 1 つのデータベース・ドライバー

ENS 資料の他の個所にも記載されているように、ENS 環境 (ENS セルと呼ばれる) にあるすべての ENS ホストおよびサーバー・マシンに対して、データベース・ドライバーが同じでありかつ 1 つの場所にあることが必要です。データベース・ドライバーの場所への動的リンクを作成できます。す

すべてのマシンが動的リンクを指定している場合、各マシンで変更を行わなくてもデータベース・ドライバーの場所を変更できます。

既知の問題に関する最新情報を参照する方法

既知の問題は、サポート・ポータル (http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/InfoSphere_Global_Name_Recognition) の技術情報で文書化されています。

1. 「サポートの検索」機能を使用して、「用語、エラー・コード、または **APAR** 番号の入力 (**Enter terms, error code or APAR #**)」フィールドに、検索するキーワード、句、エラー・コード、APAR 番号を入力します。
2. 「問題の解決 (**Solve a problem**)」を選択します。
3. 「検索」をクリックします。

問題の検出および解決に応じて、IBM サポート・チームがサポート・ポータルを更新します。サポート・ポータルを検索することで、問題の解決策を迅速に見つけることができます。

公開時にインストールに関する既知の問題はありませんでした。最新情報については、サポート・ポータルを調べてください。

製品資料

バージョン 6.0 の製品資料は、次の場所にあります。

バージョン **6.0** インフォメーション・センター

- [ibm.com](http://pic.dhe.ibm.com/infocenter/gnrgnm/v6r0m0/index.jsp) の <http://pic.dhe.ibm.com/infocenter/gnrgnm/v6r0m0/index.jsp> にアクセスします。
- 製品インストールの一部としてインストールされるローカル・サーバーにアクセスします。

Web ブラウザーを開き、サーバーの URL を `http://<servername>:<HTTP port#>/help/index.jsp` の形式で入力します。

`<HTTP port#>`

インストール中に指定する資料のポート番号。

`<servername>`

WebSphere Application Server のホスト名または IP アドレス。

Name Analyzer の国/地域別情報の参照

IBM InfoSphere Global Name Management Name Analyzer ツールで、国/地域別情報の参照情報が提供されています。

IBM 製品のサポート・ホーム

[ibm.com](http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/InfoSphere_Global_Name_Recognition) の http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/InfoSphere_Global_Name_Recognition にアクセスします。

技術情報やサポートに関するその他の情報に加えて、インフォメーション・センターへのリンク、PDF 版の製品情報、およびリリース情報の最新の更新も含まれています。

発表

最新の発表レターへのリンクが、<http://www.ibm.com/software/data/infosphere/global-name-recognition/> ページにあります。以下の情報については、発表をご覧ください。

- 新機能の説明を含む詳細な製品説明
- 製品の位置付けについての記述
- パッケージ化と注文の詳細
- 国際的な互換性情報

著作権および商標

IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml の「Copyright and trademark information」をご覧ください。

関連概念:

1 ページの『第 1 章 IBM InfoSphere Global Name Management の概要』
IBM InfoSphere Global Name Management 製品には、多文化の名前データ・セットを管理、検索、分析、および比較するためのテクノロジーが含まれています。この操作は、名前の国/地域に関連する文化圏固有の名前データと言語規則を活用することによって行います。

バージョン 6.0 - 機能および機能強化

IBM InfoSphere Global Name Management バージョン 6.0 には、重要な新しい製品機能、および既存の機能に対する機能拡張が含まれています。

表 1. バージョン 6.0 製品の機能と機能強化

バージョン 6.0 の機能強化	説明
新しく標準となった国/地域別情報: ポーランド語、ポルトガル語、およびトルコ語	ポーランド語、ポルトガル語、およびトルコ語が、新しい標準の国/地域別情報として追加されました。これらの新しい国/地域別情報は、国/地域に固有の変形、性別データ、および比較パラメーターをサポートします。ポルトガル語はヨーロッパ・グループの国/地域別情報にも含まれ、トルコ語は東南アジア・グループの国/地域別情報にも含まれています。
ネイティブ文字サポートの拡張	ネイティブ文字のサポートが拡張されて、拡張ラテン語 (ヒスパニック、ポーランド語、ポルトガル語) とデーバナーガリー文字 (インド語) で記述された個人名、およびキリル文字 (ロシア語)、ハングル文字 (韓国語)、漢字 (日本語)、漢字 (中国語)、デーバナーガリー文字 (インド語) で記述された組織名を扱うことができるようになりました。
新しい組織名の国/地域別情報	組織名スコアリングが拡張されて、さらに多くの国/地域別情報 (中国語、ヒスパニック、日本語、韓国語、ポーランド語、ポルトガル語、ロシア語) をサポートするようになりました。

表 1. バージョン 6.0 製品の機能と機能強化 (続き)

バージョン 6.0 の機能強化	説明
改善された短い名前のスコアリング	新しいスコアリング手順と比較パラメーターにより、可能な誤植エラーによる相違のある 2 つの短い名前が、実際には別個の名前であることの可能性が検討されるようになりました。例えば、「Amir」と「Mair」には 1 カ所の入れ替わりによる違いがありますが、これらは別個の名前であり、一方「Amir」と「Aimr」は誤植エラーを表しています。名前データ・アーカイブからの頻度情報を使用してこの判別が行われます。この後に特殊ロジックが名前の比較に使用されません。
改善された性別情報	分析に使用できる性別データの量が 10 倍以上に増大し、国/地域に固有の性別情報が使用可能になりました。例えば、「Juan」はほとんどの国では男性の名前ですが、中国では主に女性の名前です。
単純化された NameParser C++ API 同じ外部 ID の名前が複数存在する場合の ENS のサポート	NameParser の C++ API が単純化されました。単一の呼び出しで、名前に対するすべての解析が返されるようになりました。以前の ENS バージョンでは、名前リスト内の名前はすべて固有である必要がありました。バージョン 6 ではこの制限がなくなり、同じリスト内に同じ外部 ID の名前が複数含まれていてもかまいません。
GNM のスクリプトおよび国/地域別情報の変更に対する ENS のサポートおよび使用	スクリプトおよび国/地域別情報に対する GNM の変更が ENS でサポートされ使用されるようになりました。これには、国/地域別情報の追加、スクリプト・タイプからの国/地域別情報の推測、スクリプト・タイプの認識の追加、および組織名の国/地域別情報のサポートが含まれます。
名前に含まれているコンマに対する ENS NameLoader のサポート	ENS NameLoader のコンマ区切り値入力ファイル (.csv) で、名前フィールドを引用符で囲み、フィールド中にコンマを含められるようになりました。
DB2 10.5 および Oracle 12c に対する ENS のサポート	ENS の実行に DB2 10.5 および Oracle 12c を使用できるようになりました。これにより、DB2 10.5 の関数索引のサポートを活用できます。Oracle では、データベースの詳細を指定する際にサービス名 (SID) を使用できます。
ENS での WebSphere Liberty の使用	ENS は、Web コンポーネントをホストするアプリケーション・サーバーとして、これまでの Embedded WebSphere に代わり、WebSphere Liberty を内部的に使用するようになりました。その結果、セキュリティーの実装が変更され、ENS の構成がよりシンプルになりました。

小さな変更を含む修正や新規 API および変更された API を含む改善点のすべてに関する詳細なリストについては、製品インストール・メディアまたは IBM InfoSphere Global Name Management 製品サポート・ポータルにあるリリース情報のファイルを参照してください。

製品体系

IBM InfoSphere Global Name Management の製品体系は、コンポーネント API、IBM NameWorks API、およびこれらの API とやり取りするクライアント・アプリケーションとサーバー・アプリケーションで構成されます。

サーバー・アプリケーションは、コンポーネント API の機能に基づいて構築され、その機能を提供する、サーバー・サイドのアプリケーションです。IBM InfoSphere Global Name Management には、NameHunter および Distributed Search サーバー・アプリケーションが含まれます。このコンポーネントと IBM NameWorks API を使用して、独自のサーバー・アプリケーションを開発することもできます。

以下に示すアプリケーションやコンポーネントに加えて、バージョン 6.0 には Enterprise Name Search が含まれています。これは別個にインストールされたコンポーネントのセットで、名前検索の管理、保守、および実行を容易にするための Web アプリケーション・インターフェースが備わっています。

コンポーネント API

IBM InfoSphere Global Name Management コンポーネント API は C++ ライブラリーであり、任意の C++ アプリケーションに統合可能です。

すべてのコンポーネント API は 1 つの名前の分析機能を実行しますが、NameHunter および DateCompare は複数のオブジェクト (それぞれ、複数の名前と複数の日付) を受け入れて、それらを比較します。それぞれのコンポーネント API を以下にリストします。

NameClassifier

NameClassifier パッケージ (ibmgmr::classifier) は、ある個人名が 1 つ以上の国/地域別環境に関連している可能性の高さを判別します。

NameGenderizer

NameGenderizer パッケージ (ibmgmr::genderizer) は、個人名の名の部分の性別分散統計を提供します。

NameParser

NameParser パッケージ (ibmgmr::parser) は、個人名を構成要素 (名、姓、称号、および修飾子) に解析します。

NameVariationGenerator

NameVariantGenerator パッケージ (ibmgmr::nvg) は、個人名各構成要素の変形から成るリストを生成します。これらの代替的なスペリングは、同じ文化的/民族的背景を持つ名前によく見られるスペル変形パターンに基づいています。

NameSifter

NameSifter パッケージ (ibmgmr::sifter) は、組織名を個人名から分離します。

NameClassifier-Country of Association (NC_COA)

NameClassifier-Country of Association パッケージ (ibmgmr::cc) は、NameClassifier と共に Country of Association (COA) を使用して、名前に関連した国/地域別環境に関する非常に正確な結果を生成します。

Country of Association (COA)

COA パッケージ (ibmgmr::coa) は IBM 名前データ・アーカイブ (NDA) に含まれるデータを参照して、個人名のそれぞれの構成要素がこれまでに出現した国々をリストします。

NameHunter

NameHunter パッケージ (ibmgmr::hunter) は個人名と組織名から成るペアを比較し、これらの名前タイプを持つリストを検索します。

DateCompare

DateCompare (ibmgmr::datecompare) は 2 つの日付値を比較して類似性スコアを戻します。DateCompare は、12 カ月から成るグレゴリオ暦の日付だけを比較できます。

IBM NameWorks

IBM NameWorks は個々の IBM InfoSphere Global Name Management コンポーネントを 1 つの統一された、使いやすいアプリケーション・プログラミング・インターフェース (API) に結合します。さらに、この機能を Java アプリケーションに、および Web サービスとして拡張します。

IBM NameWorks は以下のような 2 つの異なる API クラスで構成されています。

Analytics クラス

1 つの名前を評価するのに必要な機能 (名前の解析、国/地域別情報の分類、性別の特定、カテゴリー化、変形の生成、関連国情報など) が含まれています。これらの言語処理を個別に、または一緒に使用することができます。例えば analyze() メソッドはすべての言語操作を実行して、名前に関するすべての分析情報を含む 1 つの結合された結果を生成します。

Scoring クラス

2 つの名前の比較、または 1 つ以上のデータ・リストにおける名前の検索を行うために必要な機能が含まれています。さらに、検索結果を調整するために使用できる日付比較、名前分類などの補助タスクもあります。検索の準備 (解析および国/地域別情報の分類) を別個に実行することも、検索操作に含めることもできます。

IBM NameWorks にアクセスする方法として、C++ 関数、Java 関数、または Web サービス の 3 つがあります。C++ および Java インターフェースは、サポートされる任意のプラットフォーム上で直接使用できます。Web サービス・インターフェースは、SOA 環境でリモートに、またはローカルに使用できます。Web サービスを使用できるすべてのプログラミング環境は、IBM NameWorks の名前分析/比較ツールを活用できます。同様に、Java インターフェースを使ってカスタム SOA アプリケーションを作成できます。

クライアント・アプリケーション

クライアント・アプリケーションは、コンポーネント API または IBM NameWorks に基づいて構築されます。これらのアプリケーションは、同じフレームワークに基づいて構築されたサーバー・サイド・アプリケーションと通信することができます。

どちらの API パッケージを使用しても、広範な物理的アーキテクチャーを示すアプリケーションを構築できます。1 つのホスト・プラットフォーム上で稼働する単純なスタンドアロン・ソリューションから、複数のネットワーク・ホスト・プラット

フォーム上 (例えばクライアント/サーバー環境) で独立処理として作動するより複雑なソリューションに至るまで可能です。クライアント・アプリケーションには、主に次のような 2 種類が存在します。

エンド・ユーザー・アプリケーション

コンポーネント API または IBM NameWorks パッケージに基づいて構築され、ユーザー・マシン上で稼働するようにコンパイルされるアプリケーション。

クライアント・サイド・アプリケーション

コンポーネント API または IBM NameWorks パッケージに基づいて構築されるサーバー・アプリケーションと通信するクライアント・サイド・アプリケーション。例えば、SOAP API から構築される IBM NameWorks Web サーバー・クライアント。

Distributed Search

Distributed Search によって公開される単一サーバー・プロセス形式の NameHunter API 機能は、検索対象のデータ・リストのサイズのため、またはある時点で発生する検索トランザクション数のために、複雑で高パフォーマンスを要する検索要件に対応することができます。

Distributed Search は、何百万もの名前から成る大きなデータ・リストをロードするのに最も適しています。ただし、アプリケーションは複数のデータ・リストを 1 つの検索にロードすることができません。この制限のために、クライアントは単一の XML メッセージから複数のデータ・リストを検索できません。複数の小さなデータ・リストを 1 つのサーバー・アプリケーションにロードする必要がある場合は、IBM NameWorks に組み込まれた検索アプリケーションを使用してください。

Distributed Search との直接的なインターフェースが利用可能です。または、IBM NameWorks に基づいて構築されたエンド・ユーザー・クライアント・アプリケーションおよびサーバー・アプリケーションを介してやり取りすることもできます。

第 2 章 名前および名前の突き合わせの概要

名前の突き合わせはとりわけ難しいものとなる場合があります。その理由として、名前には一貫性のあるグローバル・スタンダードがなく、名前にはさまざまな情報が含まれる場合があります (その多くは任意指定)、それによって名前が非常に異なったものに見える場合があるからです。IBM InfoSphere Global Name Management 製品は、国/地域ごとに固有の最良の名前検索および突き合わせ機能を提供できる、多数の国/地域に対応した名前および言語情報を格納する固有の知識ベースを活用します。

名前突き合わせの方法

名前突き合わせの自動化ソフトウェアでは、通常 4 つの基本的な方法を利用します。つまり、完全一致突き合わせ、名前ディクショナリー、キー・ベース、および分析の 4 つです。

完全一致突き合わせ

完全一致突き合わせシステムで結果を返すには、データベース内で名前に完全一致する照会名が必要です。つまり、システムが返す結果は照会ストリングと完全に一致するもので、それ以外のは含まれません。

ディクショナリー

ディクショナリー突き合わせシステムは、名前をディクショナリー内で検索してその変形を見つけ出し、その後それらをデータベース項目に突き合わせることができます。特定の名前に関連付けられる一連の変形スペル形式は、前もって手動で編集しておくものであるため、未知または未確認の変形形式は、名前ディクショナリーには存在しません。また、特定の名前形式が複数の異なる名前の変形としてリストされる場合があります、その場合には、同じ照会名に対してディクショナリーの複数の一致項目を検出することに基づく決定ロジックが複雑になります。

キー・ベース

キー・ベース・システムは、キーという正規化形式に名前を短縮するアルゴリズムを適用します。理論的には、キー生成アルゴリズムで処理する際、等価または一致するスペル形式として理解される名前から生成されるキーはすべて同じになります。最も古く有名なキー生成アルゴリズムは Soundex で、この特許が最初に取得されたのは 1911 年です。

分析

分析名前突き合わせシステムは、つづり (スペル) 情報、ノイズ・フィルター技法、セマンティック・パターン、国/地域別情報パターン、および文法的パターンを同時に考慮し、2 つの名前間の類似性を測定します。名前比較をするためのこのペアワイズ方式は、名前に関連付けられた言語的コンテキストおよび国/地域別コンテキストの中でのその名前の使用方法に関する膨大な経験的情報へのアクセスにかなり依存する方式です。

IBM InfoSphere Global Name Management 製品が名前突き合わせで利用する方法は、分析です。名前分析で使用する言語データおよび国/地域別データは、世界中のさまざまな国で収集された、名前に関するデータの大規模なリポジトリによって

支えられています。この知識ベースを使用することにより、名前を突き合わせるときに詳細な区別をすることができ、2つの名前の相対的な類似性を正確に測定できるようになります。分析名の突き合わせにより、突き合わせ結果を階層順にリストして、最良の一致を最初に表示することが可能になります。さらに、さまざまな異なる操作設定、ユーザー設定、およびビジネス・ルールに突き合わせ結果を合わせられるように、ランキング・アルゴリズムおよびスコア・アルゴリズムを調整できます。

名前カテゴリー

名前処理の際、名前は名前カテゴリー、つまり個人名または組織名のいずれかに関連付けられます。これら2つのカテゴリーの名前の使用法は類似していますが、これらの名前は重要な相違点に基づいて分離されているため、分析および突き合わせの際には、異なるタイプの言語リソースおよび参照データ・リソースが各カテゴリーの名前に適用されます。

名前をカテゴリー化するとき、IBM InfoSphere Global Name Management のコンポーネントは、名前を以下のカテゴリーに分類します。

- 個人名。他のカテゴリーに属することを示唆する標識が含まれません (例えば、「Linda K. Smith」)
- 組織名。非個人に使用される何らかの形式の標識が含まれます (例えば、「Smith & Company」)
- 不明の名前。つづりの誤りがあるように見える要素、または個人名にも組織名にも通常現れないその他の構造が含まれる要素が含まれます (例えば、「SMI」)
- 両方。職業修飾子を含んだ名前です。名前が個人名から派生したビジネス名である可能性があります (例えば、「Linda Smith Architect」)

名前が個人名以外にカテゴリー化された場合、名前を非個人として修飾する標識またはパターンを示す理由コードがコンポーネントによって提供されます。

個人名

個人名は、1つ以上の名、その名前の属する国/地域別環境で姓に相当するものとして使用される家名/グループ名 (部族名や氏族名など) などの要素、およびその人に関連付けられている称号やその他の名前修飾子で構成されます。完全な個人名は個人を表すものであり、場合によって、社会的階級、宗教的背景、政治的背景、教育水準、民族または国/地域の背景、および地域的由来を示す情報が織り込まれていることがあります。

IBM InfoSphere Global Name Management 個人名モデル

個人名について議論したり作業したりするには、その固有の形式に関わらず、一貫性のある用語を使用することが重要です。また、等価の部分と比較できるように、名前を構成要素となる各部に一貫して解析できることも重要です。

IBM InfoSphere Global Name Management の個人名モデルの形状は、実世界のデータ・セットにおけるエンコードにより名前を処理するという必要性に基づいて作成されました。これは、名前の構造を判別するための実際的なアプローチです。例えば、世界の多くの場所で使用される名前には、西洋の感覚における本物の姓が含ま

まれていませんが、それらの名前も、姓が含まれていることを前提とするデータベースに強制的に取り込まれます。そのため、名前処理を一貫性のあるものにする目的で、IBM InfoSphere Global Name Management では、強制的に 2 フィールドの構造を使用します。名前のさまざまな部分がどのフィールドに属するかを判別する方法の一部に、名前の各部がどの程度の頻度で名フィールドまたは姓フィールドに関連付けられてきたかを調べる方法があります。各フィールドの中で、名前の個々の要素はより大きな単位に解析されます。例えば、姓「de la Salle」は、名前の 3 つの別個の部分としてではなく、1 つの主要な名前語幹と 2 つの接頭部で構成される 1 つの名前句として認識されます。

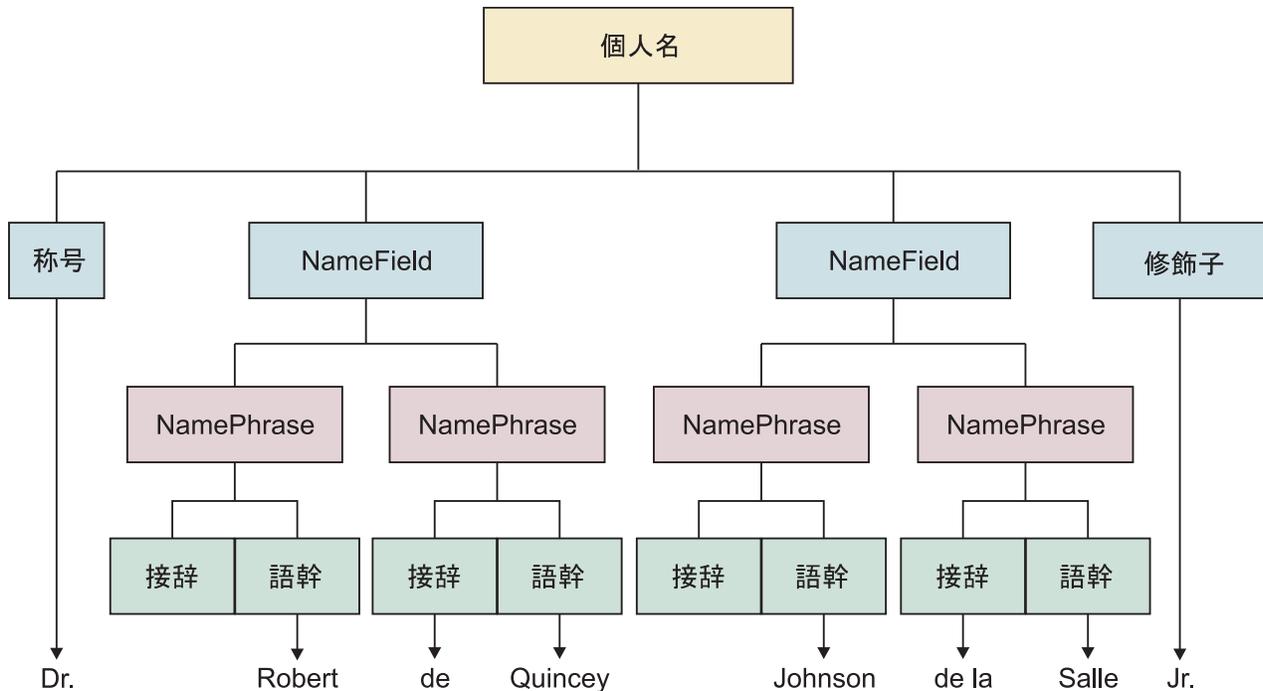


図 1. IBM InfoSphere Global Name Management 個人名モデル

個人名の構造および構成要素

個人名には多くの異なる構成要素が含まれる場合があります。これらの構成要素およびこれらの構造化方法は、国/地域グループによって異なります。

個人名に使用される可能性がある構成要素には、以下のものがあります。

- 名
- 姓
- 家名
- 部族、氏族、またはカーストの名前
- 親戚関係または血統のマーカーとなるもの (父親の名前から取られた名前、母親の名前から取られた名前、子供の名前から取られた名前、世代を示すマーカーなど)
- 出生順序、性別、宗教、または所属宗教を示す修飾子
- タイトル

- 小辞 (アラビア語の「bin」(息子) および「al」(英語の the に相当) またはスペイン語とフランス語の「de」(の/から))

個人名の構造、つまり名前構成要素の順序も、国または地域グループごとに異なります。

名前の構造には次のような例があります。

名 (1 つ以上) + 家名

- Megan Marie Andrews (ヨーロッパ)
- Fereshteh Gholamzadeh (イラン)
- Rattima Nitisaroj (タイ)
- Hasan Incirlioglu (トルコ)

家名 + 名

- Lim Yauw Tjin (中国)
- Pak Mi-Ok (韓国)
- Suzuki Ichiro (日本)

家名 + ミドルネーム + 名

- Trinh Van Thanh (ベトナム)

名 + 父親の名

- Ahmed bin Eisa (一部のアラブ地域)
- Abdurrahman Wahid (インドネシア)
- Mahmud bin Haji Basir (マレーシア)

名 + 父親の名 + 家名

- Ivan Andreyevich Saratov (ロシア)
- Basimah Ali Al-Qallaf (一部のアラブ国家)

部族名 + 宗教名

- WOUKO Philomene (カメルーン)

名のみ

- Sukarno (インドネシア)
- Habibullah (アフガニスタン)

子孫の名前の言及

- Abu Hassan (文字どおりには *Hassan* の父親 と翻訳される。アラブ国家)

等位結合名

等位結合名とは 1 つの名前構造の中の 2 人以上を表し、別個の 2 つの名前が称号で連結されたり (例えば Mr. and Mrs. Smith)、名 (例えば John and Marie Smith)、フルネーム (例えば John Smith and Marie Smith)、その他の名前要素の組み合わせによって連結されたりします。結合子として *and*、*or*、または同等の句読記号 (アンパーサンド (&) など) が使用されます。

等位結合名は、まず個別の名前に解析されます。これにより、それ以降の解析その他のプロダクト機能をそれぞれの個人名に適用できるようになります。IBM InfoSphere Global Name Management 製品 (IBM NameWorks および NameParser) は、次のような 6 種類の等位結合名の構造を認識します。

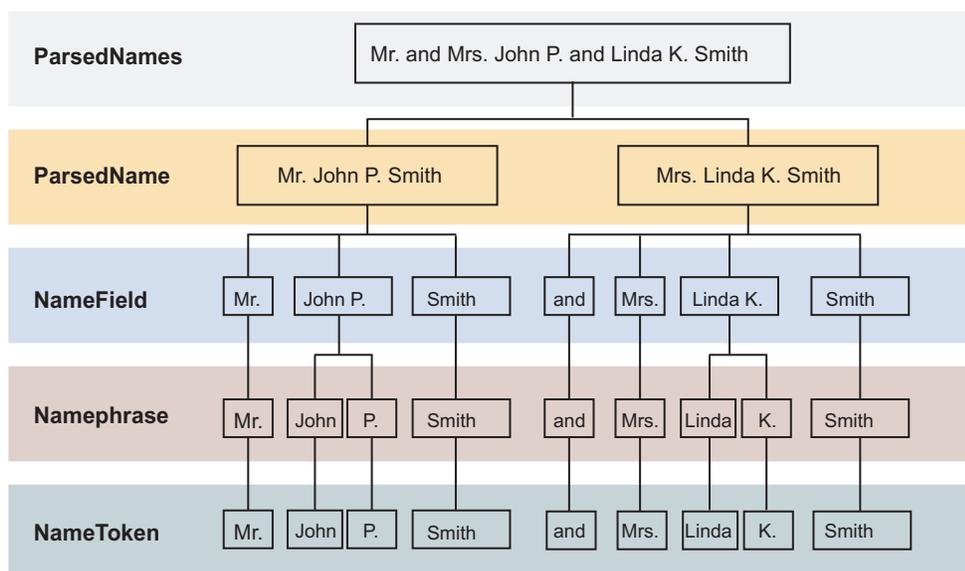
- 等位結合された称号 (例えば *Mr. and Mrs. John Smith*)
- 等位結合された名 (例えば *John and Linda Smith*)
- 等位結合された称号と名のペア (例えば *Mr. John and Mrs. Linda Smith*)
- 並列構造で等位結合された称号と名 (例えば *Mr. and Mrs. John and Linda Smith*)
- 名前全体の等位結合 (例えば *John Smith and Maria Jones*)
- 他の 5 つの等位結合された名前構造の組み合わせ (例えば *John and Linda Smith and Bob and Maria Jones*)

デフォルトでは、IBM InfoSphere Global Name Management 製品の解析機能は、結合子 *and*、*or*、またはアンパサンド (&) の存在によって等位結合名を認識します。これらのどの標識も含まない名前は、等位結合名として扱われません。例えば *John Smith Marie Smith* というストリングは 2 つの別個の名前には解析されません。さらに、コンマ区切りの名前リスト (例えば *Bob, Karen, and David Smith*) は、たとえ結合子が存在しても、適切な構成要素には解析されません。

解析機能は、結合子のように扱うべき文字や単語 (例えば円記号 (¥)、*et*、*y*) のカスタム・リストを受け入れます。外部トークン・リストを使ってこれらの文字を追加することができます。

等位結合名の例

以下の例は、2 つの等位結合された称号および 2 つの等位結合されたフルネームから成る等位結合名構造のために作成された解析ツリーを示しています。まず個別の名前が分離された後、それぞれのフルネームが構成要素に解析されます。



関連概念:

『解析ツリー』

NameParser は、入力した名前の構造パターンおよび分布パターンを分析した結果から解析ツリーを作成します。解析ツリーは、名前のエレメントを構造単位にグループ化する階層であり、個別のトークン (スペースまたは句読点で区切られたストリング) で始まり、結合して名前の句になる場合があり、個人の氏名を形成するように結合します。

組織名

組織名 は、何らかの共通した機能を実行するために存在する、構造化された 1 人以上の集団を指す非人格名です。組織には、会社、クラブ、学校、政府機関、政党、ワールド・ワイド・ウェブ開設団体などがあります。通常、組織名には、それらを非個人名として識別する助けになるある種の標識またはパターン、あるいは単語が含まれます。

組織名には、多くの場合 (ただし常にではない)、その機能を示す単語または句が含まれます。例えば、「高校」、「水道設備」、「警察署」、「銀行」などです。

組織名には、命名要素、つまりこの組織を同種の他の組織とは異なるものとして一意的に識別するストリング (文字、単語、または句) も含まれます。例えば、「第一組合銀行」、「Joe's イタリアン・レストラン」、「AAA 洗車場」などがあります。会社などの一部の組織は、政府の規定で「PTY」や「LTD」などの登録状況を示す所定の名前要素があります。

組織名にあるトークンの種類とトークンの組み合わせは、通常、個人名の場合とは外観やパターンが異なります。これらのパターンは、名前が個人名ではなく組織名として分類された理由を識別するコード (名前カテゴリ理由コードと呼ばれる) に対応します。これらの理由コードが組織名を定義するわけではありませんが、個人名では予期されないパターンを示します。例えば、連続する 3 つの同一の子音で構成されるストリング (「DDD」など) は、個人名では非常にまれですが、組織名では珍しいことではありません。

IBM InfoSphere Global Name Management コンポーネントが名前をカテゴリ化するとき、名前が 1 つ以上の名前カテゴリ理由コードと一致した場合、組織名と想定されます。それ以外の場合、個人名の候補となります。

名前のパーツ

個人名は、言語固有の規則および国/地域別情報に従って構造に結合された 1 つ以上の単語で構成されます。

解析ツリー

NameParser は、入力した名前の構造パターンおよび分布パターンを分析した結果から解析ツリーを作成します。解析ツリーは、名前のエレメントを構造単位にグループ化する階層であり、個別のトークン (スペースまたは句読点で区切られたストリング) で始まり、結合して名前の句になる場合があり、個人の氏名を形成するように結合します。

統計的分布パターンに基づいて名前句を名フィールドおよび姓フィールドに分割する処理の際、名前に 2 フィールド・データ構造が強制的に使用されます。ツリーの

ルートにあるオブジェクトは、入力ストリング全体を表します。その子は名前を最も大まかに分けたサブディビジョンを示し、さらにその子はサブディビジョンのサブディビジョンを表し、以降同様に階層が続いていきます。

以下の例は、かなり複雑な名前の構造要素を示しています。個々の名前トークンを示す一番下の行から始まり、レベルが 1 つ上がると名前句が取り込まれ、さらに 1 つレベルが上がると名前フィールドへの分割が示されます。元の名前形式はツリーの最上部に **ParsedName** として示されます。この解析ツリーは、NameParser によって認識される構造を表しています。

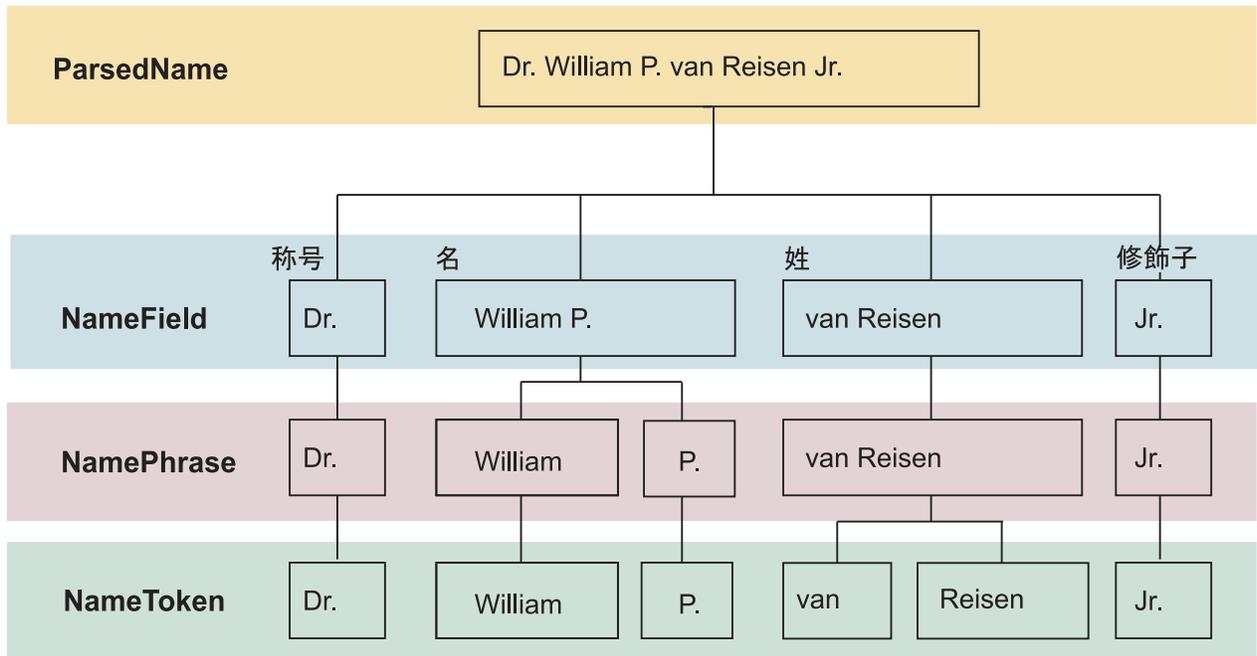


図 2. NameParser 解析ツリーの例

関連概念:

16 ページの『等位結合名』

等位結合名とは 1 つの名前構造の中の 2 人以上を表し、別個の 2 つの名前が称号で連結されたり (例えば Mr. and Mrs. Smith)、名 (例えば John and Marie Smith)、フルネーム (例えば John Smith and Marie Smith)、その他の名前要素の組み合わせによって連結されたりします。結合子として *and*、*or*、または同等の句読記号 (アンパーサンド (&) など) が使用されます。

『解析した名前』

名前解析は、名前に含まれるトークンを、トークンを含んだより大きな構造単位に編成するプロセスです。名前句のようなこれらの一部の構造は、名前の元になる言語内に見い出される本来の文法構造です。名前解析は、統計情報と、名前の中の統語構造を認識するための言語学ベースの規則の両方を使用して実行されます。

解析した名前

名前解析は、名前に含まれるトークンを、トークンを含んだより大きな構造単位に編成するプロセスです。名前句のようなこれらの一部の構造は、名前の元になる言

語内に見い出される本来の文法構造です。名前解析は、統計情報と、名前の中の統語構造を認識するための言語学ベースの規則の両方を使用して実行されます。

検索や突き合わせのような高水準の名前処理操作を使用すると、名前の各部分はその相対的な情報値にしたがって処理されるときに最良の結果が戻ります。つまり、名前語幹のような内容値が大きいトークンは、検索の際、接頭部や接尾部のような文法的小辞、または名前本体とは別に付随する称号のようなトークンよりも多くの重みが付与されます。同様に、名または姓を表す名前の各部分は、他の名または姓と並行して処理する必要があります。名前のさまざまな部分が果たす役割を判別するために、名前を解析する必要があります。

名前フィールドのようなその他の名前構造は、国/地域別コミュニティまたは言語コミュニティで認識されるセマンティック構造または社会構造に対応する可能性がある (対応しないこともある) 人工的なデータ構造です。例えば北米文化圏では、多くの人に名、ミドルネーム、および姓があります。 *Karen Lee van der Meer* のような名前は、次の構成要素で構成されます。

- 語幹 *Karen* で構成される、名の役割を果たす名前句
- 名前語幹 *Lee* で構成される、ミドルネームの役割を果たす名前句
- 2 つの接頭部と語幹 *Meer* で構成される、姓の役割を果たす名前句 *van der Meer*

ミドルネームのさまざまな位置付けは認識されないため、この名前は 2 フィールド・データ構造にはぴったりとマップしません。

IBM InfoSphere Global Name Management 製品では、階層的解析構造を使用しています。この構造では、トークン (スペース区切りまたは句読点区切りの文字ストリング) は名前句という本来持っている構造にグループ化され、さらにそれが、名前句および国/地域固有の名前パターンの統計的分布に基づいて、名フィールドと姓フィールドの要素にグループ化されます。2 つの名前フィールドに、入力した元の名前中出现するタイトルまたは修飾子を加えたものが、氏名になります。

関連概念:

18 ページの『解析ツリー』

NameParser は、入力した名前の構造パターンおよび分布パターンを分析した結果から解析ツリーを作成します。解析ツリーは、名前のエレメントを構造単位にグループ化する階層であり、個別のトークン (スペースまたは句読点で区切られたストリング) で始まり、結合して名前の句になる場合があり、個人の氏名を形成するように結合します。

名前フィールド

名前フィールド とは、データ処理を容易にするために名前に対して定められる人工的データ構造のことです。多くのデータベースでは名前を 2 つのフィールド (通常は名と姓に対応するフィールド) に分割しますが、名前を 1 つのフィールドに入力するものや、3 つ以上のフィールドを使用するものもあります。名前フィールドは個人名解析ツリーの最初のブランチです。

IBM InfoSphere Global Name Management 製品は、個人名構造として 2 つのフィールド (名および姓のラベルが付く) を使用し、称号および修飾子のために 2 つの追加フィールドがあります。各フィールドには次のような独自のフィールド・タイプ・インディケーターがあります。

姓 姓 (あるいは SN) は、通常 (ただしそうではない場合もある)、人のグループ (家族、部族、カーストなど) が共通して持つ、名前の一部分です。世界のある場所では、姓は各個人に固有のもの (個性や職業を示すものなど) である場合があります。

姓は、個人名の主要な内容保持要素です。ただし、すべての人が姓を使用するとは限りません。例えばインドネシアの一部では、ほとんどの人には名しかありません。

名 名 (あるいは GN) は、同じ家族またはグループの他のメンバーとは異なったものとして個人を一意的に特定する、名前の一部分です。英語名では、ファーストネームとミドルネームが名です。

名は、世界中のすべての国/地域別環境において普遍的な命名要件として知られている唯一の名前要素です。すべての国/地域別環境において姓要素があるとは限りませんが、すべてにおいて名が個人に割り当てられます。

タイトル

タイトルは、名前自体とは別に付けられる単語または句ですが、名前の所有者に関するある種の情報を伝えます。タイトルは、結婚状況、出生順序、学歴、専門職、宗教的立場、社会的地位、その他の情報を示す場合があります。タイトルは任意指定になる傾向があり、名前に常に付帯するとは限らないため、中核となる名前要素とタイトルでは処理方法が異なります。

修飾子

修飾子は、個人名を区別するためにその名前に追加される用語または句で、世代順序 (Junior や Senior など。フランス語では Junior は「fils」)、その人が獲得した功績や名誉 (例えば、Ph.D.)、あるいはある種の資格 (D.D.S. など) を指定します。修飾子は、通常、名前の後に置かれます。タイトルと同様、修飾子は名前に付帯するものですが、個人名の一部とは見なされません。

先行結合子

名前に等位結合名の構造が含まれる場合、その構造の最後の要素の前には、入力ストリング内の名前をその前にある名前に結合する結合子が必要です。例えば、入力ストリング *John and Mary Smith* は、*John Smith* と *Mary Smith* という 2 つの名前を表す等位結合名の構造です。名前 *Mary Smith* には、*and* という先行結合子があります。NameParser では、この結合子を独自のフィールドに保持できます。

NameHunter 検索エンジンは、タイトルを名フィールドに、修飾子を姓フィールドに取り込みます。これらの非名前要素は、検索および突き合わせの各操作の際、NameHunter 検索パラメーターの構成方法にしたがって異なる方法で処理されます。それに比べて類似製品では、非名前要素用に別個のフィールドを保持しています。

名前句

名前句は、言語の名詞句または前置詞句に類似した、単一の言語構造で構成される 1 つまたは一連のトークンです。名前句は、1 つ以上の名前語幹と、その語幹に関連した接頭部、接尾部、結合子、またはその他の文法的要素で構成されます。

例えば、スペイン語の名前句 *de la Cruz* には、1 つの名前語幹 (Cruz) と 2 つの接頭部 (de および la) が含まれます。中国語の名前句 *Mei-hui* には 2 つの語幹が含まれ、英語の名前 *Smith* は単一の語幹だけで構成される名前句です。名前句は、中間レベルの解析ツリーの構成要素で、個々のトークンをより大きな構造にグループ化します。名前フィールドには複数の名前句が含まれる場合があるため、名前句が必ず 1 つの名前フィールドになるわけではありません。

IBM NameWorks および NameParser は、世界のほとんどすべての国における 8 億以上の名前に基づくオリジナル・データ・セットから得られる、名前句に関する内部情報リポジトリを使用します。このリポジトリは、解析、変形形式の生成、分類、および検索を含むすべての名前処理操作において情報を提供します。

名前トークン

名前トークン とは、「空白文字」または文字の句読区切りストリングで構成される、これ以上分割できない名前の最小要素をいいます。

たいていの場合、名前トークンは接辞または語幹となりますが、名前のより小さい文法単位が単一の単語として記述される場合には、名前トークンは氏名句になる場合もあります。例えば、氏名句 *de la Cruz* は、接辞 *de* および *la* と、語幹 *Cruz* で構成されます。名前トークンは、解析ツリーの最後のレベルで、リーフ・ノードと呼ばれることがあります。名前トークンの機能は、その内容、および名前に含まれる他の要素との相対位置に依存します。例えば *de* のようなトークンは、それが出現する名前の言語的起源、およびそれが名前のどの位置に置かれるかに応じて、接頭部、名、または姓になる場合があります。

名前トークンは、単一トークンまたは複数トークンのいずれかの構成にできます。例えば、複数トークンの用語には *Limited Liability Corporation* があります。

名前トークンのタイプには、次のものが含まれます。

- 接頭部
- 接尾部
- タイトル
- 修飾子
- 組織指定子
- 職業修飾子
- ストップワード
- 組織接辞
- イニシャル
- 結合子
- 名前語幹
- 用語

タイトル、接辞、および修飾子 (TAQ)

IBM InfoSphere Global Name Management 製品は、称号、接辞、および修飾子 (集合的に TAQ という) のスコア処理に、特殊ロジックを使用します。タイトルおよび修飾子は、多くの場合、名前のうち特に不可欠というわけではない任意指定の要素です。接辞は、文法的には名前の一部ですが、内容的価値はあまりないため、名前比較演算では重みが少なく付与されます。

タイトル

タイトルは、人に呼びかけるときの用語で、多くの場合、名前の前に置かれ、礼儀、社会的立場、または職業的地位を示す場合があります。

タイトルの例には、以下のものが含まれます。

- Dr. (博士)
- Mr. (氏)
- Hajj (メッカに巡礼をしたことがあるイスラム教徒)

接辞

接辞は、名前に付く接頭部または接尾部です。文法的には名前の一部ですが、通常は内容的には大きな価値がないため、2 つの名前を比較するときには重みが少なく付与されます。

個人名の接辞の例には、以下のものが含まれます。

- de la Torres の *de la*
- van der Meer の *van der*
- Abdul Rahman の *Abdul*
- Nur Al Din の *Al Din*

修飾子

修飾子は、個人名を区別するためにその名前に追加される用語または句で、世代順序、その人が獲得した功績や名誉、あるいはある種の資格を指定します。通常、修飾子は名前の後に置かれ、一般的に実際の名前の一部とは見なされません。

修飾子の例には、以下のものが含まれます。

- Jr. (ジュニア)
- Sr. (シニア)
- Esq. (殿)
- PhD (博士号)
- D.D.S (歯科学博士)

組織名 TAQ

組織名だけに出現する用語については、特殊処理が必要になります。これらの用語には、一貫性を持たせるために TAQ というラベルが付きます。組織名にはタイトルはありませんが、特定の修飾子および接辞があります。

これらのタイプのトークンは、組織名において意義のある要素であるとは見なされません。そのため、NameHunter 比較において名前語幹のように処理されないように、以下のすべての TAQ が TAQ ファイル (taq.ibm) にリストされています。これらの用語は、TAQ ファイルで割り当てられる以下のデフォルトの TAQ 係数の影響を受けます。

埋め込み空白文字が含まれる組織用語を認識するため、複数トークンの用語がサポートされています。例えば、Limited Liability Corporation などです。

表 2. 組織名 TAQ およびデフォルトの TAQ 係数の例

TAQ タイプ	例	相違 TAQ 係数 (デフォルト)	欠落 TAQ 係数 (デフォルト)
組織指定子 (OD)	COMPANY、 CORP、 LLC、 LIMITED LIABILITY CORPORATION、 COUNTRY CLUB	.98	.99
ストップワード (SW)	OF、 THE	1.0	1.0
組織接辞 (OA)	DE、 DEL、 LA、 LAS	.97	.98

組織指定子:

組織指定子 (OD) は、組織の法的登録に関する情報を提供するものですが、一般的には組織名の主要な部分ではありません。そのため、OD はしばしば完全に無視され、異なる形式で表示される場合もあります。

例えば、以下のすべての名前は同じ組織を指します。

- 20TH CENTURY PRODUCTIONS INCORPORATED
- 20TH CENTURY PRODUCTIONS INCORP
- 20TH CENTURY PRODUCTIONS INC
- 20TH CENTURY PRODUCTIONS

NameHunter でこのリストにある最後の 2 つの名前を比較するとき、欠落 TAQ 係数が適用され、本来は完全に一致しているのに .99 のペナルティーが割り当てられます。ただし最初の 3 つの名前の比較では、相違 TAQ 係数に影響を受けることはありません。なぜなら、これらの OD は互いに等価であるからです (INCORPORATED、INCORP、および INC はすべて同じです)。実際、ほとんどの OD には、交換可能な変形形式があります。例えば、以下の OD は互いに類義語です。

- CO/COMPANY
- JSC/JOINT STOCK COMPANY
- LLC/ LIMITED LIABILITY COMPANY

OD の変形の認識は、照会名とデータ・リスト名の両方に適用される一連の正規化規則によって処理されます。それぞれの規則によって OD の変形形式が事前定義された正規化形式に変換され、これらの正規化された形式がすべて TAQ ファイルに組み込まれます。例えば、正規化規則によって INC および INCORP が INCORPORATED に変換された後、それが TAQ ファイルにリストされます。前

述のリストの最初の 3 つの名前は、その後 20TH CENTURY PRODUCTIONS INCORPORATED に正規化されるため、比較の際に 1.0 のスコアを受け取ります。このソリューションは、あいまいさを減らし、複数トークン TAQ を処理するために採用されました。

あいまいさ

OD の中でも意義のある名前要素であるものが多数あります。例えば、LLC はしばしば LIMITED LIABILITY COMPANY を表す OD になりますが、LLC COUNSELING SERVICES のような組織名もあり、この LLC は、この特定の組織を同じタイプの他の組織 (例えば、FRANKLIN COUNSELING SERVICES) と区別する名前語幹になります。

LLC が単に TAQ ファイルに含まれている場合には、NameHunter 比較の際に LLC が出現すると、すべて TAQ として処理されてしまいます。そうすると、LLC COUNSELING SERVICES のような名前は、データ・リスト内にある句、COUNSELING SERVICES が含まれる他のすべての名前と一致してしまいます。正規化を行うと、LLC のような用語の位置を認識する機能が提供されます。つまり、そのような用語が名前の末尾に出現した場合には OD である可能性が高いため、正規化規則によって LLC = LIMITEDLIABILITYCOMPANY と判定されます。この用語が名前の先頭に出現した場合には、この用語はおそらく名前語幹です。

複数トークン TAQ

多くの OD には、複数のトークンが含まれています (例えば LIMITED LIABILITY COMPANY)。しかし検索エンジンは、複数トークン項目を含む TAQ および変形ファイルをサポートしません。正規化を実行すると、複数トークン・ストリングを、TAQ ファイルでリストできる単一のトークンに変換する機能が提供されます。正規化を実行することにより、以下の名前を検索エンジンで比較すると、1.0 の完璧な比較スコアを受け取ります。

元のストリング	正規化したストリング
BISON LEGACY LLC	BISON LEGACY
BISON LEGACY LIMITED LIABILITY COMPANY	LIMITEDLIABILITYCOMPANY

組織名においては、NameHunter は、正規化された照会名と正規化されたデータ・リスト名の間のみで比較します。元の照会名とデータ・リスト名は比較されません。以下の例では、照会名は OD 正規化規則の影響を受けません。LLC が名前の先頭に出現するためです。データ・リスト名では LLC が正規化されます。これが名前の末尾に出現するためです。

照会名	データ・リスト名
LLC CONSULTING SERVICES	BEVERLY CONSULTING SERVICES LIMITEDLIABILITYCOMPANY

この比較では、ストリング LLC と BEVERLY は一致せず、LIMITEDLIABILITYCOMPANY がトリガーとなって TAQ ペナルティーが適用さ

れるため、低い比較スコアが戻ります。この結果は正しいと言えます。これらの組織は同じではないからです。しかし、元の形式も検索可能として維持されている場合には、以下の名前が比較されます。

照会名	データ・リスト名
LLC CONSULTING SERVICES	BEVERLY CONSULTING SERVICES LLC

TAQ ファイルには LIMITEDLIABILITYCOMPANY のような正規化された形式しか含まれず、LLC のような元の形式は含まれていません。そのためこの場合、どちらの LLC も TAQ とは認識されず、2 つの LLC は語幹として比較されます。この比較では、4 つのトークン中 3 つが同一であるため、比較的高いスコアが戻ります。正規化されていない組織名の比較を防ぐことにより、この問題を回避できます。

正規化によって処理される OD の差異の他に、差異が変形によって処理される事例がまれにあります。INC/INCORPORATED のようなペア (完全なスコアを受け取る、互いに等価のペア) とは異なり、等価というよりは変形に近い他のペアがあります。これらの場合には、変形のペアは、スコアおよび国/地域別情報 (これは組織名では常に 0 です) を割り当てた上で TAQ ファイルにリストされます。この例には、COMPANY, COMPANIES, .99, 0 のような項目があります。この変形セットがあるため、ストリング COMPANY と COMPANIES が含まれるペアでは、相違 TAQ 係数を適用する場合より高い比較スコアを受け取ります。

ストップワード:

ストップワードは、組織名に何ら意味を付与しない単語であるため、どの名前比較および名前スコア処理にも含まれません。IBM InfoSphere Global Name Management の現行リリースには、ストップワードとして OF および THE だけが含まれています。

ストップワードのデフォルトの TAQ 係数値は 1.0 で、これにより、名前比較の際にストップワードが無視されたかのようにになります。そのため、以下の名前セットを比較するときに完全なスコアが得られます。

表 3. ストップワードが含まれる名前セット

組織名	ストップワードが含まれる組織名
ACME INKS CANADA INC.	ACME INKS OF CANADA INC.
TRACTOR & ENGINEERING CO.	THE TRACTOR & ENGINEERING CO.
ASSOCIATION OF AMERICAN GEOGRAPHERS	THE ASSOCIATION AMERICAN GEOGRAPHERS

組織接辞:

組織接辞 (OA) は、組織名に事実上何の意味も付与しない単語です (類似例としてストップワードあります)。しかし、OA の有無は、特に他の言語ではストップワードより重要な意味を持つ傾向があるため、OA に相違があるか欠落している場合に TAQ 係数では小さいペナルティを割り当てます。

OA には、英語の接続詞 AND や、数値や性別の情報を伝える英語以外のいくつかの冠詞が含まれます。以下の名前ペアなどのようにこれらのトークンがない場合、欠落 TAQ 係数が適用されます。異なる OA が比較された場合、相違 TAQ 係数が適用されます。

表 4. 組織接辞が含まれる組織名

組織名	OA が含まれる組織名
AAI ENGINEERING SALES SERVICE	AAI ENGINEERING SALES AND SERVICE
BUTCHER HAUS	DAS BUTCHER HAUS
BANCO ORO	BANCO DE ORO

語幹トークン

語幹トークン (名前語幹ともいう) は、独立して、または接辞と結合して名前句を形成できる名前要素です。例えば、名前句 *de la Torres* は、名前語幹 *Torres* と、その前の 2 つの接辞で構成されるヒスパニック系の姓です。

関連付けられた接辞がない名前語幹 (*Robert* や *Gonzales* など) は、それ自体が名前句です。多くの場合、1 つの名フィールドまたは姓フィールドに 1 つ以上の名前句が含まれます。国/地域別環境が韓国語と中国語で名フィールドに含まれる名前句の大部分、およびフランス語、ドイツ語、ヒスパニックで一般的な名の大部分は、「*Shu Dong*」、「*Eun Jung*」、「*Jean Luc*」などのように 2 つの名前語幹から構成されます。

名前リスト

IBM InfoSphere Global Name Recognition 製品は、いくつかのタイプの名前リストを使用して名前を処理します。

名前データ・アーカイブ

名前データ・アーカイブは、世界中の 10 億個近い名前を集めたコレクションであり、各名前に関連付けられた性別や国の情報も含まれています。名前情報を収めたこの大規模なリポジトリは、名前のカテゴリー化、分類、解析、性別識別、および突き合わせを行うために IBM InfoSphere Global Name Management 製品で使用するアルゴリズムおよび規則によって利用されます。

名前データ・アーカイブから取得する個別の名前トークンおよび名前句の頻度カウントは、IBM InfoSphere Global Name Management 製品が名前を分析するために使用する統計アルゴリズムおよび計算アルゴリズムの基礎となります。例えば名前解析コンポーネントは、これらの統計を使用して、名フィールドと姓フィールドの特定の組み合わせの妥当性スコアを計算します。妥当性スコアが低い場合、名前のフィールド分割に誤りがあった可能性があります。この場合、解析エンジンによって、代替案となるより妥当な組み合わせが 1 つ以上提示されます。

外部トークン・リスト

外部トークン・リストは、IBM InfoSphere Global Name Management 内部データベースの情報を補足するために称号、接辞、修飾子、または名前語幹を追加でき

るファイルです。名前の分析またはスコアリング中に、内部データベースより前にカスタム・トークン・リストが検索されます。

デフォルトでは、NameParser 外部トークン・リストおよび IBM NameWorks カスタム・トークン・リストはどちらも空です。IBM NameWorks のカスタム・トークン・リストには、構成ファイルを通してアクセスできます。

関連資料:

NameParser の外部トークン・リスト

トークンを外部トークン・リストに追加して、NameParser の動作をカスタマイズできます。外部トークン・リストは、トークン・データの補足リストであり、NameParser がトークン・データのこの補足リストでトークンを検索します。トークンが見つからない場合、NameParser は数百万の名前トークンについての情報を含む独自の内部表を検索します。

名前音訳

名前音訳 は、特定の書記体系または文字エンコード規則から別の書記体系または文字エンコード規則に名前を変換するプロセスです。例えば、名前音訳を行うことによって、アラビア文字で書かれた名前を分析して、ラテン・アルファベットで書かれた類似の名前と突き合わせるすることができます。

音訳 は翻訳 と混同されることがあります。翻訳とは、話された事柄や書かれた事柄の意味をある言語から別の言語に伝えることです。例えば、英語の「a horse」はフランス語の「un cheval」になります。音訳とは、ある表記法 (記述システム) で表された音声を、別の表記法で表すことによって伝えることです。そのため、馬を意味する北京語の単語を発音するときに使用される中国語の文字は、ローマ字の「ma」に音訳されます。

IBM InfoSphere Global Name Management 製品には、アラビア語、ギリシャ語、キリル文字、デーバナーガリー文字、仮名文字、ハングル文字、漢字、および拡張ラテン語を含む多くの書記体系における名前突き合わせをサポートする組み込みサポートがあります。この拡張ラテン語とは、多くのヨーロッパ言語およびアジア言語で使用される追加文字および発音区別符号を付け加えた標準ラテン文字です。

名前は、名前の分析処理およびスコア処理の最初のステップとして音訳されます。NameTransliterator は、名前の解析、分類、またはスコア処理の前に、前処理ステップとして、名前をその固有のエンコード方式から ASCII エンコード方式に変換します。IBM NameWorks が搭載する多くのメソッドは、音訳を最初に実行してからその他のメソッド機能を実行するように設計されています。

漢字からかなへの音訳サポートが別個の Java パッケージで提供されています。これは、IBM NameWorks の Java インターフェースまたは Web サービス・インターフェースと組み合わせて使用できます。com.ibm.gnr.java パッケージについては、Java API Reference セクションに説明があります。

音訳規則ファイル

Transliterator 規則ファイルは、特定の書記体系またはアルファベットの処理を可能にする、暗号化されたバイナリー・ファイルです。各規則ファイルを使用すると、特定のスクリプトでの入力を IBM InfoSphere Global Name Management コンポ

ーネントで処理できるようになります。テキストは、名前の分析やスコアリングに適した形式で、大文字の ASCII 文字に変換されます。

以下の規則ファイルまたはモジュールを使用して、基本的な名前音訳ツールの機能を拡張できます。

arabicTransRule.ibm

アラビア文字で記述された個人名用。

chineseTransRule.ibm

中国語の漢字で記述された個人名用。

chineseOnTransRule.ibm

中国語の漢字で記述された組織名用。

cyrillicTransRule.ibm

キリル文字で記述された個人名用。

cyrillicOnTransRule.ibm

キリル文字で記述された組織名用。

greekTransRule.ibm

ギリシャ文字で記述された個人名用。

hindiTransRule.ibm

デーバナーガリー文字で記述された個人名用。

hindiOnTransRule.ibm

デーバナーガリー文字で記述された組織名用。

japaneseTransRule.ibm

仮名 (カタカナとひらがな) で記述された個人名用。

japaneseOnTransRule.ibm

漢字または仮名 (カタカナとひらがな) で記述された組織名用。

koreanTransRule.ibm

ハングル文字で記述された個人名用。

koreanOnTransRule.ibm

ハングル文字で記述された組織名用。

latinTransRule.ibm

基本ラテン文字またはラテン語の補足や拡張のいずれかで記述された個人名および組織名用。このモジュールは *NameWorks* 内に組み込まれているので、構成ファイルで参照する必要はありません。

中国語の音訳についての概要

IBM InfoSphere Global Name Management は、中国語の漢字の名前をその正確な同等の形式で突き合わせ、漢字と音訳された対応するローマ字名を相互に突き合わせます。

書き言葉では、中国語と日本語には多くの共通文字があります。漢字で書かれた中国語の名前は日本語漢字の名前に類似していて、多くの共通文字があります。中国語漢字の名前には、日本語漢字の名前とは異なる次のような特徴があります。

- 中国語の姓が複数文字になることはほとんどなく、1 つの姓と 1 つの名を解析するだけでよいので、あいまいな中国語氏名はほとんどありません。
- 現代北京語では、読みが複数ある文字はほとんどありません (声調の差異は除く)。
- 複数の読みがある文字であっても、ほとんどの場合、個人名として使用されるときに最も一般的な読みがあります。姓に固有の発音を持つ文字のセットはわずかです。そのため、中国語の個人名用の発音解析支援機能は、通常の使用条件で提供されることはあまりありません。

これらの特徴から分かるのは、北京語のほとんどすべての中国語名には 1 つの発音しかありません。そのため、中国語の名前は、NameTransliterater コンポーネントの中で直接音訳できます。この点、日本語の名前は異なり、漢字名とローマ字形式の間に多対多の関係が存在することがよくあります。

International Components for Unicode (ICU) のオープン・ソース・プロジェクトには、一般的に使用される中国語文字を北京語ピンイン表記に音訳するシステム規則セットがあります。各文字の出力形式は 1 つだけです。複数の発音がある文字の場合は、最も一般的な発音が選択されます。IBM InfoSphere Global Name Management の音訳処理では、ほとんどの中国語文字で ICU の内部ルール・セットを使用します。例外については特別の規則で処理されます。

中国語の名前の処理を行うには、音訳規則を追加する以上の作業が必要になります。北京語のすべての中国語文字は単音節です。つまり単一の音節として発音されます。北京語には約 1,760 の可能な音節があります。声調付きの音節が約 1,350、声調なしの音節が約 410 あります。しかし、中国語の文字は何万個もあり、それぞれに異なる意味があります。つまり、意味の異なる何十もの別々の文字が厳密に同じ発音となっています。その結果、異なる文字で書き表される名前が同じローマ字形式に音訳されることがあります。ローマ字化は意味ではなく発音に基づいて行われるためです。言い換えると、中国語文字の名前とローマ字形式の間には、多対 1 の関係があります。問題となるのは、照会時の名前が中国語文字の名前であり、データ・リストに発音が同じであるため同じローマ字形式に音訳され、かつ異なる中国語文字の名前が含まれている場合です。追加のフィルター手順を用意しなければ、中国語文字が異なるこれらの名前は名前検索で完全一致として返されることとなります。

次の 5 つの異なる名前を示すリストを考えてみます。それぞれには、他の名前とは異なる文字が少なくとも 1 つあります。

1. **黄书东** - 簡体字の文字セットで書かれた名前です
2. **黃書東** - (1) と同じ名前ですが、繁体字の文字セットで書かれています
3. **黄书冬** - 名の最後の文字が異なります
4. **皇书东** - 姓の文字が異なります
5. **皇舒冬** - 文字がすべて異なります

これらの名前はすべて同じローマ字形式である「HUANG SHU DONG」(数値声調記号を含める場合には HUANG2 SHU1 DONG1) に音訳されます。しかし、(1) と (2) の名前だけが同じ中国語名です。これらのローマ字形式がすべてデータ・リスト

に含まれている場合には、(1) の "黄书东" を照会すると、(3)、(4)、(5) も 1.0 で返されます。しかしネイティブ・スピーカーにとってはこれらはすべて異なる名前です。NameHunter の検索プロセスは、この種の問題を扱うように拡張されています。

中国語漢字の名前データの処理

NameHunter 機能は、中国語漢字の名前データを、以下の汎用プロセスを使って分析します。

- 漢字の名前音訳は、NameHunter の外部で行われます。漢字の名前は、NameHunter に送られる前に音訳される必要があります。
- NameHunter には、ローマ字の対応する名前と元の漢字名のデータの両方が必要です。
- NameHunter は、最初にローマ字形式の名前を比較してから、一致する可能性のある元の漢字文字を比較することにより、偽陽性 (これは、多対 1 の漢字とローマ字のマッピングによって作成される場合がある) を除去します。

次の機能があります。

- 中国語の漢字で記述された個人名の名と姓の要素を認識し、それらの要素を適切に処理する。
- 同じ漢字文字の変形形式を一致させる。
- 漢字の個人名と対応するローマ字名を突き合わせる。

中国語のスコアリングは、合格か不合格かに基づいて適用されます。つまり、両方の名前に同じ漢字文字 (または同じ文字の変形形式) が含まれれば合格で、そうでなければ比較は不合格となります。漢字スクリプトの比較が合格になると、ローマ字名マッピングのために生成されるスコアが使用されますが、ローマ字のスコアが 1.0 で、漢字のスコアが文字変形から生じた不一致の値のために 1.0 未満である場合は処理が異なります。この場合、ローマ字スコアに -.02 のペナルティーが適用されて .98 になります。これは、漢字の名前形式が同一ではなく、代わりに同じ文字の変形が含まれていることを示します。

スコアリング・アルゴリズムで使用する中国語異形テーブルには、簡体字と繁体字の対照も他の異形とともに含まれています。この異形スコアで最大のものは .995 です。このテーブルのフォーマットは、その他の NameHunter 異形テーブルに似ており、拡張可能です。例えば、本当の変形ではないものの互いに関連している (つまり発音が同じで画数も類似している) 文字セットを追加して、まったく異なる文字によって一致が失敗することを回避できます。

通常の使用条件では、中国語の姓と名は区切られません。居住申請用紙のような構造化された名前データであっても、1 つの氏名欄しかないのが一般的です。音訳規則ファイルには、解析アルゴリズムが組み込まれているため、未解析の中国語文字の氏名は音訳の前に解析されて姓と名に分けられます。この解析は、多言語間名前

処理では不可欠で、最も一般的な発音には従わない、ごくわずかの例外的な姓の文字に対しても正しいローマ字形式を得るために役立ちます。

中国語漢字の名前データ分析には、次の制約があります。

- 中国語の音訳は、北京語ピンインだけを生成します。
- 中国語文字間の比較は、それらのローマ字化形式が事前定義されたしきい値で一致する場合にのみ可能です。
- 類似した漢字文字を使用しているても発音が異なる名前は、(発音に基づく) ローマ字形式が一致する可能性が小さいため、初期突き合わせで合格する可能性はほとんどありません。そのような文字を文字異形テーブルに追加する操作は、中国語文字の直接検索と比較がサポートされていないため、効果がありません。

中国語の音訳には、`chineseTransRule.ibm` と `chineseOnTransRule.ibm` の両方のファイルが必要です。最初のファイルは個人名用、2 番目のファイルは組織名用です。本製品の以前のリリースからマイグレーションする場合には、`Distributed Search` および `NameWorks` の構成ファイルの更新が必要です。

日本語の音訳についての概要

ネイティブの日本語の名前は、日本で漢字と呼ばれる、漢字文字セットで記述されます。これらの文字の大半は、中国語の記述に使用されるものと同じです。

北京語など、中国語のいずれかの方言では、単一の漢字文字には 1 つの発音つまり読み方があります。しかし日本語では、1 つの文字または一連の文字に 1 つの読み方があるわけではありません。同じ漢字文字が異なる名前の記述に使用されていて、その読み方は全く異なる場合があります。このため、漢字の個人名を含むレコードには、どの読み方を使用するかを明確にするために、その名前の仮名によるスペルも含まれている場合があります。

仮名は日本で使用される発音ベースの記述システムで、アルファベットと似ています。各仮名記号は、通常は子音と母音の組み合わせとなる、別個の 1 つの音節を表しています。仮名にはひらがなとカタカナの 2 つの「スタイル」があり、どちらも GNM でサポートされています。

日本での外国名は仮名で記述されます。

GNM は、漢字の個人名と組織名を異なる方法で扱います。漢字の個人名は、入力ストリングに対して複数の仮名読みを生成する、特殊な Java 専用の `Kanji Transliteration` コンポーネントによって処理する必要があります。その後、それらの仮名読みの 1 つまたはすべてを GNM に渡して、処理を続けます。このアプローチにより、ローマ字化した日本語の名前との最も広範な一致が可能になります。

漢字の組織名は、GNM の音訳モジュールを介して直接渡されます。各漢字文字は、ローマ字のアルファベットで表される 1 つの読みに一貫してマップされます。

中国語と日本語で同じ漢字文字セットが使用されるので、日本語の名前は日本語の読みで音訳が行われるように、日本の国/地域別情報のマークを付ける必要があります。漢字文字によるすべての入力ストリングに対するデフォルト解釈は、その名前が中国語であるというものです。国/地域別情報が指定されていない漢字文字のすべての名前は、中国語の音訳モジュールを介して渡されることとなります。

第 3 章 名前の解析

名前の解析は、名前の構成要素を識別すること、それらの要素を名フィールドと姓フィールドに分離すること、および称号 (Mr.、Hajj、など) や修飾子 (Jr、Esq.、など) などの名前以外の要素を認識することから構成されます。名前の解析は、名前の分析とスコアリングの重要な要素です。解析が正確であれば、名前の各要素が正確に分析される可能性が高くなり、より正確な検索結果が生成されます。

NameWorks を使用した名前の解析

名前を解析して構成要素に分けることは、名前の分析とスコアリングの重要なステップです。NameWorks は、音訳を名前の解析に使用するメソッドに組み込んでいます。

名前を個々の部分に解析する

`parse()` メソッドを使用して、1 つ以上の名前を音訳して構成パーツに分解します。このメソッドは、入力ストリングに含まれる各名前の解析ツリーを返して、各名前の代替解析に関する情報を提供します。さらに、国/地域に固有の試行錯誤法と構成要素となっている名前句 (名と姓) の相対頻度に基づいて、解析が正しい可能性を表すスコアを返します。

このタスクについて

NameWorks の Analytics クラスの `parse()` メソッドを使用して、個人名を名と姓の構成要素に解析し、さらに名前から称号と修飾子を分離します。有効にされると、信頼性スコアが十分高ければ、代替解析に関する情報が NameWorks により返されます。例えば、オリジナル検索照会が名前 `David, Robert` を対象としている場合、NameWorks は代替解析の `Robert, David` を返す場合があります。両方の名前トークンが、同等の可能性で名にも姓にもなるからです。NameWorks が代替解析を返すためには、`alternateThreshold` 値を満たさなければなりません。名前の解析を開始するには、`parse()` メソッドに以下の値を渡します。

- 解析する氏名。フルネームで表します。例えば、`ROBERT E JONES` のようにします。名前は、`parse()` メソッドにストリング 値として渡す必要があります。
- `alternateThreshold` 値を表す 0 から 100 までの整数 (0 を渡した場合、代替解析は常に抑止されます)。この値は、初期解析で名前が再配列されて代替解析が考慮に入れられるために必要な最小信頼値です。

タスクの結果

`parse()` メソッドは、入力ストリングを音訳して解析し、入力ストリングごとに以下のデータ構造体のコレクションを戻します。`parse()` メソッドは、名前解析に関する以下のオブジェクトのそれぞれを戻します。

ParseAlternate

代替 (等位) の名前の解析データが含まれます。

ParseName

可能な名前解析に関する情報が含まれます。

ParseField

名前フィールドに関する解析情報が含まれます。

ParsePhrase

単一の名前句の解析データが含まれます。

NameParser を使用した名前の解析

NameParser は、個人名を、名、姓、称号、および修飾子などの構成要素の部分に分離する、低レベルのコンポーネントです。

NameParser は個人名を解析するとき、次の各アクションを実行します。

- 名と姓との間の適切な境界を判別します。
- 名前から称号および修飾子を分離します。
- 名および姓を個別に操作可能な (プレフィックスやステムなどの詳細を扱う) 句に分割します。
- 単一の名前ストリングから複数の名前を分離します。例えば、ストリング『John and Mary Smith』は、『John Smith』 および 『Mary Smith』 というストリングに分割されます。

入カストリングのタイプ

parseName() 関数には、アプリケーションが名前を処理するために現在使用している方式との統合を容易にする、いくつかの多重定義バージョンがあります。名前が既に名フィールドと姓フィールドに分割されている場合は、NameParser を使用して、各フィールドに適切な項目が確実に配置されたことを確認できます。

parseName() 関数を使用することにより、次の方式で名前を NameParser に渡すことができます。

単一ストリングで、解析されていない名前として

解析される名前全体は、元の形式で単一ストリングとして受け渡すことができます。呼び出し元のアプリケーションが、入力ストリングで予期されるフィールドの順序を指定するために使用できる、gnFirst フラグが別個にあります。gnFirst が true の場合、名前では名が最初になると予期されます。つまり、NameParser はストリング内で名が姓に先行すると想定します。gnFirst が false の場合、NameParser は入力ストリングで姓が名に先行すると想定します。中国語、日本語、および韓国語と判別された名前では、デフォルトで、NameParser はストリング内で姓が先行すると想定しますが、あいまいなケースでは gnFirst フラグがこのアルゴリズムにどちらかの方向にバイアスをかけます。gnFirst フラグは、デフォルトで true に設定されます。

注: gnFirst フラグは、処理しているストリングのフィールドの順序が分かっている状態のときに使用します。順序は分からないものの、混合されている可能性があることが分かっている場合は、gnFirst を true の設定のままにしておきます。

単一文字列で、解析された名前として

名前が以前に名と姓に解析されている場合、姓が最初で姓と名の間をコンマで区切った単一文字列 (例: 「King, Martin Luther」) として名前を渡すことができます。NameParser は、名前を自然順に変換し、再解析します。修飾子を区切るコンマ (例: Martin Luther King, Jr.) が使用可能であり、姓と名の間で区切りに間違われることはありません。

2 つの文字列で、解析された名前として

名前が既に個別の名および姓の文字列に分割されているコンテキストでは、これらの 2 つの文字列 (例: 「Martin Luther」 および 「King」) を NameParser に渡すことができます。NameParser は、これらを自動的に自然順の単一文字列に連結し、再解析します。

3 つの文字列で、解析された名前として

名前が既に個別のファーストネーム、ミドルネーム、およびラストネームの文字列に分割されているコンテキストでは、これらの 3 つの文字列 (例: 「Martin」、「Luther」、および 「King」) を NameParser に渡すことができます。NameParser は、これらを自動的に自然順の単一文字列に連結し、再解析します。

名前の解析のための NameParser 関数

NameParser を使用した名前の分析のためのメソッドは、parseName() 関数です。

parseName() 関数

氏名の分析を行い、解析ツリーを生成します。解析ツリーは、名前の要素を構造化されたユニットにまとめた階層です。個別のトークン (スペースまたは句読点で区切られた文字列) で始まり、トークンを結合すると名前の句になる場合があります。名前句を結合すると 1 つ以上の完全な個人名になります。

代替解析および妥当性検査のスコア

NameParser は、指定の gn-first パラメーターと名前が提供された順序を照合する解析から始めて、名前に対する複数の解析を検索します。NameParser は、最初の解析よりも可能性の高い代替解析についても検討します。

称号と修飾子がすべて削除された後で、名前の残りの句の順番を循環させることができますが、相対順序は保持する必要があります。例えば、元の入力名が句 A B C D を含んでいる場合、再解析によって句を B C D A または D A B C に変換することはできますが、A D B C または B A C D には変換できません。2 つ以上の循環が可能な場合、実際の再解析の結果は、最も高い妥当性検査スコアを出す結果になります。

parseName() によって返される ParseData オブジェクトは、1 つ以上の結合名を含んでいて、各結合名の代替解析 (存在する場合) をリストします。1 つの名前に複数の解析が存在することがあります。

NameParser の句オーバーライド・リスト

句を句オーバーライド・リストに追加して、NameParser の動作をカスタマイズできます。句オーバーライド・リストは、NameParser が検索を行う新しい句の補足

リストです。補足リストに句がない場合、NameParser は数百万の名前句についての情報を含む独自の内部表を検索します。

デプロイメントごとに要件が異なるため、NameParser のトークンと句のデータベースを変更する必要がある場合があります。例えば、NameParser 内に構築されたセットへ称号および修飾子を追加すること、データでの発生頻度が高い名前の語幹トークンまたは接辞トークンによって NameParser の内部リソース表を補足すること、または名と姓の頻度データのオーバーライドによって、処理中の名前のローカルの使用法により十分に準拠することが必要な場合があります。

句オーバーライド・リストは、std::string オブジェクトから PhraseOverrideData オブジェクトにマップする STL マップです。std::string オブジェクトは、大文字の ASCII 文字で表された、リストに追加する名前トークンまたは句です。PhraseOverrideData オブジェクトは、トークン・タイプに加えて、名および姓の係数を提供するトークンに関する情報の記録です。

名と姓の係数は NAME_STEM トークン以外のすべてのトークンで無視されるので、他のすべてのトークン・タイプではこれらの値を 0 に設定する必要があります。NAME_STEM トークンでは、名および姓の係数は、NameParser が名前でトークンを検出した場合に、名フィールドまたは姓フィールドにトークンを関連付ける必要があるかどうかを判別する際に使用されます。一般に、フィールドの関連付けの判別は、トークンで名の係数が大きいか、または姓の係数が大きいかに基づいて決定が行われます。全体的なロジックではより多くの要素が検討されますが、名と姓の係数は、名前トークンを関連付ける方法を決定するための主な測定基準となります。

実際の句オーバーライド・リストは、直接にはアクセスできません。このリストと連動する API が、名前および PhraseOverrideData レコードを、クライアント・コードによって提供されたリストにコピーしたりそこからコピーしたりすることによって操作します。句オーバーライド・リストは、デフォルトでは空です。

句オーバーライド・リスト内の各エントリは PhraseOverrideData オブジェクトに関連付けられ、これがそのトークンのタイプ、姓の係数、および名の係数を指定します。PhraseOverrideData オブジェクトのタイプ・メンバーは、関連データの処理方法を setPhraseOverrides() に知らせます。PhraseOverrideData クラスには、以下の文字のサポートが含まれます。

- 小文字のローマ字は、大文字のローマ字に等しくなります。
- 拡張ローマ字は、ASCII の最も近い文字に等しくなります。例えば、発音区別符号は、ほとんどのアルファベット文字では削除されます。次の Unicode エンコードのすべてのアルファベット文字が許可されます。
 - Latin-1
 - ラテン文字拡張 A
 - ラテン文字拡張追加ブロック
 - ラテン文字拡張 B ブロック (ほとんどの文字)
- 数字 0-9、スペース、および次の特殊文字に意味があります。@ \$ % & + = /
- ハイフン、ピリオド、コンマ、タブ、および改行は、スペースに等しくなり、1 行の中の連続する複数のスペースに相当する文字は、シングル・スペースに等しくなります。

アポストロフィを含む前述のリストに含まれていないその他の文字は、すべて無視されます。

第 4 章 名前の分析

名前を分析するときには、名前についてのさまざまな属性を識別します。例えば、名前に関して予想される性、名前に関して予想される民族 (国/地域)、名前の由来地として予想される国、可能な名前の変形、名前のカテゴリ (個人名か組織名か) などです。

NameWorks を使用した名前の分析

名前の完全分析を行うときは、`analyze()` メソッドを使用します。このメソッドは、名前を音訳して解析してから、性別情報、国/地域別情報種別、名前の変形名前形式、名前が存在する国のリスト (関連国情報) を提供します。

このタスクについて

名前の完全分析を行うには、NameWorks の Analytics クラスの `analyze()` メソッドを使用し、以下の値をメソッドに渡します。

- フルネームを単一のストリング値として渡すか、または特定の解析が必要な場合には、姓と名を別個のストリングとして渡します。
- `alternateThreshold` 値を表す 0 から 100 までの整数。これにより、NameWorks が代替解析を検索すべきかどうかを決定する許容される最小の信頼値が設定されます。最初の解析の信頼値がこのしきい値よりも大きい場合、代替解析は試行されません。`alternateThreshold` 値がゼロ (0) の場合、代替解析は常に抑止されません。
- `maxForms` 値を表す整数。これは、各名前句に対して生成される変形形式の数を制限します。値がゼロ (0) の場合、可能なすべての変形形式が返されます。
- `maxElements` 値を表す整数。これは、各名前句に対して返される国要素の数を制限します。値が 1 より小さい場合、可能なすべての国要素が返されます。

タスクの結果

`analyze()` メソッドは、入力ストリングを音訳して解析した後、分析データを表す以下のセットのネストされたデータ構造体を返します。

- `AnalysisData`
- `AnalysisAlternate`
- `AnalysisName`
- `AnalysisField`
- `AnalysisPhrase`

NameWorks を使用した名前の国/地域別情報の識別

NameWorks には、NameClassifier—Country of Association (NC_COA) API コンポーネントを使用して、個人名の国/地域別情報を識別する機能があります。これらの機能は、まず単純な国/地域別情報による分類を実行した後、1 つの国/地域情報を選択できない場合には、より複雑な関連国の分析を実施して可能性のある国/地域情報を判別します。組織名の場合、国/地域別情報は、名前がサポート対象の非ラテン文字で書かれているならその文字タイプから自動的に推測されたり、ユーザーによって提供されたりすることがあります。ラテン文字の組織名の場合、国/地域別情報はユーザーによって指定されていなければなりません。そうでない場合、国/地域別情報は未確定として推定されます。

国/地域別情報の識別

世界中の名前は場所によって、また国/地域グループによってさまざまに異なります。このような差異は、名前で使われる音の違いから、フルネームにおける名と他の部分との位置関係に至るまで、広範囲に及びます。名前の国/地域別情報を識別することで、名前の突き合わせ機能を大きく改善させることができます。

IBM InfoSphere Global Name Management 製品では、名前の国/地域を識別することにより、文化圏固有の知識を適用できます。これにはニックネームの認識や、検索リコールを増やして誤検出を減らすための事前調整されたパラメーター設定などがあります。IBM InfoSphere Global Name Management 製品は、ラテン・アルファベットによって表される個人名の文化的性質を推測して識別する目的で、言語的、統計的、および確率的な技法を組み合わせることで適用します。

国/地域を識別できれば名前の突き合わせ機能を強化することができますが、文化圏固有の知識がなくても、IBM InfoSphere Global Name Management 製品には名前の解析、性別特定、および突き合わせを効果的に行う優れた能力があります。

この例は、スペイン語系の名前の国/地域別情報を識別することにより、名前の突き合わせ機能がどのように強化されるかを示しています。

スペイン語系のコミュニティでは、通常、人物名に 2 つの姓が含まれます。最初の (左側の) 姓は父親の姓であり、その人自身のファミリー・ネーム (苗字) としてその名前が使われます。最後の姓は母親の姓であり、これは省略されることがあります。

IBM InfoSphere Global Name Management 製品は、「Ana Garcia Valdez」という名前をヒスパニックの名前として識別し、処理中に文化圏固有のスコアリング・パラメーターを適用できるため、「Ana Garcia」という名前は最上位の名前マッチングになります。しかし「Ana Valdez」という名前は最上位の名前マッチングとは見なされません。これら 2 つの名前は同じ名前構成要素を含んでいますが、最初の姓の構成要素にマッチしないためです。

国/地域別情報コード

国/地域別情報コードは、国/地域の分類中に個人名に関連付けられる、1 つ以上の国/地域別情報を記述します。

IBM InfoSphere Global Name Management 製品では、以下の国/地域別情報コードが使用されます。

注: ロールアップ国/地域別情報コードは、特定の地域に含まれる、定義済みの一連の国/地域別情報を示します。ある名前に対して、その地域に含まれる複数の国/地域別情報コードが返される場合、名前の国/地域別情報をより適切に示したものがロールアップ・コードです。

表 5. 国/地域別情報コードおよびそれらに関連付けられた国/地域

コード	関連付けられた国/地域別環境
0	未確定 (有効なロールアップ・コードを判別できない)
1	英語
2	アラビア語
3	中国語
4	ヒスパニック
5	韓国語
6	ロシア語
7	フランス語
8	ドイツ語
9	タイ語
10	インドネシア語
11	ヨルバ語
12	ペルシア語
13	パキスタン語
14	インド語
15	日本語
16	アフガニスタン語
17	ベトナム語
18	ポーランド語
19	ポルトガル語
20	トルコ語
38	南西アジア (アラビア語、ペルシア語、パキスタン語、アフガニスタン語、およびトルコ語の国/地域を組み合わせたものを表すロールアップ国/地域別情報コード)
39	ヨーロッパ (英語、フランス語、ドイツ語、ヒスパニック、およびポルトガル語の国/地域を組み合わせたものを表すロールアップ国/地域別情報コード)
40	ハン (中国語、韓国語、およびベトナム語の国/地域を組み合わせたものを表すロールアップ国/地域別情報コード)

さらに、カスタマイズされた国/地域別情報のコードを以下のように追加できます。

表 6. カスタムの国/地域別情報コード

コード	関連付けられた国/地域別環境
41	CUSTOM_01
42	CUSTOM_02

表 6. カスタムの国/地域別情報コード (続き)

コード	関連付けられた国/地域別環境
43	CUSTOM_03
44	CUSTOM_04
45	CUSTOM_05
46	CUSTOM_06
47	CUSTOM_07
48	CUSTOM_08
49	CUSTOM_09
50	CUSTOM_10
51	CUSTOM_11
52	CUSTOM_12
53	CUSTOM_13
54	CUSTOM_14
55	CUSTOM_15
56	CUSTOM_16
57	CUSTOM_17
58	CUSTOM_18
59	CUSTOM_19
60	CUSTOM_20

氏名の国/地域別情報の識別

名前の国/地域別情報を識別し、その国/地域別情報種別の注釈をデータに付けることができれば、名前の突き合わせをより効果的に、より高い信頼性で行えるようになり、名前の突き合わせにおけるパフォーマンスも大幅に高めることができます。このタスクは、まだ別個の名前句に解析されていない名前ストリングの、国/地域別情報を識別するときに役立ちます。

このタスクについて

名前の国/地域別情報を識別するには、IBM NameWorks の Analytics クラスの `classify()` メソッドを使用し、氏名をストリング値でメソッドに渡します。

このメソッドは、名前に対して音訳を行い、名前を解析し、解析した名前に国を関連付けた後、国/地域別情報種別を、標準の 20 の国/地域別情報コードの 1 つとして戻します。

タスクの結果

`classify()` メソッドは、CultureData オブジェクトを戻します。

名前フィールドの国/地域別情報の識別

名前の個々の部分 (名と姓) の国/地域別情報を識別し、その国/地域別情報種別の注釈をデータに付けることができれば、名前の突き合わせをより効果的に (より高い信頼性で) 行えるようになり、名前の突き合わせにおけるパフォーマンスも大幅に高めることができます。

このタスクについて

既に名フィールドと姓フィールドに解析されている名前の国/地域別情報を識別するには、IBM NameWorks の Analytics クラスの `classify()` メソッドを使用し、名前フィールド (gn と sn) をストリング値のペアでメソッドに渡します。どちらかの名前フィールドを NULL または空ストリングにすることができます。

このメソッドは、名前を解析し、解析した名前フィールドに国/地域別情報を関連付けた後、一方または両方の名前フィールドの国/地域別情報種別を、標準の 20 の国/地域別情報コードの 1 つとして戻します。

タスクの結果

`classify()` メソッドは、CultureData オブジェクトを戻します。

組織名の国/地域別情報の識別

組織名の国/地域別情報を識別できる場合には、数字や特殊記号をその言語の文字で書き表した形式にマッピングするための言語固有の規則を適用できます。例えば、数字ストリングの「62」は英語では「sixty-two」に相当しますが、ハンゲル名では「**육십이**」と一致します。

このタスクについて

組織名の国/地域別情報に固有のサポートは、英語、北京語、ヒスパニック、インド語、日本語、韓国語、ポーランド語、ポルトガル語、ロシア語、およびトルコ語で使用可能です。

一部の文字に関しては、その文字で書かれた名前に使用されている筆記体系に基づいて、組織名の国/地域別情報が自動的に判別されます。その他のすべての組織名では、存在する国/地域固有のリソースを使用できるように、ユーザーが国/地域別情報を提供する必要があります。次の表は、文字が国/地域別情報にマップされる方法を示しています。

漢字と仮名の混用	日本語
仮名	日本語
漢字	中国語
デーバナーガリー文字	インド語
キリル文字	ロシア語

他のすべての組織名は、特定の国/地域別情報が提供されない限り、未確定として分類されます。

漢字文字でエンコードされた名前はすべて中国語であると推定されることに注意してください。漢字の日本語音訳を起動するためには、日本語の漢字を使用しているすべての組織名に対して、日本の国/地域別情報を指定する必要があります。

日本語の漢字の筆記体系は、中国語に使用される漢字の体系とほぼ同じ文字セットが使用されます。そのため、多くの場合、漢字文字の組織名が日本語か中国語かを判別できません。それら 2 つの言語の筆記形態は類似していますが、文字の発音は完全に異なっています。漢字文字に対して日本語の読みが取得されるようにするには、日本語の漢字名の国/地域別情報を入力で指定する必要があります。

仮名で書かれた日本語の組織名は、文字タイプに基づいて、日本語として自動的に検出されます。ただし、日本のほとんどの組織名は、漢字のみか、漢字と仮名の混用で記述されます。

NameWorks を使用した名前の性別の識別

NameWorks には、名 (ギブンネーム) に関連した性別の相対頻度を識別する機能が含まれています。名前の性別頻度は、名前データ・アーカイブ内にその名前が含まれるすべての国/地域での頻度の合計として取り出すことも、特定の国/地域での頻度だけを表示するようにすることもできます。

頻度は、指定の名前の出現総数のパーセンテージとして表されます。例えば、名前の「ILHAM」は女性が 79% で男性が 20% という結果を返します。これは、名前データ・アーカイブで、「ILHAM」のエントリーの 79% が女性として報告され、そのエントリーの 20% が男性として報告され、1% には特定の性別の報告がないことを意味します。

氏名の性別の識別

このタスクは、氏名の名フィールドを識別して名に関する性別データを戻したい場合に使用します。特定の国/地域の性別頻度は、その国/地域を指定することで取り出すことができます。例えば、トルコ語地域を指定することによって、名前の「ILHAM SAYIL」に対するトルコでの名の性別分散を検索できます。国/地域の値を指定しないと、すべての国/地域での性別分散が返されます。

このタスクについて

まだ名前フィールドに解析されていない名前の性別を識別するには、IBM NameWorks の Analytics クラスの `genderize()` メソッドを使用し、氏名をストリング値でメソッドに渡します。

このメソッドは名前を解析し、解析した名に性別を関連付けた後、性別データを戻します。

タスクの結果

`genderize()` メソッドは、GenderData オブジェクトを戻します。戻り値はデータ構造体であり、名の出現相対頻度の値と、名に関連付けられた性別が男性、女性、または不明である可能性を表すパーセンテージの値が含まれます。

名の性別の識別

このタスクは、名に関する性別データを戻したい場合に使用します。特定の国/地域の性別頻度は、その国/地域を指定することで取り出すことができます。例えば、トルコ語地域を指定することによって、名前の「ILHAM」に対するトルコでの名の性別分散を検索できます。国/地域の値を指定しないと、すべての国/地域での性別分散が返されます。

このタスクについて

名の性別を識別するには、NameWorks の Analytics クラスの `genderizeField()` メソッドを使用し、名をストリング値 (`givenName`) でメソッドに渡します。

このメソッドは、名に性別を関連付けた後、性別データを戻します。

タスクの結果

`genderizeField()` メソッドは、GenderData オブジェクトを戻します。戻り値はデータ構造体であり、名の出現相対頻度の値と、名に関連付けられた性別が男性、女性、または不明である可能性を表すパーセンテージの値が含まれます。

NameWorks を使用した名前の関連国の識別

NameWorks に含まれる機能を使用すると、名前の分析に役立つ名前の関連国を識別できます。

関連国 (Country of Association)

国を名前に関連付けることができれば、IBM InfoSphere Global Name Management の国/地域別情報の識別機能および名前突き合わせ機能をさらに拡張できます。名前を国に関連付けることで、「不明な」国/地域別情報の数を減らし、国/地域別情報の識別精度を高めることができます。

IBM InfoSphere Global Name Management 製品には、国/地域別情報の識別精度を高めるための、関連国 (COA) 統計が含まれています。これを使用することにより、開発者は IBM InfoSphere Global 名前データ・アーカイブ (NDA) の分布データを使って国/地域別情報のバランスを調整し、検証することができます。COA 機能は、名前を国に関連付ける操作に関連した以下の 3 つの異なる値を提供できます。

頻度 ある国の中で特定の名前がその国の他の名前と比較してどれほど一般的であるかを示します。頻度値は、名前によって異なります。

信頼性

NDO に格納されている特定の国のデータ量が、NDO に格納されている他の国のデータ量と比較してどの程度であるかを示します。信頼性は、国ごとに決まった値が設定されます。

重要度

多国間重要度とも呼ばれます。特定の名前がある国でどれだけ代表的であるかと他の国でどれだけ代表的であるかを比較して示す値が、この機能によって提供されます。重要度は、関連国を決定するための方式の中では完成度が

高い方式といえます。なぜなら、NDO に格納されている特定の国に関するデータ量に加え、その国で普及している名前の総数などの要素が考慮されるからです。

COA 機能は、以下の情報をこの順序で返します。

1. 重要度 (高い順)
2. 頻度 (高い順)
3. 信頼性 (高い順)
4. ISO 3166 国別コード (英字表記の昇順)

氏名の関連国の識別

名前の関連国を識別することによって、名前分析にもう 1 つの情報層を追加できます。このタスクは、名前が既に名前フィールドに解析されている場合に使用します。

このタスクについて

名と姓の関連国を識別するには、IBM NameWorks の Analytics クラスの `associate()` メソッドを使用し、以下の値をメソッドに渡します。

- 氏名 (ストリング値)。
- *maxElements* 値を表す整数。この値は、名前句ごとに戻される国要素の数を制限します (1 より小さい値を渡した場合は、すべての国要素が戻されます)。

このメソッドは氏名を解析し、名前フィールドそれぞれに対して音訳を行い、解析による名前フィールドに、信頼性の最も高い国を関連付けます。

タスクの結果

`associate()` メソッドは、名前句ごとに以下のネストされたオブジェクトのセットを返します。

- CountryData
- CountryPhrase
- CountryElement

名と姓の関連国の識別

名と姓の関連国を識別できれば、名前分析にもう 1 つの情報層を追加することができます。このタスクは、名前が既に名前フィールドに解析されている場合に使用します。

このタスクについて

名と姓の関連国を識別するには、IBM NameWorks の Analytics クラスの `associate()` メソッドを使用し、以下の値をメソッドに渡します。

- 名前フィールド (gn と sn)。ストリング値のペアで渡します (どちらかの名前フィールドを NULL または空ストリングにすることができます)。
- *maxElements* 値を表す整数。この値は、名前句ごとに戻される国要素の数を制限します (1 より小さい値を渡した場合は、すべての国要素が戻されます)。

このメソッドは、名前フィールドそれぞれに対して音訳を行い、名前フィールドに国を関連付けます。

タスクの結果

associate() メソッドは、名前句ごとに以下のネストされたオブジェクトのセットを返します。

- CountryData
- CountryPhrase
- CountryElement

NameWorks を使用した名前変形の生成

NameWorks に含まれる機能を使用すると、名前の名前フィールドに関する変形のリストを生成できます。

名前の変形

名前の変形 とは、ある名前の代わりの名前で、その名前と同等であると見なされてはいても、特定の外形が異なるものをいいます。言い換えると、これら 2 つの名前はある意味で同等であると見なされ、コンテキストによっては互いに置き換えることができます。

名前の変形は、以下のような多くの理由で出現します。

- スペルの変形 (例えば、*Geoff* と *Jeff*)
- ニックネーム (例えば、*William* の代わりに *Bill*)
- 省略形 (例えば、*Guadalupe* の代わりに *GPE*)
- 同語源語、つまり翻訳語 (例えば、*Pierre* の代わりに *Peter*)
- 国/地域別の相違
- 構成要素の順序の変形 (例えばある名前を別の国/地域別環境に適合させる場合。*Gandikota Lakshmi Narayana Sastry* を変形した *L.N.S. Gandikota* など)
- ある書記体系から別の書記体系への音訳 (例えば、表語文字の漢字からローマ字へ)

IBM InfoSphere Global Name Management 製品は、スペルの変形、ニックネーム、国/地域別相違、および省略形に起因する名前の変形を生成できます。

名前にどのような変形があるかが分かれば、名前の照会に変形形式を含めるように拡張したり、検索で一致候補のリストを生成するときに名前のそれらの変形を含めたりする助けになります。名前の変形が分かれば、名前の分析をするときにも役立つ場合があります。分析者は、出現する可能性がある名前変形形式のリストを参照できるからです。

IBM InfoSphere Global Name Management 製品では、名前の変形リストの生成プロセスには、名前フィールド (名および姓の各フィールド) を名前句に分割し、その後それぞれの名前句ごとに変形のリストを生成することが含まれます。どの名前が名でどの名前が姓であるかを識別することは重要です。句がどのフィールドに出現するかに応じて、異なる変形が存在するためです。

名前の変形は国/地域間で大きく異なるため、名前の各フィールドの背後にある国/地域別情報を識別することも重要です。ある特定の名前が多く国/地域別環境で見られ、スペルも同じであったとしても、それらの名前が同じであるとは見なされません。実際、それらは異なる名前であり、その名前が関連付けられている国/地域別環境に応じて異なる変形形式が生成されます。

例えば、ヒスパニックの名前 *Juan* と中国語の名前 *Juan* の変形を比較すると、それぞれの国/地域別環境においてこの名前は同じ名前ではないため、名前の変形形式にはかなりの相違があることが分かります。

ヒスパニックの名前 *Juan* の変形には、以下が含まれます。

- Juam
- Juanch
- Juancho
- Xuan

中国語の名前 *Juan* の変形には、以下が含まれます。

- Chuan
- Chwan
- Jwan
- Zhuan

両方の国/地域別環境に共通した同じ変形が一部あるものの、これら 2 つの名前はかなり異なっています。この相違は、名前の変形の順序の相違によってはっきり示されています。これら 2 つの国/地域別環境の間ではスペルの出現頻度に相違があり、それがこの順序に反映されています。

氏名の名前変形のリストの生成

名前変形のリストを生成することにより、名前照会を拡張できるほか、より適切な一致候補リストを生成できます。また、名前に関連付けることができる変形を調べられるので、名前の分析を改善できます。名前変形リストの生成プロセスは、名前を名前フィールドに分解することと、名前フィールドそれぞれの国/地域別情報を特定することに依存しているため、このタスクは、名前がまだ名フィールドと姓フィールドに解析されていないときに使用します。

始める前に

名前の変形のリストは、名前に関連付けられた国/地域別情報によって大きく異なるため、`getVariants()` メソッドは国/地域別情報コードをパラメーターとして取りまます。このタスクを実行する前に、名と姓の国/地域別情報コードを取得するとよい場合があります。(Analytics クラスの `analyze()` メソッドか `classify()` メソッドのどちらかを使用することによって、国/地域別情報コードを取得できます。)

このタスクについて

氏名の名前変形のリストを生成するには、IBM NameWorks の Analytics クラスの `getVariants()` メソッドを使用し、以下の値をメソッドに渡します。

- 氏名 (ストリング値)。

- 名前の国/地域別情報を示す国/地域別情報コード。名前の国/地域別情報が不明の場合は、NULL (Java の場合) または -1 (Web サービスの場合) をメソッドに渡すと、IBM NameWorks が名前に対して可能性の高い国/地域別情報を判別します。
- 名前句ごとに戻される変形形式の数を制限するための整数。(負の値またはゼロ値を渡した場合は、すべての変形名前形式が戻されます。)

このメソッドは、変形のリストを戻します。IBM NameWorks に国/地域別情報を判別させる -1 または NULL を国/地域別情報パラメーターとして渡した場合は、VariantData オブジェクトの国/地域別情報フィールドを検査することにより、IBM NameWorks が名前に関連付けた国/地域別情報を検出できます。

タスクの結果

getVariants() メソッドは、ツリー構造内にネストされたオブジェクトの以下のセットを戻します。このツリー構造は、分析対象の名前で検出された名前句の分解を表します。

- VariantData
- VariantPhrase
- VariantForm

名と姓の名前変形のリストの生成

名前変形のリストを生成することにより、名前照会を拡張できるほか、より適切な一致候補リストを生成できます。また、名前に関連付けることができる変形を調べられるので、名前の分析を改善できます。名前変形リストの生成プロセスは、名前を名前フィールドに分解することと、名前フィールドそれぞれの国/地域別情報を特定することに依存しているため、このタスクは、既に名フィールドと姓フィールドが存在するときに使用します。

始める前に

名前の変形のリストは、名前に関連付けられた国/地域別情報によって大きく異なるため、getVariants() メソッドは国/地域別情報コードをパラメーターとして取りまます。このタスクを実行する前に、名と姓の国/地域別情報コードを取得するとよい場合があります。(IBM NameWorks の Analytics クラスの analyze() メソッドか classify() メソッドのどちらかを使用することによって、国/地域別情報コードを取得できます。)

このタスクについて

既に名フィールドと姓フィールドに解析されている名前の名前変形のリストを生成するには、IBM NameWorks の Analytics クラスの getVariants() メソッドを使用し、以下の値をメソッドに渡します。

- 名フィールドと姓フィールド (どちらかの名前フィールドを NULL または空ストリングにすることができます)。

- 名前の国/地域別情報を示す国/地域別情報コード。名前の国/地域別情報が不明の場合は、NULL (Java の場合) または -1 (Web サービスの場合) をメソッドに渡すと、IBM NameWorks が名前に対して可能性の高い国/地域別情報を判別します。
- 名前句ごとに戻される変形形式の数を制限するための整数。(負の値またはゼロ値を渡した場合は、すべての変形名前形式が戻されます。)

このメソッドは、変形のリストを戻します。IBM NameWorks に国/地域別情報を判別させる -1 または NULL を国/地域別情報パラメーターとして渡した場合は、VariantData オブジェクトの国/地域別情報フィールドを検査することにより、IBM NameWorks が名前に関連付けた国/地域別情報を検出できます。

タスクの結果

getVariants() メソッドは、ツリー構造内にネストされたオブジェクトの以下のセットを戻します。このツリー構造は、分析対象の名前で検出された名前句の分解を表します。

- VariantData
- VariantPhrase
- VariantForm

コンポーネント **API** を使った名前の分析

名前を分析するときには、名前についてのさまざまな属性を識別します。例えば、名前に関して予想される性、予想される文化圏、名前の由来地として予想される国、名前のさまざまな変形、個人名または組織名としての名前の分類などです。

IBM InfoSphere Global Name Management 製品で提供される C++ ライブラリーをリンクさせると、テクノロジーをアプリケーションまたはワークフローに直接統合できます。API と共に、ライブラリーの使用方法を示すサンプル・アプリケーションもあります。

第 5 章 名前の検索

IBM InfoSphere Global Name Recognition 製品を使用して、複数のデータ・リスト間で名前を検索できます。

IBM NameWorks におけるデータ・リストの管理

データ・リストは、メモリー・ベースの名前コレクションで、そのデータは、IBM InfoSphere Global Name Management アプリケーションの初期設定時に外部データ・ソース (フラット・ファイルなど) から追加されます。データ・リストの各項目には、検索処理中にアクセスおよび考慮される 1 つの名前に関する広範な情報が含まれます。名前検索の際にこれらの情報を使用して、いくつかの言語指標、国/地域別指標、およびストリング類似性指標がきめ細かく適用されます。

データ・リストは、IBM が名前の自動化検索および突き合わせで使用する主要なデータ構造です。

通常、システム管理者は、リストごとに一連の構成パラメーターを作成および保守することにより、データ・リストを構成、データ設定、および管理します。IBM NameWorks および NameHunter は、不定数の構成可能名前リストをサポートできるため、単一の検索要求で処理される名前が特定数に制限されることはありません。さらに、メモリーに常駐する各種名前リストにさまざまな外部ファイルをマップする機能があるため、各検索要求において、トランザクションごとに関連するリストだけを考慮する動的検索スコープ設定が可能になります。

IBM NameWorks の構成ファイルには、各データ・リストとその検索エンジン・インスタンスの間のマッピング情報に加え、システム初期設定の際に使用されるその他の主要な情報が含まれています。

IBM NameWorks の構成ファイルには、検索エンジン・インスタンスとデータ・リスト間のマッピングを指定するデータ・リスト情報と、データ・リストに対して実行できる検索のタイプ (全検索または固有の名前検索) を示すフラグが必要です。この検索エンジン・インスタンスでデータ・リストに名前を追加できる場合、データ・リスト・セクションには追加フラグが含まれています。

アクティブ検索でデータ・リストを参照している最中であっても、システム管理者およびクライアント・アプリケーションは、そのデータ・リストに対して名前を追加、削除、または更新できます。アクティブ検索では、その操作を完了するためにキャッシュ情報が使用され、後続の検索では変更されたデータ・リスト名情報が使用されます。

データ・リスト

データ・リストはメモリー常駐型のデータ構造で、(フラット・ファイルなどの) 外部ソースから得られる名前のセットがこれに格納されます。データ・リストが作成されて、そこに名前が格納されたなら、それ以降の検索要求でそのデータ・リスト

が使用可能になります。各データ・リストには固有の名前があり、特定の形式に従う必要があります。データ・リストの中には 1 個から何億個もの名前を含めることができます。

データ・リストは動的データ構造であるため、IBM NameWorks はデータ操作トランザクションをサポートします (データ・リストの内容の追加、更新、削除など)。動的操作により、メモリーに保管されるデータ・リストの内容は、(その内容が表す) 基礎となるデータ・ソースが変化する場合でも、そのデータ・ソースと常に同期化されます。

データ・リストは IBM InfoSphere Global Name Management 製品の検索エンジンによって使用され、検索要求は 1 つ以上のデータ・リストに対して実行されます。それぞれの検索要求では、使用するデータ・リストの名前を示す必要があります。

データ・リストは通常、システム管理者によって管理されます。組織のニーズを満たすために、検索エンジン・サーバーの構成に応じて 1 つ以上のサーバーに配置することができます。

通常、システム管理者は 1 つのデータ・リストを単一の検索エンジン・インスタンスに関連付けます。IBM NameWorks ではこれは Distributed Search プロセスのインスタンスです (1 つの通信管理プロセスと 1 つ以上の Searcher プロセス)。

システム管理者またはクライアント・アプリケーションは、必要に応じてデータ・リスト上の名前を追加、更新、または削除できます。ただし、Distributed Search プロセスの各インスタンスにおいて、セッション中に追加される名前の受信側としてただ 1 つのデータ・リスト (追加リスト) だけを指定できます。Distributed Search のインスタンスと共に構成される他のすべてのデータ・リストには、関連する構成ファイルに従ってセッション初期化中に格納される名前だけを含めることができます。このため、初期化の後に追加されるすべての名前は、追加リストに格納されます。

データ・リストへの名前の追加

システム管理者は、データ・リスト管理の一環で、データ・リストの定期的リフレッシュの合間の暫定更新として、データ・リストに名前を追加しなければならない場合もあります。データ・リストはメモリー常駐なので、アクティブ検索がデータ・リストにアクセス中であっても、いつでもデータ・リストに名前を追加できます。名前が追加されるときにアクティブ検索が進行中だった場合は、データ・リストに対する後続の検索時に新規名にアクセスできます。

始める前に

- IBM NameWorks 構成ファイルで、追加フラグの立ったデータ・リストが構成されていなければなりません。そうでない場合は、データ・リストに名前を追加しようとするとエラー・メッセージが表示されます。
- データ・リストの名前が分かっている必要があります。(既存のすべてのデータ・リスト名のリストを戻すには、`getDatalistNames()` メソッドを使用します。)

- 名前は、名フィールドと姓フィールドに既に解析されていなければなりません。`analyzeForSearch()` メソッドを使用して、名前をフィールドに解析し、分類できます。これにより、データ・リストに名前を追加するのに必要な情報を得ることができます。
- 名前フィールド (名と姓) それぞれの国/地域別情報が既に存在する場合は、国/地域別情報コードも名前に追加できます。

このタスクについて

既存のデータ・リストに名前を追加するには、IBM NameWorks の Scoring クラスの `addNameToDatalist()` メソッドを使用し、以下の値をメソッドに渡します。

- データ・リストの名前。
- データ・リストに追加する名フィールド値と姓フィールド値。
- オリジナル名前解析、または名前のオリジナル・スクリプト (ラテン・アルファベットで表されない場合)、あるいはその両方。両方を渡す場合、その追加にはオリジナルの名前またはスクリプトの代替解析のフラグを立てる必要があります。
- 新しい名前レコードに関連付ける補足データ。(この情報は、生年月日や運転免許証番号など、名前に関連した追加または補足データを識別するためのキー値となります。これにより通常は、一致した名前に関する、より完全な情報を取得できるようになります。補足データは、検索後フィルタリングや加重で使用できます。データ・リスト上の名前を更新または削除する際にも使用できます。同じ補足データを共有するすべての名前レコードが、更新または削除されます。)
- 名と姓の国/地域別情報コード。(このような国/地域別情報コードがない場合は、値 `-1` を渡します。この値は、名フィールドと姓フィールドの国/地域別情報を最初に分類するようメソッドに通知する意味を持ちます。)

タスクの結果

`addNameToDatalist()` メソッドは、示されたデータ・リストに、名前とその関連情報を追加します。国/地域別情報コードとして値 `-1` を渡すと、IBM NameWorks は名前に最適な国/地域別情報コードを自動的に決定します。

データ・リスト上の名前の更新

データ・リストの定期的リフレッシュの合間に、システム管理者は、データ・リスト管理職務の一部として、既存のデータ・リスト上の名前を更新しなければならない場合があります。データ・リストはメモリー常駐なので、アクティブ検索がデータ・リストにアクセス中であっても、いつでもデータ・リスト上の名前を更新できます。名前が更新されるときにアクティブ検索が進行中だった場合は、データ・リストに対する後続の検索時に新しい情報にアクセスできます。

始める前に

- データ・リストの名前が分かっている必要があります。(既存のすべてのデータ・リスト名のリストを戻すには、`getDatalistNames()` メソッドを使用します。)
- 名前に関連付けられたオリジナル補足データ値 (*originalData* 値) が分かっている必要があります。このデータ値は多くの場合、生年月日、運転免許証番号、または同様のデータです。この補足データ値は、更新する名前レコードを識別するためのキーとなります。また、補足データ値があることで、通常は名前に関して

より完全な情報を取得できるようになり、この情報を加重やフィルタリングなどの検索後タスクで使用できます。このデータ・リスト上の名前のうち、一致する補足データ値が含まれる名前がすべて更新されます。

- 名または姓、あるいはその両方を変更する場合は、名前が既に解析されていなければならないことに注意してください。(この更新の前に、`analyzeForSearch()` メソッドを使用して名前を準備できます。このメソッドは、名前を名前フィールドに解析し、名前フィールドそれぞれの国/地域別情報を分類します。)

このタスクについて

既存のデータ・リスト上の名前またはその関連情報を更新するには、IBM NameWorks の Scoring クラスの `updateNameInDatalist()` メソッドを使用し、以下の値をメソッドに渡します。

- データ・リストの名前
- オリジナル補足データ値 (*originalData* 値)。(この補足データ値が含まれるすべてのレコードが更新され、オリジナル補足データ値はこのデータ値に置き換えられます。)
- データ・リスト上の変更する名フィールド値と姓フィールド値。
- オリジナル名前解析、または名前のオリジナル・スクリプト、あるいはその両方(この情報はオリジナル項目を見つけるためのキーであるため、この追加にはオリジナルの名前またはスクリプトの代替解析のフラグを立てる必要があります)。
- *originalData* (オリジナル補足情報) を置き換える何らかの補足データ。(例えば、この個人に関連付けられた運転免許証が変わった場合。)
- 名と姓の国/地域別情報コード。(このような国/地域別情報コードがない場合は、値 `-1` を渡します。この値は、名フィールドと姓フィールドの国/地域別情報を最初に分類するようメソッドに通知する意味を持ちます。)

タスクの結果

`updateNameInDatalist()` メソッドは、指示されたデータ・リスト上の同じ補足データ情報が含まれるすべての名前レコードを更新します。名または姓(あるいはその両方)の国/地域別情報コードとして `-1` 値を渡した場合、このメソッドは最初に国/地域別情報コードを分類し、識別された国/地域別情報コードで名前を更新します。

データ・リストからの名前の削除

システム管理者は、データ・リストを保守する際に、データ・リスト暫定更新の一部として、データ・リストから名前を削除しなければならないこともあります。データ・リストはメモリー常駐なので、アクティブ検索がデータ・リストにアクセス中であっても、いつでもデータ・リストから名前を削除できます。削除された名前に対する後続の検索では、そのデータ・リストでは一致なしと示されます。

始める前に

- データ・リストの名前が分かっている必要があります。(既存のすべてのデータ・リスト名のリストに戻すには、`getDatalistNames()` メソッドを使用します。)
- 名前に関連付けられたオリジナル補足データ値 (*originalData* 値) が分かっている必要があります。このデータ値は多くの場合、生年月日、運転免許証番号、ま

たは同様のデータです。この補足データ値は、削除する名前レコードを識別するためのキーとなります。このデータ・リスト上の名前のうち、一致する補足データ値が含まれる名前がすべて、データ・リストから削除されます。

このタスクについて

既存のデータ・リスト上の名前とその関連情報を削除するには、IBM NameWorks の Scoring クラスの `updateNameInDatalist()` メソッドを使用し、以下の値をメソッドに渡します。

- データ・リストの名前
- 名前レコードに関連付けられた補足データ

指示したデータ・リストから同じ補足データ情報が含まれる名前レコードをすべて削除するには、`deleteNameFromDatalist()` メソッドを使用します。

IBM NameWorks のマイグレーション

IBM InfoSphere Global Name Management のこのリリースには、IBM NameWorks API に対する多くの変更が含まれています。この情報を使用して、既存の API をマイグレーションするために必要な変更について学習します。

新しい Java メソッドおよびオブジェクト

以下の表は、このリリースでの新しい Java メソッドおよびオブジェクトを示しています。

表 7. 新しい Java メソッドおよびオブジェクト

Java メソッドまたはオブジェクト	説明
<code>createName()</code> メソッド	<code>search()</code> 、 <code>compare()</code> 、 <code>addName()</code> 、および <code>updateName()</code> メソッドは、Name オブジェクトをパラメーターとして受け入れるようになりました。そのため、このバージョンの検索を呼び出す前に、 <code>createName()</code> メソッドを使用して Name オブジェクトを作成する必要があります。
Name オブジェクト	直前のバージョンでは、名前、姓、または名のパラメーターの入力として、名前ストリングが使用されていました。このリリースでは、Name オブジェクトが導入され、名前フィールド (名および姓)、国/地域別情報、および名前の <code>NameCategory</code> がカプセル化されています。このクラスは、入力ファイルから照会名またはデータ・リスト名のいずれかで入力される名前を表します。

表 7. 新しい Java メソッドおよびオブジェクト (続き)

Java メソッドまたはオブジェクト	説明
NameCategory オブジェクト	<p>NameCategory は、IBM InfoSphere Global Name Management 製品でサポートされる名前カテゴリーを記述するために使用されます。以下の名前カテゴリーがこのリリースでサポートされます。</p> <ul style="list-style-type: none"> • 不明 • 個人 • 組織 • 両方 <p>createName() メソッドを使用して名前を作成するときに、名前カテゴリーを指定します。</p>
NameCategorySet オブジェクト	<p>NameCategorySet は、1 つ以上の NameCategory 値の集合を表します。このデータ・タイプは、特定の名前カテゴリーを示し、search() メソッドを呼び出した後にどの名前カテゴリーを返すかを示します。</p>

変更された Java メソッドおよびオブジェクト

いくつかの IBM NameWorks API データ構造がデータ・クラスになりました。各オブジェクトは Name クラスから派生したものであるため、名前情報および国/地域別情報が含まれるいくつかのメソッドを継承します。いくつかのクラスでは get() メソッドを呼び出してメンバー・データにアクセスしなければならなくなりました。それらの各クラスを以下の表で示します。

表 8. 変更された Java メソッドおよびオブジェクト

Java メソッドまたはオブジェクト	変更の説明
addName() メソッド	<p>このメソッドには 2 つのバージョンが存在し、両方ともパラメーターとして Name オブジェクトが必要です。元の名前および音訳されたバージョンの名前をリストに追加するかどうか、あるいは音訳されたバージョンのみを追加するのかが選択できます。</p>

表 8. 変更された Java メソッドおよびオブジェクト (続き)

Java メソッドまたはオブジェクト	変更の説明
analyzeForSearch() メソッド	<p>このメソッドには 2 つのバージョンが存在します。1 つのバージョンは氏名パラメーターを受け入れ、2 つ目のバージョンは名パラメーターと姓パラメーターを受け入れます。</p> <ul style="list-style-type: none"> 氏名パラメーターを受け入れるメソッドが呼び出されると、NameSifter を使用してまず名前がカテゴリー化されます。その後の処理は、名前カテゴリー (個人名、組織名、または両方) に基づいて決定されます。結果データを使用して関連 QueryName オブジェクトが作成されます。 名パラメーターおよび姓パラメーターを受け入れるメソッドが呼び出された場合、名前は個人名として扱われます。名前が解析および分類され、結果データを使用して関連 QueryName オブジェクトが作成されます。
compare() メソッド	<p>このバージョンには Name オブジェクトが必要で、複数の比較タイプをサポートします。queryName の NameCategory に基づいて 2 つの名前が比較され、比較の結果は CompareData オブジェクトで返されます。名前について名、姓、および国/地域別情報の各パラメーターを個別に受け入れる代わりに、比較の際に 2 つの Name オブジェクト (1 つは左側の名前、もう 1 つは右側の名前) が使用されます。</p>
search() メソッド	<p>search() メソッドは、Name オブジェクトをパラメーターとして受け入れるようになりました。そのため、このバージョンの検索を呼び出す前に、createName() メソッドを使用して Name オブジェクトを作成する必要があります。</p>
updateName() メソッド	<p>このメソッドには 2 つのバージョンが存在し、両方ともパラメーターとして Name オブジェクトが必要です。元の名前および音訳されたバージョンの名前でリストを更新するかどうか、あるいは音訳されたバージョンのみで更新するのかが選択できます。</p>
CategorizeData オブジェクト	<p>CategorizeData は、categorize() メソッドを呼び出した結果として作成されます。名前は個人名、組織名、両方、または不明のいずれかにカテゴリー化されます。</p>

表 8. 変更された Java メソッドおよびオブジェクト (続き)

Java メソッドまたはオブジェクト	変更の説明
CompareData オブジェクト	CompareData は、compare() メソッドを呼び出した結果として作成されます。このオブジェクトには、2 つの Name オブジェクトと名前比較の結果が含まれます。オブジェクトの 1 つは照会名として使用され、もう 1 つは評価名として使用されます。このオブジェクトの構造が圧縮され、名前の国/地域別情報が Name オブジェクトを介して返されるようになりました。また、比較のために名前のどのパラメーター (左または右) が使用されたかを示すフラグを返すように指定できます。
OriginalName オブジェクト	OriginalName には、名前、名、および姓の各パラメーターが含まれます。さらにこのオブジェクトには、名前の各部分に対応する国/地域別情報に加え、正規化状態および代替状態に関する情報が含まれます。
QueryName オブジェクト	QueryName は、名前一致の信頼性を判別するために使用されます。このオブジェクトは、名前比較の際に使用できるタイトルおよび修飾子ストリングを受け入れます。
SearchMatch オブジェクト	SearchMatch には、一致した名前レコードが入ります。検索操作で返されるそれぞれの SearchMatch オブジェクトは、SearchResults オブジェクト内に含まれるリストに入力されます。

検索のための名前の準備

analyzeForSearch() メソッドを使用して、名前を検索トランザクションで使用できるように準備します。このメソッドは、個人名か組織名かを判別することによって名前をカテゴリー化し、個人名を名フィールドと姓フィールドに解析し、個人名の各フィールドの国/地域別情報種別コードを判別します。名前が組織名であると NameSifter が判別した場合は、名前カテゴリー情報のみが戻されます。

このタスクについて

analyzeForSearch() メソッドは、UTF-16 エンコードの入力ストリングのみをサポートします。サポートされないエンコードが入力ストリングに使用されている場合、IBM NameWorks からエラーが通知されるか、好ましくない結果になります。検索用に名前を準備するには、IBM NameWorks の Scoring クラスの analyzeForSearch() メソッドに、以下の値を渡します。

- 解析する氏名。フルネームで表します。例えば、ROBERT E JONES のようにします。名前は、parse() メソッドにストリング 値として渡す必要があります。

- **alternateThreshold** 値を表す 0 から 100 までの整数 (0 を渡した場合、代替解析は常に抑止されます)。この値は、初期解析で名前が再配列されて代替解析が考慮に入れられるために必要な最小信頼値です。

タスクの結果

`analyzeForSearch()` メソッドは、名前をまずカテゴリ化してから音訳します。名前音訳は、特定の書記体系または文字エンコード規則から別の書記体系または文字エンコード規則に名前を変換するプロセスです。例えば、名前音訳を行うことによって、アラビア文字で書かれた名前を分析して、ラテン・アルファベットで書かれた類似の名前と突き合わせるすることができます。このメソッドは次に、入力ストリングを解析し、入力ストリングとその音訳済みバージョンごとに、`QueryName` オブジェクトのコレクションを戻します。

シナリオ: 名前の検索

IBM InfoSphere Global Name Management には、名前を検索するためのさまざまな方法があります。以下のシナリオでは、名前を検索するときに行うことができる共通したいくつかの実装方法を説明します。

名前検索のための名前オブジェクトの作成

以下のシナリオでは、名前情報のインターフェースの計画方法に基づいて、検索の準備として名前オブジェクトを作成するための異なるメソッドを説明します。

両方のシナリオとも、検索の準備としてどのように名前を作成できるかを説明していますが、シナリオによってこの目的の達成方法が異なります。最初のシナリオでは、`search()` メソッドを呼び出す前に `createName()` メソッドを使って `Name` オブジェクトを作成する方法を説明します。2 番目のシナリオでは、`analyzeForSearch()` メソッドを使用して、名前検索で使用される `QueryName` オブジェクトを作成します。

`search()` メソッドを使用するとき、`createName()` メソッドを使用して、個人名および組織名の `Name` オブジェクトを最初に作成する必要があります。`Name` オブジェクトは、名前の名前フィールド (名および姓)、国/地域別情報、および名前カテゴリーをカプセル化したものです。この点、IBM InfoSphere Global Name Management バージョン 4.2 以前のリリースでは名前ストリングを受け入れていました。`Name` クラスは、入力ファイルから照会名またはデータ・リスト名のいずれかで入力される名前を表します。

または、`analyzeForSearch()` メソッドの結果を使用して `search()` メソッドを呼び出すこともできます。`QueryName` オブジェクトは `analyzeForSearch()` によって返され、`QueryName` オブジェクトは `Name` クラスから派生したため、`QueryName` オブジェクトのリストに含まれる各要素は、`search()` メソッドに送信できます。

`createName()` メソッドは、`NameCategory` が個人名を示す場合にのみ、名前に対する最も信頼性の高い解析結果と、名と姓の両方における推奨する国/地域別情報を生成します。一方、`analyzeForSearch()` メソッドは、それぞれの名と姓に対して最大

6 つの国/地域別情報を提供し、それぞれの結合された名前に対して可能な最大 3 つの解析を提供します。使用するメソッドは、達成しようとする結果によって異なります。

QueryName オブジェクトの作成と名前の検索

以下のシナリオでは、名前検索の準備として、analyzeForSearch() メソッドを使って QueryName オブジェクトを作成する方法を説明します。

始める前に

以下のガイドラインを使用して、検索機能を使用する対象となる QueryName オブジェクトを作成します。QueryName オブジェクトは Name クラスから情報を継承しますが、analyzeForSearch() メソッドを介してのみ作成されます。

analyzeForSearch() メソッドは、createName() メソッドのように、名前をカテゴリー化、解析、分類、および音訳します。さらに、analyzeForSearch() は、それぞれの名と姓に対して最大 6 つの国/地域別情報を返し、それぞれの結合された名前に対して可能な最大 3 つの解析を返します。

手順

1. analyzeForSearch() メソッドを呼び出します。
2. オプション: QueryName オブジェクトの国/地域別情報コードをオーバーライドする場合、createName() メソッドを呼び出して、analyzeForSearch() で提供されたものとは異なる国/地域別情報を使って Name オブジェクトを作成する必要があります。createName() メソッドを使用すると、search() メソッドで使用される複数の Name オブジェクトを作成できます。
3. search() メソッド呼び出して、検索操作を実行します。名前の NameCategory によって、検索で使用される一連の比較パラメーターが決まります。
 - a. オプション: getDataListNames() メソッドを呼び出して、システム内で使用可能なすべてのデータ・リストの名前を返します。
 - b. オプション: getSearchStrategyNames() メソッドを呼び出して、システム内で使用可能なすべての検索方針の名前を返します。
4. オプション: dataFetch() メソッドを呼び出して、固有の名前の一致に関連付けられた元の名前データおよび名前レコードの補足のデータを取得します。

タスクの結果

QueryName オブジェクトのリストが analyzeForSearch() から返されます。

Name オブジェクトの作成と名前の検索

以下のシナリオでは、名前検索の準備として、createName() メソッドを使って Name オブジェクトを作成する方法を説明します。

始める前に

以下のガイドラインを使用して、追加、更新、検索、および比較の各機能を使用する対象となる Personal オブジェクトまたは Organization Name オブジェクトを作成します。IBM NameWorks では、NameSifter を使用して名前を自動的にカテゴリー化するため、名前カテゴリー（「個人名」または「組織名」）を指定する必要はありません。ただし、名前カテゴリーによって名前の処理方法が決まります。例

例えば、NameCategory に「個人名」を指定すると、createName() メソッドによって名前が音訳、解析、および分類されますが、「組織名」では、音訳のみが実行されます。

手順

1. createName() メソッドを呼び出して Name オブジェクトを作成します。

オプション	説明
名前について何も分からない場合	単一の名前ストリングを createName() メソッドに渡します。IBM NameWorks によって名前が音訳され、NameCategory が判別されます。 <ul style="list-style-type: none"> 名前が個人名であると判定された場合、IBM NameWorks によって名前が解析および分類されます。 名前が組織名であると判定された場合、受け取る名前の国/地域別情報は「未確定」になります。
名前が個人名であるか組織名であるかが分かっている場合	NameCategory を指定します。これで名前の処理方法が決まります。
名および姓が分かっている場合	これらの値を createName() メソッドに渡します。これで名前が個人名であることが決まります。Name オブジェクトから国/地域別情報が必須になるときに、名前フィールドが分類されます。
名、姓、および国/地域別情報が分かっている場合	これらの値を createName() メソッドに渡します。これで名前の NameCategory の割り当ては、自動的に個人名になります。

2. search() メソッド呼び出して、検索操作を実行します。名前の NameCategory によって、検索で使用される一連の比較パラメーターが決まります。
 - a. オプション: getDataListNames() メソッドを呼び出して、システム内で使用可能なすべてのデータ・リストの名前を返します。
 - b. オプション: SearchStrategyNames() メソッドを呼び出して、システム内で使用可能なすべての検索方針の名前を返します。
3. searchOpt= パラメーターを指定して、検索対象となる NameCategory のタイプを示します。この値は、IBM NameWorks に渡される検索方針で指定できます。
 - 1 = 個人名
 - 2 = 組織名
 - 3 = すべての名前カテゴリー

タスクの結果

検索結果は SearchResult オブジェクトを介して返されます。ここには SearchMatch オブジェクトのリストが入ります。各 SearchMatch オブジェクトには、以下の情報が含まれます。

- 一致した名前が見つかったデータ・リスト名

- 名前に関連付けられた補助データ
- 氏名の類似性スコア
- 名の類似性スコア
- 姓の類似性スコア
- 一致する名前が代替解析で見つかるかどうか
- 一致する名前が正規化名前項目で見つかるかどうか
- この固有の名前に関連付けられた、一致する名前レコードの数

SearchMatch オブジェクトは、OriginalName クラスから派生するため、以下の情報も返します。

- 名前カテゴリー
- 氏名
- 名
- 姓
- 補足データ

IBM NameWorks を使用した名前の検索

IBM NameWorks は、氏名検索および固有名検索をサポートします。 IBM NameWorks は、氏名検索では、データ・リストに出現する一致する名前レコードをすべて検索結果で返します。固有名検索では、一致する固有の氏名それぞれのコピーが 1 つずつ、そしてデータ・リストにその名前が出現する回数が返されます。

注: IBM NameWorks で検索を実行する前に、システム管理者は、IBM NameWorks 構成ファイルの項目を使って、名前の外部ファイルを IBM InfoSphere Global Name Management データ・リストと関連付ける必要があります。

固有名検索は、多くの共通した名前が何百回、何千回と出現する可能性があるような大規模な名前リストを検索するときに便利です。そのような繰り返しがあると、普通の方法では同一の名前が表示された大量の検索結果を調べなければならなくなり、IBM NameWorks の検索ロジックにとってもユーザーにとっても問題発生の原因になりかねません。固有名検索を実行した場合には、その後で dataFetch() メソッドを使用すると、さらに調べる名前を選択できます。

物理的な各外部ファイルは、別個の 1 つの論理的な項目にすることもできます (例えば、顧客用に 1 つのファイル、従業員用に 1 つのファイルなど)。ただしこれらの各外部ファイルは、名前を集めた 1 つの論理的な集合として理解される大きな 1 つのファイルのサブセットです。大きなファイルを一連の小さなファイルに分割すると検索を並列処理できる利点があるため、IBM NameWorks およびその検索コンポーネントである Distributed Search で複数のサブセットを同時に検索した後、すべての結果を単一の応答に統合できます。この物理/論理の関連は、IBM NameWorks の構成ファイルで設定します。

名前データが、別個のステップではなく IBM NameWorks の初期化時に前処理される、氏名検索を行う組み込み検索を使用できます。名前の前処理を実行する複数の検索機能を単一のプロセスに結合すると、大規模なデータ・リストの氏名検索に

伴いがちな、通信および管理上のオーバーヘッドが削減されます。 IBM NameWorks 構成ファイルの [DataList] セクション内の設定により、データ・リストが組み込みであるか外部であるかが決まります。追加設定により、初期化中に組み込みデータ・リストがロードされるときに実行する名前前処理のタイプが制御されます。

検索方針の管理

ユーザーが一貫して簡単に名前を検索できるように、検索比較パラメーター値のセットを検索方針に定義します。検索ごとの検索方針は、名前検索でそれを使用する前に定義する必要があります。

このタスクについて

検索方針は、構成ファイル内、または Strategy クラスを使用して作成する Strategy オブジェクト内に定義できます。検索操作が呼び出されるときに使用する検索方針の名前を指定できます。検索方針で定義されている比較パラメーターが検索に使用されます。

検索方針

検索方針 とは、Scoring クラスの search() メソッドが使用できる、検索比較パラメーター値の集合です。検索方針を使用して、search() メソッドが使用可能な一連の比較パラメーター値を定義します。それぞれの検索方針には固有の名前があります。

IBM NameWorks の出荷時の構成ファイルには、システム管理者が使用したりカスタマイズしたりできる、次のような 3 つのサンプル検索方針が含まれています。

- Standard (デフォルトの比較値が含まれる)
- Broad (検索を広げる比較パラメーター値が含まれる)
- Narrow (検索を制限する比較パラメーター値が含まれる)

検索方針は必要に応じていくつでも作成できます。また、クライアント・アプリケーションの必要や実行する検索のタイプに応じてさまざまな検索方針を作成できます。

注: MinScore 比較パラメーターは以前のリリースの NAME_THRESH 比較パラメーターに対応し、構成ファイルの他のセクションで別個に設定する必要はありません。両方のパラメーターを設定すると、IBM NameWorks は MinScore を使用し、NAME_THRESH はエラーとして報告されます。

相対調整係数は、計算済みの比較パラメーターのスコアを調整します。これらの係数は、以下の例では、パラメーター名の後に _ADJ として出現します。

Broad 検索方針のサンプル:

Broad 検索方針を使用すると、検索での名前のリコールを高くすることができますが、その代償として精度は低くなります。検索結果には全体的により多くの名前が含まれることとなりますが、それと共に、検索に関連する名前のパーセンテージは低くなる可能性があります。以下の例は、**Broad** 検索方針がどのようなものであるかを示しています。

```

[Strategy:Broad]
  MinScore=65

[GNParms:Broad]
  ANCHOR_FACTOR=0.97
  COMPRESSED_SCORE_MAX=1.00
  DO_COMPRESSED_SCORE=Y
  FIELD_THRESH=0.50
  FIELD_WEIGHT=0.50
  NAME_UNKNOWN_SCORE=0.75
  NO_NAME_SCORE=0.75
  INITIAL_INITIAL_SCORE=1.00
  INITIAL_TOKEN_SCORE=0.85
  MATCH_INITIALS=Y
  OOPS_FACTOR=0.95

[SNParms:Broad]
  ANCHOR_FACTOR=0.97
  COMPRESSED_SCORE_MAX=1.00
  DO_COMPRESSED_SCORE=Y
  FIELD_THRESH=0.50
  FIELD_WEIGHT=0.50
  NAME_UNKNOWN_SCORE=0.75
  NO_NAME_SCORE=0.75
  OOPS_FACTOR=0.95

```

Narrow 検索方針のサンプル:

Narrow 検索方針を使用すると、検索での名前のリコールが低くなりますが、精度は高くなります。検索結果に含まれる名前は全体的に少なくなります。検索に関連した名前がより高いパーセンテージで返されます。以下の例は、**Narrow** 検索方針がどのようなものであるかを示しています。

```

[Strategy:Narrow]
  MinScore=80

[GNParms:Narrow]
  ANCHOR_FACTOR=0.85
  COMPRESSED_SCORE_MAX=0.95
  DO_COMPRESSED_SCORE=Y
  FIELD_THRESH=0.70
  FIELD_WEIGHT=0.50
  INITIAL_INITIAL_SCORE=0.75
  INITIAL_TOKEN_SCORE=0.70
  NAME_UNKNOWN_SCORE=0.70
  NO_NAME_SCORE=0.70
  OOPS_FACTOR=0.85

[SNParms:Narrow]
  ANCHOR_FACTOR=0.85
  COMPRESSED_SCORE_MAX=0.95
  DO_COMPRESSED_SCORE=Y
  FIELD_THRESH=0.70
  FIELD_WEIGHT=0.50
  NAME_UNKNOWN_SCORE=0.70
  NO_NAME_SCORE=0.70
  OOPS_FACTOR=0.85

```

IBM NameWorks 構成ファイルを使用した、検索方針の作成と変更

システム管理者は、必要に応じて検索方針を作成または変更して、名前検索を補助したり、特定のユーザー・ロールまたはビジネス・ルールのセットにマップしたり

します。検索方針は IBM NameWorks 構成ファイルを使用して管理できます。そのため、変更を実施するためには、その前に IBM NameWorks 構成ファイルを再ロードしなければなりません。

手順

1. テキスト・エディターで IBM NameWorks 構成ファイルを開きます。
2. 構成ファイルの適切なセクションを作成または変更します。
 - 新規検索方針を作成するには、以下の新しいセクション見出しを作成し、セクション見出しごとに適切な項目と値を指定します。

```
[Strategy:name]
  GNCulture=-1
  SNCulture=-1
  ONCulture=0
  MinScore=-1
  MaxReplies=-1
  SearchOpt=1
  IncludeTAQs=
[GNParms:name]
...
[SNParms:name]
...
[ONParms:name]
...
```

name は新規検索方針の固有の名前です。各セクション見出しが該当検索方針名を指定するようにしてください。そうしないと、セクションは無視されることとなります。

[Strategy:] 見出しは、検索方針を作成または実行するための唯一の必須セクション見出しです。他のセクション見出しを含めない場合、検索方針は、それらのセクションのパラメーターとしてデフォルトの検索パラメーターを使用します。

- 既存の検索方針を変更するには、該当する検索方針セクション見出し内で変更を行います。各セクション見出しが該当検索方針名を指定するようにしてください。そうしないと、セクションは無視されることとなります。
3. 構成ファイルを保存します。
 4. 変更を有効にするために、IBM NameWorks を再始動して構成ファイルを再ロードします。

Strategy クラスを使用した検索方針の作成

IBM NameWorks 構成ファイルの [Strategy:] セクションを使用する代わりに、Strategy クラスを使用して検索方針を定義できます。

このタスクについて

以下の手順には Java™ 言語を使用するコード例が含まれていますが、C++ で検索方針を作成するためのステップもこれと同じです。

手順

1. 開発アプリケーションを使用して、Strategy オブジェクトを作成し、それに固有の名前を付けます。

```
Strategy broad = new Configuration.Strategy();
```

2. さまざまなメソッドを呼び出して、デフォルトの最小スコアしきい値、返す一致結果の最大数、検索する名前カテゴリーなど、検索方針の検索比較パラメーターの値を指定します。以下に例を示します。

```
broad.setMinScore(70)
.setMaxReply(1000)
.setSearchOptions(EnumSet.of(NameCategory.PERSONAL));
Configuration configuration = new Configuration();
configuration.addStrategy("Broad", broad);
```

3. Strategy オブジェクトを保存します。
4. Configuration クラスを使用して Configuration オブジェクトを作成し、検索方針を Configuration オブジェクトに追加します。

構成ファイルからの検索方針の削除

検索方針を使用しなくなった場合、または検索方針が重複している場合には、その検索方針を構成ファイルから削除します。変更を有効にするためには、構成ファイルを再ロードする必要があります。

手順

1. テキスト・エディターで IBM NameWorks 構成ファイルを開きます。
2. 削除する検索方針の名前が付いた [Strategy:] セクション・ヘッダーを見つけ、そのセクション・ヘッダーの中の情報をすべて削除します。検索方針ごとに、存在し得る以下のセクション見出しをすべて削除したことを確認してください。

```
[Strategy:name]
[GNParms:name]
[SNParms:name]
[ONParms:name]
```

name は、削除する検索方針の固有の名前です。

3. IBM NameWorks 構成ファイルを保存します。
4. 変更を有効にするために、IBM NameWorks を再始動して構成ファイルを再ロードします。

比較パラメーターのオーバーライド

IBM NameWorks では、比較パラメーターの集合を名前付き検索方針としてまとめ、検索オーバーライド情報の処理を簡素化することができます。デフォルトの比較パラメーターは、さまざまな検索方針を作成することにより、トランザクション・ベースでオーバーライドできます。

比較パラメーター・オーバーライドのオプション

比較パラメーターのオーバーライドを指定するには、IBM NameWorks 構成ファイル内の検索方針、CompParmsOverride クラス、または Configuration クラスを使用することができます。各オプションは、名前検索を処理するクライアント・アプリケーションを作成する方法に応じて、オーバーライドを指定するためのさまざまな方法を提供します。

例えば、英語圏の名前の姓フィールドおよび名フィールドのデフォルト・フィールドしきい値が 0.49 (FIELD_THRESHOLD = 0.49) であるとしみます。検索方針で FIELD_THRESHOLD = 0.60 と指定して、IBM NameWorks の構成ファイルにあるデ

フォルト値を変更すると、名前しきい値を 22% 増やすことができます。このオーバーライドは、比較パラメーターの新しい値を指定するので、絶対指定となります。

同様に、計算されるスコアを調整するために使用される相対調整係数を含めることもできます。直前の例で `FIELD_THRESHOLD_ADJ = 0.60` と指定して調整係数を入力した場合、新しい値はデフォルト値と調整係数を乗算して計算されます。

`FIELD_THRESHOLD = 0.49 * 0.60 = 0.29`

構成ファイル内のデフォルト値は変更されずに、調整された値 0.29 が検索の期間中使用されます。

検索方針のオーバーライド

検索方針を作成して、任意またはすべてのデフォルトの比較パラメーターをオーバーライドすることもできますし、どれもオーバーライドしないようにすることもできます。オーバーライドは検索が行われている間のみ使用され、次のトランザクションではデフォルト値に戻ります。

検索方針で指定するオーバーライドは絶対指定と相対指定の両方がありますが、Embedded Search で使用されるデータ・リストによって指定する場合など、サーバー・サイドではオーバーライドは絶対指定です。

サーバーの比較パラメーター値は固定であるため、値の `FIELD_THRESHOLD = 0.49` は、比較パラメーターのデフォルト値を含む IBM NameWorks 構成ファイル内で値を変更しない限り変更できません。サーバー上で比較パラメーターの値を変更すると、それらの値を構成ファイル内で変更しない限り、すべてのトランザクションは更新された値を使用します。

Distributed Search または Embedded Search に固有のデフォルト比較パラメーターを使用する場合、検索方針の名前は空にして渡します。検索方針名を空にすると、デフォルトの比較パラメーターを使用するようにシグナル通知されます。

データ・リストごとのオーバーライド

データ・リストごとのオーバーライドは、組み込みデータ・リストだけに適用されます。1 つ以上の `CompParmsDefaults` 項目を構成ファイルの `[DataList:]` セクションに追加することにより、個別のデータ・リストの比較パラメーター・オーバーライドを指定できます。このオプションを使用すると特定のデータ・リストにデフォルト値を設定できますが、`CompParmOverrides` クラスで示される検索方針とオーバーライドでは個別の照会のデフォルト値だけがオーバーライドされます。

CompParmOverrides クラスのオーバーライド

`CompParmOverrides` クラスを使用して、個別の検索照会のオーバーライドを提供できます。このクラスには、検索方針のオーバーライド情報が含まれていますが、デフォルト値の情報はありません。このクラスによってオーバーライドを示すには、`CompParmOverrides` オブジェクトを作成して、1 つ以上のオーバーライド・メソッド (`addSurnameOverride()` など) を使用します。オーバーライドを IBM NameWorks 構成ファイルに格納される検索方針に指定するときや、XML メッセージによって Distributed Search サーバーに送信するときと同じ方法で、名前/値ペアとしてオーバーライド

を指定します。絶対オーバーライドおよび相対オーバーライドの両方が、`CompParmOverrides` クラスでサポートされます。

このクラスを使用して指定したオーバーライドは、`Scoring.search()` および `Scoring.compare()` メソッドに送られたときに処理されます。比較パラメーターの名前または値のエラーは、処理中に報告されます。これはオーバーライドが検索方針で処理される時の方法です。

Configuration クラスのオーバーライド

`Configuration` クラスは、IBM NameWorks 構成ファイルを置き換えて、構成をデータベースに格納したり構成を動的に指定したりする機能を提供します。代わりに、外部構成ファイルを使用してデフォルト値を `[Strategy:]` セクションに指定してから、このクラスを使用して特定のデータ・リストのオーバーライドを指定することもできます。`Configuration` クラスでオーバーライドを指定するには、以下のいずれかのオプションを使用します。

- `Strategy` クラスを使用して検索方針を作成し、オーバーライド・メソッド (`addSurnameOverride()` など) を使用してオーバーライドを指定してから、`addStrategy()` メソッドによって検索方針を `Configuration` オブジェクトに追加します。
- 外部オーバーライド・ファイルを使用して、`setDefaultCompParmOverrides()` メソッドによりそのファイルを呼び出します。

検索方針を使用した比較パラメーター・オーバーライドの指定:

絶対オーバーライド、相対オーバーライド、またはデータ・リストごとのオーバーライドを IBM NameWorks 構成ファイルに指定することにより、または `Configuration` クラスを使用して `Configuration` オブジェクトを作成することにより、IBM NameWorks の比較パラメーターをオーバーライドできます。

このタスクについて

相対調整係数は、計算済みの比較パラメーターのスコアを調整します。これらの係数は、パラメーター名の後に `_ADJ` として出現します。

手順

1. テキスト・エディターで IBM NameWorks 構成ファイルを開きます。
2. 構成ファイルの `[Strategy:name]` セクション (`name` は検索方針の名前) は、オーバーライドするパラメーターを変更します。

オプション	説明
絶対オーバーライドを指定するには	変更する比較パラメーターの値を新しい値に置き換えます。
相対オーバーライドを指定するには	変更するパラメーターに <code>_ADJ</code> を付加して、元の値を調整するための値を指定します。

3. 検索方針を保存して、IBM NameWorks を再始動します。
4. オーバーライドを有効にするために、新しい検索を実行します。

CompParamsOverrides クラスを使用したオーバーライドの指定:

CompParamsOverrides オブジェクトを作成して、どのオーバーライドを組み込むかを示すことにより、IBM NameWorks の比較パラメーター・オーバーライドを指定できます。その後、そのオーバーライドを Scoring.search() および Scoring.compare() メソッドに送り、名前比較または名前検索の一部として組み込まれるようにします。

手順

1. 開発アプリケーションを使用して、CompParamsOverrides オブジェクトを作成します。

```
CompParamsOverrides overrides = new CompParamsOverrides();
```

2. さまざまなメソッドを呼び出して、CompParamsOverrides オブジェクトの比較パラメーター・オーバーライドを指定します。name はオーバーライドする比較パラメーターの名前、new_value は使用する値です。

```
CompParamsOverrides overrides = new CompParamsOverrides();  
overrides.addSurnameOverride("name", "new_value");  
overrides.addSurnameOverride("OOPS_FACTOR", "0.60");  
overrides.addSurnameOverride("NAME_UNKNOWN_SCORE", "0.45");  
...
```

3. CompParamsOverrides オブジェクトを Scoring.search() メソッド、Scoring.compare() メソッド、またはその両方に渡して、名前比較および名前検索の一部として処理されるようにします。

Configuration クラスを使用したオーバーライドの指定:

Configuration クラスを使用して、IBM NameWorks の比較パラメーター・オーバーライドを指定できます。外部構成ファイルに [Strategy:] セクションを作成するときと同じ方法で、Configuration クラスを使用して検索方針を作成できます。

始める前に

Configuration オブジェクトは、クライアント・アプリケーションの一部として作成しなければなりません。このオブジェクトには比較パラメーターのオーバーライド情報が格納されます。さらに、以下のいずれかのオプションによって、名前検索に関連付けられる値を指定する必要があります。

- IBM NameWorks パラメーターのためのデフォルト値を含む外部構成ファイルを使用する
- Strategy クラスを使用して作成する検索方針を使用する

手順

1. Configuration オブジェクトを開発アプリケーションで開きます。
2. 実装するオーバーライドを指定します。

オプション	説明
外部構成ファイルを使用する場合	<p><code>setDefaultCompParmOverrides()</code> メソッドを使用して、<code>Datalist</code> オブジェクトにオーバーライドを指定します。</p> <p>以下の例では、<code>compparms.config</code> という名前のオーバーライド・ファイルを使って、<code>Customers</code> という名前のデータ・リストを含む <code>Configuration</code> オブジェクトを作成します。</p> <pre>Configuration configuration = new Configuration(); Datalist customers = configuration.addDatalist("Customers"); customers.setDefaultCompParmsOverridesFile("compparms.config");</pre>
Strategy クラスを使用する場合	<p>1 つ以上のオーバーライド・メソッドを使用して、<code>Strategy</code> オブジェクトにオーバーライドを指定します。</p> <p>以下の例では、<code>broad</code> 検索方針を指定して、姓のオーバーライドを組み込みます。</p> <pre>Configuration configuration = new Configuration(); Strategy broad = configuration.addStrategy("Broad"); broad.addSurnameOverride("OOPS_FACTOR", "0.60");</pre>

3. 開発アプリケーションを保存して、IBM NameWorks を再始動します。
4. オーバーライドを有効にするために、新しい検索を実行します。

検索のための名前の準備

`analyzeForSearch()` メソッドを使用して、名前を検索トランザクションで使用できるように準備します。このメソッドは、個人名か組織名かを判別することによって名前をカテゴリー化し、個人名を名フィールドと姓フィールドに解析し、個人名の各フィールドの国/地域別情報種別コードを判別します。名前が組織名であると `NameSifter` が判別した場合は、名前カテゴリー情報のみが戻されます。

このタスクについて

`analyzeForSearch()` メソッドは、UTF-16 エンコードの入力ストリングのみをサポートします。サポートされないエンコードが入力ストリングに使用されている場合、IBM NameWorks からエラーが通知されるか、好ましくない結果になります。検索用に名前を準備するには、IBM NameWorks の `Scoring` クラスの `analyzeForSearch()` メソッドに、以下の値を渡します。

- 解析する氏名。フルネームで表します。例えば、ROBERT E JONES のようにします。名前は、`parse()` メソッドにストリング 値として渡す必要があります。
- `alternateThreshold` 値を表す 0 から 100 までの整数 (0 を渡した場合、代替解析は常に抑止されます)。この値は、初期解析で名前が再配列されて代替解析が考慮に入れられるために必要な最小信頼値です。

タスクの結果

`analyzeForSearch()` メソッドは、名前をまずカテゴリー化してから音訳します。名前音訳は、特定の書記体系または文字エンコード規則から別の書記体系または文字エンコード規則に名前を変換するプロセスです。例えば、名前音訳を行うことによって、アラビア文字で書かれた名前を分析して、ラテン・アルファベットで書かれた類似の名前と突き合わせるすることができます。このメソッドは次に、入力ストリン

グを解析し、入力ストリングとその音訳済みバージョンごとに、QueryName オブジェクトのコレクションを戻します。

IBM NameWorks を使用した名前の分類、名前の比較、および日付の比較

IBM NameWorks を使用すると、個人または組織として名前进行分类したり、2 つの個人名を比較したり、日付を比較することができます。

名前カテゴリー

名前処理の際、名前は名前カテゴリー、つまり個人名または組織名のいずれかに関連付けられます。これら 2 つのカテゴリーの名前の使用法は類似していますが、これらの名前は重要な相違点に基づいて分離されているため、分析および突き合わせの際には、異なるタイプの言語リソースおよび参照データ・リソースが各カテゴリーの名前に適用されます。

名前をカテゴリー化するとき、IBM InfoSphere Global Name Management のコンポーネントは、名前を以下のカテゴリーに分類します。

- 個人名。他のカテゴリーに属することを示唆する標識が含まれません (例えば、「Linda K. Smith」)
- 組織名。非個人に使用される何らかの形式の標識が含まれます (例えば、「Smith & Company」)
- 不明の名前。つづりの誤りがあるように見える要素、または個人名にも組織名にも通常現れないその他の構造が含まれる要素が含まれます (例えば、「SMI」)
- 両方。職業修飾子を含んだ名前です。名前が個人名から派生したビジネス名である可能性があります (例えば、「Linda Smith Architect」)

名前が個人名以外にカテゴリー化された場合、名前を非個人として修飾する標識またはパターンを示す理由コードがコンポーネントによって提供されます。

個人名:

個人名は、1 つ以上の名、その名前の属する国/地域別環境で姓に相当するものとして使用される家名/グループ名 (部族名や氏族名など) などの要素、およびその人に関連付けられている称号やその他の名前修飾子で構成されます。完全な個人名は個人を表すものであり、場合によって、社会的階級、宗教的背景、政治的背景、教育水準、民族または国/地域の背景、および地域的由来を示す情報が織り込まれていることがあります。

IBM InfoSphere Global Name Management 個人名モデル

個人名について議論したり作業したりするには、その固有の形式に関わらず、一貫性のある用語を使用することが重要です。また、等価の部分と比較できるように、名前を構成要素となる各部に一貫して解析できることも重要です。

IBM InfoSphere Global Name Management の個人名モデルの形状は、実世界のデータ・セットにおけるエンコードにより名前を処理するという必要性に基づいて作成されました。これは、名前の構造を判別するための実際的なアプローチです。例えば、世界の多くの場所で使用される名前には、西洋の感覚における本物の姓が含まれていませんが、それらの名前も、姓が含まれていることを前提とするデータベ

ースに強制的に取り込まれます。そのため、名前処理を一貫性のあるものにする目的で、IBM InfoSphere Global Name Management では、強制的に 2 フィールドの構造を使用します。名前のさまざまな部分がどのフィールドに属するかを判別する方法の一部に、名前の各部分がどの程度の頻度で名フィールドまたは姓フィールドに関連付けられてきたかを調べる方法があります。各フィールドの中で、名前の個々の要素はより大きな単位に解析されます。例えば、姓「de la Salle」は、名前の 3 つの別個の部分としてではなく、1 つの主要な名前語幹と 2 つの接頭部で構成される 1 つの名前句として認識されます。

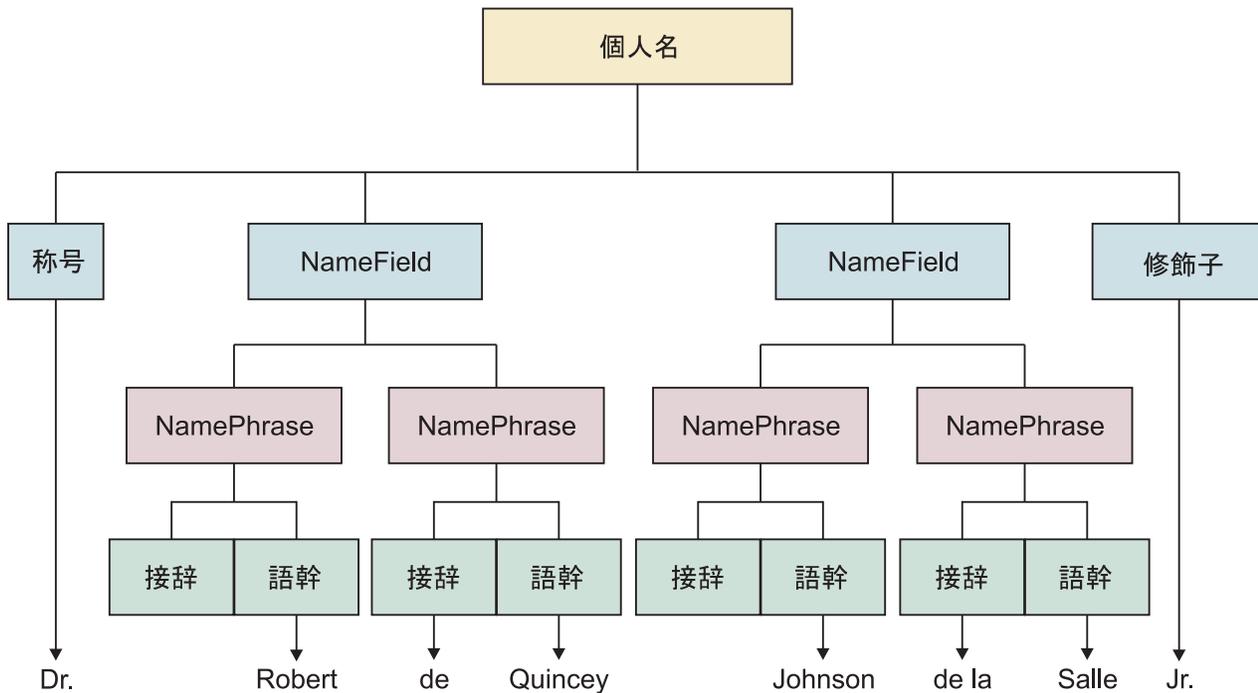


図 3. IBM InfoSphere Global Name Management 個人名モデル

個人名の構造および構成要素

個人名には多くの異なる構成要素が含まれる場合があります。これらの構成要素およびこれらの構造化方法は、国/地域グループによって異なります。

個人名に使用される可能性がある構成要素には、以下のものがあります。

- 名
- 姓
- 家名
- 部族、氏族、またはカーストの名前
- 親戚関係または血統のマーカとなるもの (父親の名前から取られた名前、母親の名前から取られた名前、子供の名前から取られた名前、世代を示すマーカなど)
- 出生順序、性別、宗教、または所属宗教を示す修飾子
- タイトル
- 小辞 (アラビア語の「bin」(息子) および「al」(英語の the に相当) またはスペイン語とフランス語の「de」(の/から))

個人名の構造、つまり名前構成要素の順序も、国または地域グループごとに異なります。

名前の構造には次のような例があります。

名 (1 つ以上) + 家名

- Megan Marie Andrews (ヨーロッパ)
- Fereshteh Gholamzadeh (イラン)
- Rattima Nitisaroj (タイ)
- Hasan Incirlioglu (トルコ)

家名 + 名

- Lim Yauw Tjin (中国)
- Pak Mi-Ok (韓国)
- Suzuki Ichiro (日本)

家名 + ミドルネーム + 名

- Trinh Van Thanh (ベトナム)

名 + 父親の名

- Ahmed bin Eisa (一部のアラブ地域)
- Abdurrahman Wahid (インドネシア)
- Mahmud bin Haji Basir (マレーシア)

名 + 父親の名 + 家名

- Ivan Andreyevich Saratov (ロシア)
- Basimah Ali Al-Qallaf (一部のアラブ国家)

部族名 + 宗教名

- WOUKO Philomene (カメルーン)

名のみ

- Sukarno (インドネシア)
- Habibullah (アフガニスタン)

子孫の名前の言及

- Abu Hassan (文字どおりには *Hassan* の父親 と翻訳される。アラブ国家)

組織名:

組織名 は、何らかの共通した機能を実行するために存在する、構造化された 1 人以上の集団を指す非人格名です。組織には、会社、クラブ、学校、政府機関、政党、ワールド・ワイド・ウェブ開設団体などがあります。通常、組織名には、それらを非個人名として識別する助けになるある種の標識またはパターン、あるいは単語が含まれます。

組織名には、多くの場合 (ただし常にではない)、その機能を示す単語または句が含まれます。例えば、「高校」、「水道設備」、「警察署」、「銀行」などです。

組織名には、命名要素、つまりこの組織を同種の他の組織とは異なるものとして一意的に識別するストリング (文字、単語、または句) も含まれます。例えば、「第一組合銀行」、「Joe's イタリアン・レストラン」、「AAA 洗車場」などがあります。会社などの一部の組織は、政府の規定で「PTY」や「LTD」などの登録状況を示す所定の名前要素があります。

組織名にあるトークンの種類とトークンの組み合わせは、通常、個人名の場合とは外観やパターンが異なります。これらのパターンは、名前が個人名ではなく組織名として分類された理由を識別するコード (名前カテゴリ理由コードと呼ばれる) に対応します。これらの理由コードが組織名を定義するわけではありませんが、個人名では予期されないパターンを示します。例えば、連続する 3 つの同一の子音で構成されるストリング (「DDD」など) は、個人名では非常にまれですが、組織名では珍しいことではありません。

IBM InfoSphere Global Name Management コンポーネントが名前をカテゴリ化するとき、名前が 1 つ以上の名前カテゴリ理由コードと一致した場合、組織名と想定されます。それ以外の場合、個人名の候補となります。

IBM NameWorks を使用して名前を個人または非個人にカテゴリ化する

名前の解析、分類、分析、または検索の前のプリプロセッシング・ステップとして、データ・リスト上の名前を個人名か非個人 (組織) 名のどちらかにカテゴリ化できます。

このタスクについて

名前を個人か非個人のどちらかにカテゴリ化するには、IBM NameWorks の Scoring クラスの `categorize()` メソッドを使用し、カテゴリ化する名前をメソッドに渡します。

タスクの結果

`categorize()` メソッドは名前を調べて、選択された名前カテゴリ、名前カテゴリの理由コード (そのカテゴリが選択された理由の識別)、および 0 から 100 までの信頼性スコア (名前が正しくカテゴリ化された可能性のパーセンテージ) を戻します。パーセンテージが 100 に近いほど、名前カテゴリ化に対する信頼性が高くなります。

次のタスク

名前がカテゴリ化された後、IBM NameWorks で個人名と識別された名前を使用して、解析、分類、分析、または検索を行えます。

名前カテゴリの理由コード

名前カテゴリ化の理由によって、検出された非個人の標識またはパターンのタイプが示されます。これらは、そのカテゴリが選択された理由を説明します。理由コードを使用してさらに詳細な分析を行うこともできますし、これらの理由コードに基づいて、名前のカテゴリ化を独自に設定することもできます。

表 9. IBM InfoSphere Global Name Management 製品の名前カテゴリ理由コードとそれらの説明

名前カテゴリの理由コード	説明
BadData	名前が長すぎます。
UrlEnding	インターネット URL を示す共通インディケータ (「.COM」、「.ORG」、「.NET」など) が名前に含まれています。
EstateOf	名前に単語「 <i>estate of</i> 」が含まれています。
KnownOrg	名前に既知の組織句が含まれています。
Phrase	名前に組織のみの句が含まれています。
NoTokens	既知の名前トークンはありません。
AndCompany	名前に何らかの形式の「& Company」標識が含まれていません。
MultipleInitials	名前に複数のイニシャルが含まれています。
SingleSequence	連続した単一文字が名前に含まれています。
NameAndName	名前に「名前 & 名前」が含まれています。
LeadingToken	トークンが名前の先頭にしか出現しません。
Triplet	名前に先頭単一文字トリプレットが含まれています。
NforAnd	名前に「 <i>n for and</i> 」が含まれています。
SingleHyphen	名前に単一文字間のハイフンが含まれています。
MultipleHyphen	名前に複数のハイフンが含まれています。
MultiSlash	名前に複数のスラッシュが含まれています。
Enumeration	名前に列挙が含まれています (「1st」、「2nd」など)。
Possessive	名前に所有格が含まれています (「 <i>Smith's</i> 」、「 <i>Jones'</i> 」など)。
OrgWord	名前に、組織のみの単語で既知のものが含まれています。
HyphOrgWord	名前に、組織のみのハイフン付き単語で既知のものが含まれています。
AllSymbols	トークンに記号しか含まれません。
ConsPlusC	名前が、全部子音のトークンにタイプ C の単語を加えたもので構成されています。
CPL	名前に、タイプ C の単語、前置詞、および場所が含まれています。
TwoTypeC	名前に 2 つのタイプ C の単語が含まれています。
LandC	名前に場所とタイプ C の単語が含まれています。
AandC	名前に形容詞とタイプ C の単語が含まれています。
TandC	名前にタイプ T の単語とタイプ C の単語が含まれていません。
TwoTypeT	名前に 2 つのタイプ T の単語が含まれています。
LandT	名前にタイプ T の単語と場所が含まれています。
TwoPreps	名前に 2 つの前置詞が含まれています。
PrepL	名前に前置詞と場所が含まれています。
PrepT	名前に前置詞とタイプ T の単語が含まれています。

表 9. IBM InfoSphere Global Name Management 製品の名前カテゴリ理由コードとそれらの説明 (続き)

名前カテゴリの理由コード	説明
PrepC	名前に前置詞とタイプ C の単語が含まれています。
Parens	名前に括弧が含まれています。
NonAlpha	名前トークンにアルファベット以外の文字が含まれていません。
AllCons	名前トークンに含まれる文字がすべて子音です。
Name1andName2	名前に「名前 1」 & 「名前 2」のパターンが含まれています。
AndName	名前に「... & 名前」のパターンが含まれています。
LLC	名前に職業修飾子と「LLC」が含まれています。
ProfQual	名前に職業修飾子 (「Architect (建築士)」など) が含まれています。
FallThru	名前はすべてのテストで失敗しました。

IBM NameWorks を使用した 2 つの名前の比較

IBM NameWorks を使用して、2 つの名前が互いにどの程度類似しているかを判別できます。

このタスクについて

IBM NameWorks の Scoring クラスの `compare()` メソッドを使用し、2 つの Name オブジェクトをメソッドに渡すことによって、2 つの名前を比較します。それぞれの Name オブジェクトは、比較する名前の単一の名前ストリング (個人または組織) か、名と姓の入った 2 つのストリングのどちらかを含んでいます。

`compare()` メソッドは、2 つの名前を音訳し、照会名の `NameCategory` に基づいてそれらを比較します。比較に使用される比較パラメーター (`CompParms`) は、以下のように 2 つのオペランドの `NameCategory` に基づきます。

- 両方の名前とも個人名: 最初 (左) の名前に関連付けられた `CompParms` が適用されます。
- 両方の名前とも組織: 最初 (左) の名前に関連付けられた `CompParms` が適用されます。
- 一方の名前が個人でもう一方が組織: 名前は組織名として比較され、組織名の `CompParms` が適用されます。
- 一方の名前が `NameCategory` に関連付けられていない: `NameSifter` を使用して `NameCategory` が判別されます。使用される `CompParms` は、選ばれた `NameCategory` に基づきます。

`NameCategory` が個人の場合は、名前が解析されて分類されます。個人名の各フィールドは分類され、照会名の国/地域別情報コードは単一のロールアップ国/地域別情報コードに省略されて、比較のための正しい組み込みパラメーターを選択するのに使用されます。

タスクの結果

CompareData オブジェクトは、2 つの名前について 0 から 100 までの範囲の比較スコアを返します。0 は類似性なしを表し、100 は完全類似性を表します。

IBM NameWorks を使用した 2 つの日付の比較

多くの ID 検索タスクや ID 検査タスクは、キーとなる 2 つの情報、つまり名前と生年月日 (DOB) を基にしています。IBM NameWorks は、この 2 つのタイプのデータのそれぞれについて比較をサポートすることによって、多くの検索タスクが完全に IBM NameWorks の機能に基づいて行えるようになっています。

始める前に

日付は、YYYYMMDD の形式の数字で構成される 8 文字のストリングでなければなりません。

このタスクについて

2 つの日付を比較するには、IBM NameWorks の Scoring クラスの dateCompare() メソッドを使用し、比較する 2 つの日付を内容とするストリング対をメソッドに渡します。

タスクの結果

dateCompare() メソッドは 2 つの日付の日付比較を行い、0 から 100 までの範囲の類似性スコアを返します。0 が最小類似、100 が完全類似です。日付引数が必須フォーマット (YYYYMMDD) で渡されなかった場合、メソッドは -1 値を返します。

IBM NameWorks を使用して 2 つの日付値の差を判別する

IBM NameWorks を使用して、2 つの日付値の差を計算できます。

始める前に

日付値は、YYYYMMDD の形式の数字で構成される 8 文字のストリングでなければなりません。

このタスクについて

2 つの日付値の差を判別するには、IBM NameWorks の Scoring クラスの dateDifference() メソッドを使用し、2 つの日付値を内容とするストリング対をメソッドに渡します。

タスクの結果

dateDifference() メソッドは、2 つの日付値の差を計算して返します。日付引数が必須フォーマット (YYYYMMDD) で渡されなかった場合、メソッドは -1 値を返します。

データ・リストでの名前の検索

名前の検索は、IBM NameWorks と共に使用するようシステム管理者が IBM NameWorks 構成ファイルに構成するデータ・リストに対して行います。検索要求で 1 つ以上のデータ・リストの名前を指定すると、システムは、データ・リストに対して構成された検索のタイプ (*Type*= 設定。氏名検索か固有の名前の検索のどちらか) に基づいて、一致する名前データを戻します。複数のデータ・リストからの結果が、単一の応答にまとめられます。

始める前に

- 検索対象のデータ・リストの名前が分かっているなければなりません。(必要であれば、`getDatalistNames()` メソッドを使用して、既存のすべてのデータ・リスト名のリストを戻します。)
- 検索の一部として使用する検索方針の名前が分かっているなければなりません。(必要であれば、`getSearchStrategyNames()` メソッドを使用して、現在使用可能な検索方針をリストします。)
- `analyzeForSearch()` メソッドを使用することによって、検索のための個人名を準備をします。このメソッドは、名前を名フィールドと姓フィールドに解析し、名前フィールドごとに単一の国/地域別情報を戻します。このメソッドは、検索の一部として使用される `QueryName` オブジェクトの値も戻します。

このタスクについて

データ・リスト上の名前を検索するには、IBM NameWorks の `Scoring` クラスの `search()` メソッドを使用し、以下の値をメソッドに渡します。

- 特定の検索要求を識別するトランザクション ID 値。(値 `-1` を渡した場合、IBM NameWorks は固有値を割り当て、その値は検索結果で戻されます。この値はログ・ファイルの項目に含められるので、ログ・ファイルでこの特定の検索操作を参照するのに使用できます。)
- `QueryName` オブジェクトからの戻り値。この戻り値には、検索条件として使用される名前フィールドと国/地域別情報が含まれます。
- 検索するデータ・リストの名前。
- 使用する検索方針の名前。
- 検索するカテゴリーを表す整数。 `-1` の値は、検索方針のデフォルト値を使用する必要があることを示します。
 - 1 個人名
 - 2 組織名
 - 3 個人名と組織名の両方
- 返す応答 (つまり一致結果) の最大数を表す整数。この数は、返される一致結果の数を制限し、上位にランクされた一致結果をフィルタリングします。 `0` から `1` の値はすべての一致結果を戻し、 `-1` の値は検索方針のデフォルト値を使用する必要があることを示します。
- データ・リスト上の名前が一致結果として戻されるために合致するか超えなければならない最小スコア (`nmScore` 値) を表す `0` から `100` までの整数。ゼロより小さい値は `0` として扱われます。値が `100` より大きい場合は、無効パラメータ例外 (`GODW031E`) が発生します。

注: 検索する名前が一般的な名前である場合は、戻す応答の数を制限し、最小スコア値を大きくすることを検討してください。そうしないと、戻り結果用のメッセージ・バッファの容量を超えてしまう可能性があります。

タスクの結果

`search()` メソッドは、データ・リストに対して構成された検索のタイプを実行し、氏名スコアでソートされた一致結果のリストを返します。

次のタスク

固有の名前の検索を戻すようにデータ・リストが構成されている場合は、`dataFetch()` メソッドを使用して、固有の名前に関連付けられたすべての名前レコードの補足データを取得することができます。

固有の名前に関連付けられた名前の補足データの取得

固有の名前の検索用に構成されたデータ・リストに対して検索を行った後、戻された固有の名前の総数に含まれる名前に関連付けられた詳細情報を調べることが必要になる場合もあります。固有の名前に関連付けられたすべての名前レコードの補足データを取得するには、`dataFetch()` メソッドを使用します。

始める前に

- `search()` メソッドを使用して、固有の名前の検索を実行済みでなければなりません。
- 固有の名前が含まれるデータ・リストの名前が分かっている必要があります。(この情報は、検索操作によって戻されます。)

このタスクについて

データ・リスト上の名前を検索するには、IBM NameWorks の Scoring クラスの `search()` メソッドを使用し、以下の値をメソッドに渡します。

- 固有の名前が含まれるデータ・リストの名前。
- 特定の固有の名前を識別するためのキー値 (この値は検索操作によって戻されません)。

タスクの結果

`dataFetch()` メソッドは、キー値に基づいて、固有の名前に関連付けられたすべての名前レコードの補足データを取得します。

NameHunter を使用した名前の検索

NameHunter コンポーネントで、組織のアプリケーションにおける名前検索を向上させることができます。NameHunter は、「X に最も近い名前を 10 個検索する」、「類似性スコア 90 % 以上で Y と一致する名前を検索する」、「名前 A と名前 B の間の類似性の度合いを求める」といった要求に対応できます。

NameHunter の概要

NameHunter[®] プログラミング・ライブラリーには、開発者が新しいアプリケーションや既存のアプリケーションに対して、拡張された個人名および組織名の検索機能を追加できるようにするための関数およびクラスが含まれています。

NameHunter API を使用すると、「名前リストから JAMES SLESINGER に最も近い名前を 10 挙げる」、「データベースの JOHN WONG に対する類似性が 90% 以上の名前をすべて示す」、「PAUL VANESANN と P. VANLESANN の間の類似度を示す」などのユーザー要求をアプリケーションでサポートできます。

NameHunter は、統合された言語的、確率的、およびストリング相似性の技法を使用して、編集距離などの標準的なストリング相似性の測定基準、または標準的な Soundex または NYSIIS などの名前グループ・アルゴリズムが提供するものを超えた検索結果を提供できます。

NameHunter ライブラリーは、標準の C++ でコーディングされており、C++ で作成されたアプリケーションに統合できます。そのため、NameHunter ライブラリーは、C++ コンパイラーをサポートする任意のプラットフォームで使用できます。NameHunter は、シンプルさ、統合の容易さ、実行時の最大の柔軟性と拡張性を実現するように設計されました。

注: NameHunter では、1 つの名前フィールドについて最大 6 個のトークンを処理できます。最初の 6 個のトークンより後にあるトークンは、すべて無視されます。

国/地域別情報固有および構成可能な NameHunter の検索

NameHunter が実行する各個別の検索は、さまざまなランタイム比較パラメーターを調整することによって構成可能です。

それぞれの検索パラメーターは、2 つの名前の類似性の度合いを判別しつつ、その名前の比較の特定の面を制御します。基本的なパラメーターが、2 つの名前が一致していると見なされるためにどれくらい近接している必要があるのかを決定するしきい値を設定します。他のパラメーターは、特定の国/地域別情報に適用される、特殊な言語規則を制御します。これらのさまざまな検索制御パラメーターの最適な設定は、以下の要因によって決定されます。

- 検索する名前データの性質の注意深い分析
- ユーザーによって行われる照会のタイプ
- 精度および、組織のリコール許容度を定義するビジネス・ルール

実際には、開発者またはエンド・ユーザーは、各ランタイム検索制御パラメーターに対して IBM が確立したデフォルト値から開始して、その後必要に応じて調整できます。国/地域別情報固有の比較パラメーターは、CompParms データ構造にカプセル化されます。

NameHunter API 製品には、特定の事前定義されたパラメーターのパッケージが含まれており、それぞれが特定の国/地域または民族からの名前に対して有効な検索を行うよう調整されています。例えば、ヒスパニック系の名前には、一定の特性 (例えば、TORRES DE LA CRUZ などの複合の姓) が見られることが多いため、標準的な検索方式で処理するとたいがい問題が発生します。ヒスパニック・パラメーターの比較パラメーターには、このタイプの名前に固有の問題に対応する設定が含ま

れています。ユーザーはカスタムの国/地域別情報の新しい比較パラメーターを作成できます。また、検索を調整するために既存のパラメーターを変更できます。

称号、接辞、および修飾子 (TAQ) のデータ

IBM は、名前の突き合わせで使用される多文化の称号、接辞、および修飾子 (TAQ) のリストを提供しています。TAQ は、検索パラメーターによって左右される名前スコア全体で役立つため、名前の比較を行うときに廃棄されません。

構成ファイルを使用して TAQ リスト (taq.ibm) をロードするかどうかを選択できますが、デフォルトでは TAQ リストをロードします。また、IBM のリストを置き換える、または補足するために、カスタムの名および姓の変形リストを指定することもできます。TAQ および変形リストは、以下のパラメーターを含む、コンマで区切られたテキスト・ファイルである必要があります。

TAQ ファイル:

TAQ ファイルには、TAQ のスコアを決定するのに必要とされる情報がすべて含まれます。

デフォルトのリストを置き換える、または補足するために、独自の TAQ ファイルを指定できます。また、デフォルトの TAQ ファイルの情報をオーバーライドすることもできます。

ファイルの最初の項目は、ファイルが適切な形式になっているかどうかを判別するために NameHunter が使用するファイルのバージョンを示します。ヘッダーが欠落している場合、NameHunter はファイルが 4.2 より前のバージョン用であり、単一トークンの TAQ のみが含まれ、変形、係数、または複数のトークンのテキストの TAQ は含まれていないものと見なします。

4.2 より前のバージョンからマイグレーションしている場合、ConvertTaq 変換ユーティリティを実行する必要があります。

この製品のバージョン 4.2 では、TAQ での以下の使用が許可されています。

- 数字 (0-9)
- 特殊文字: @ \$ % & + = /
- ハイフン、ピリオド、コンマ、タブ、および改行は、スペースと同様に処理されます。

このファイルは、.ini ヘッダーを使用してセクションを分離します。

[version]

ファイルの最初のセクションは、ファイルが適切な形式になっているかどうかを判別するために NameHunter が使用するファイルのバージョンを示します。ヘッダーが欠落している場合、NameHunter はファイルが 4.2 より前のバージョン用であり、単一トークンの TAQ のみが含まれ、変形、係数、または複数トークンの用語の TAQ は含まれていないものと見なします。

4.2 より前のバージョンからマイグレーションしている場合、ConvertTaq 変換ユーティリティを使用して既存のファイルをアップグレードできます。

[taqs] ファイルのこのセクションは、次の形式を使用して TAQ を特定します。

TaqText, TaqType, TaqPosition, Culture

TaqText

トークンのテキスト。

TaqType

TAQ トークンのタイプ。次の TAQ タイプがサポートされます。

3 PREFIX

後続の名前の語幹トークンと同じ名前句に含まれているトークン。『DE』 および 『LA』 は接頭部です。

4 SUFFIX

先行する名前の語幹トークンと同じ名前句に含まれているトークン。『ALDEEN』 は接尾部です (アラビア語の名前)。

5 TITLE

名前とともに伝達されるトークンで、通常、社会的または職業的な立場を示します。『MR』 および 『GEN』 は称号です。称号は、名フィールドまたは姓フィールドのいずれにも含まれず、独自のフィールドに配置されます。

6 QUALIFIER

名前とともに伝達されるトークンで、通常は、世代関係、または社会的/職業的な地位を示します。『JR』 および 『ESQ』 は修飾子です。修飾子は、名フィールドまたは姓フィールドのいずれにも含まれず、独自のフィールドに配置されます。

9 ORGANIZATION DESIGNATOR

組織のタイプを特定する、『Inc.』、『LLC』、または『Limited Liability Corporation』などの単語。

10 PROFESSIONAL QUALIFIER

名前の職業的な特性を記述する『M.D.』や『CPA』などの単語。

11 STOPWORD

計算に影響を与えない単語。例えば、『The』は計算に影響を与えないため、『Pet Shop』の検索はデータ項目『The Pet Shop』と一致します。

12 ORGANIZATION AFFIX

名前に説明的な意味を追加しない単語。ただし、STOPWORD トークンとは異なり、ORGANIZATION AFFIX トークンは名前の計算の要素に入られます。

TaqPosition

名前の中での TAQ の位置。通常、称号は最初、修飾子は最後、接

頭部と接尾部は任意の位置になります。例えば、『Junior』は名前の最後の位置にあるときにのみ、修飾子となります。

- E 任意
- F 最初
- L 最後

Culture

国/地域別情報コードのコロンで区切られたリスト。この TAQ に関連付けられた国/地域、または民族的なグループを示します。単一の個人用の TAQ 項目に対して複数の国/地域別情報を指定できます。TAQ をすべての国/地域別情報に適用する場合は、**Culture** コードとして A を指定します。この値は、NameHunter および NameClassifier によって現在サポートされている国/地域別情報の 1 つである必要があります。

- 0 未確定
- 1 英語
- 2 アラビア語
- 3 中国語
- 4 ヒスパニック
- 5 韓国語
- 6 ロシア語
- 7 フランス語
- 8 ドイツ語
- 9 タイ語
- 10 インドネシア語
- 11 ヨルバ語
- 12 ペルシア語
- 13 パキスタン語
- 14 インド
- 15 日本語
- 16 アフガニスタン語
- 17 ベトナム語
- 18 ポーランド語
- 19 ポルトガル語
- 20 トルコ語
- 38 南西アジア
- 39 ヨーロッパ
- 40 ハン
- A すべて

[variants]

このセクションは、次の形式を使用して TAQ の変形を特定します。

score,variant:variant,culture:culture

score 変形が一致したときに割り当てられるスコア。スコアは 0.0 から 1.0 の間である必要があります。

variant(s)

関連する TAQ の変形のコロンで区切られたリスト。例えば、AMBASSADOR:AMB です。

culture(s)

国/地域別情報コードのコロンで区切られたリスト。単一の TAQ の変形項目に対して複数の国/地域別情報を指定できます。この値は、NameHunter および NameClassifier によって現在サポートされている国/地域別情報の 1 つである必要があります。

- | | |
|----|----------|
| 0 | 未確定 |
| 1 | 英語 |
| 2 | アラビア語 |
| 3 | 中国語 |
| 4 | ヒスパニック |
| 5 | 韓国語 |
| 6 | ロシア語 |
| 7 | フランス語 |
| 8 | ドイツ語 |
| 9 | タイ語 |
| 10 | インドネシア語 |
| 11 | ヨルバ語 |
| 12 | ペルシア語 |
| 13 | パキスタン語 |
| 14 | インド |
| 15 | 日本語 |
| 16 | アフガニスタン語 |
| 17 | ベトナム語 |
| 18 | ポーランド語 |
| 19 | ポルトガル語 |
| 20 | トルコ語 |
| 38 | 南西アジア |
| 39 | ヨーロッパ |
| 40 | ハン |
| A | すべて |

次の例は、IBM InfoSphere Global Name Management 製品が提供する TAQ ファイルの小さなサブセットを示したもので、独自の TAQ ファイルを作成する場合に役立ちます。

```
[version]
gnr 4.2
```

```
[taqs]
#PERSONAL TAQS
#format - TaqText,TaqType,TaqPosition,Culture1,Culture2,CultureN
#TaqPosition Typically, Titles are F, Qualifiers are L, Prefixes and
#Suffixes are E
ABD,3,E,0:1:18
MAYOR,5,F,10
LETNAN COLONEL,5,F,10

#ORGANIZATION TAQS
#format - TaqText,TaqType,TaqPosition,Culture(always 0)
#Unambiguous TAQ TaqPosition values are code E (for every position);ambiguous
#taqs are restricted to L (for last position)
&,12,E,0
A B,9,L,0
A PROFESSIONAL CORPORATION,9,L,0
```

```
[variants]
#PERSONAL TAQ VARIANTS
#format -#Factor,Variant1:Variant2:VariantN,Culture1:Culture2:CultureN
1.0,AMBASSADOR:AMB,0:1:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17
1.0,SR:SENIOR,0:1:2:3:5:6:7:8:9:10:11:12:13:14:15:16:17
1.0,DELAS:DE LAS,4:7
```

```
#ORGANIZATION TAQ VARIANTS
#format - Variant1,Variant2,Factor,Culture
COMPANIA,COMPANHIA,.99,0
COMPANY,COMPANIES,.99,0
```

```
#ORGANIZATION TAQ VARIANTS
#format -#Factor,Variant1:Variant2:VariantN,Culture1:Culture2:CultureN
.99,COMPANY:COMPANIES,0
.99,COMPANY:COMPAGNIE:COMPANIA:COMPANHIA,0
```

```
[taqFactors]
#format - TaqType,FactorType,Factor,Culture
3,1,0.97,A
3,2,0.98,A
```

```
#TaqType
# Prefix = 3
# Suffix = 4
# Title = 5
# Qualifier = 6
# OrganizationDesignator = 9
# ProfessionalQualifier = 10
# StopWord = 11
# OrganizationAffix = 12
```

```
#FactorType
# Different = 1
# Missing = 2
```

```
#Culture Codes
# Ambiguous = 0
# Anglo = 1
# Arabic = 2
# Chinese = 3
# Hispanic = 4
# Korean = 5
```

```

# Russian      = 6
# French       = 7
# German       = 8
# Thai         = 9
# Indonesian   = 10
# Yoruban     = 11
# Farsi       = 12
# Pakistani   = 13
# Indian      = 14
# Japanese    = 15
# Afghan      = 16
# Vietnamese = 17
# Polish      = 18
# Portuguese  = 19
# Turkish     = 20
# SouthwestAsian = 38
# European   = 39
# Han        = 40
# All        = A

```

converttaq.exe 変換ユーティリティ:

converttaq.exe コマンド行ユーティリティは、4.2 より前の TAQ ファイルを変換します。4.2 より前のバージョンからマイグレーションしている場合、既存の TAQ ファイルで **converttaq.exe** 変換ユーティリティを実行する必要があります。

```
converttaq <input_pre_4.2_TAQ_filename> <output_converted_4.2_TAQ_filename>
```

変換プロセス中、新しい **TaqPosition** パラメーター値がデフォルトの値 E (すべての位置) に設定されます。

名前トークンの変形

名前トークンの変形は、指定された名前の代わりに名前で、その名前と同じと見なされるものの、特定の外形が異なります。変形は通常、スペルのバリエーションから生じます。NameHunter は、それらのファイルをロードするための関数を提供します。

IBM は、姓および名に対して、名前トークンの変形 (例: Peggy = Margaret) の大量のリストを提供しています。変形はデータ・ディレクトリーにあり、gmv.ibm (名の変形)、snv.ibm (姓の変形)、および onv.ibm (組織の変形) という名称です。

IBM のリストを置き換える、または補足するために、独自の名および姓の変形を指定できます。また、既存の変形ファイルの情報をオーバーライドして、個別の変形のペアを抑制したり、変形のペアに割り当てられたスコアを変更したりできます。例えば、次の変形の項目について考えます。

名の変形

```
0.95,ROBERT:ROB:ROBBY:BOB:BOBBY:BERT,0:1
```

この項目は、リストされた名の変形を、国/地域別情報の 0 および 1 (汎用および英語) に対する関連スコア 0.95 に関連付けます。

姓の変形

```
0.95,OYANG:OYEUNG:OYONG:OYOUNG:AU YAN:AU YANG:AU YEUNG:AU YONG:AU YOUNG:
AW YAN:AW YANG:AW YEUNG:AW YONG:AW YOUNG:OU YAN:OU YANG:OU YEUNG:OU YONG:
OU YOUNG:OW YAN:OW YANG:OW,3
```

この項目は、リストされた姓の変形を、国/地域別情報の 3 (中国語) に対する関連スコア 0.95 に関連付けます。

組織名の変形

1,WRECKER:WRCKR,0

この項目は、リストされた組織名の変形を、国/地域別情報の 0 (汎用) に対する関連スコア 1 に関連付けます。

名前変形ファイル:

名前変形ファイルには、代替りの名前の変形および同等の名前の変形を見つけるのに必要とされる情報がすべて含まれます。

デフォルトのリストを置き換える、または補足するために、独自の変形ファイルを指定できます。また、デフォルトの変形ファイルの情報をオーバーライドすることもできます。

4.2 より前のバージョンからマイグレーションしている場合、**convertvar.exe** 変換ユーティリティを実行する必要があります。

このファイルは、.ini ヘッダーを使用してセクションを分離します。

[version]

ファイルの最初のセクションは、ファイルが適切な形式になっているかどうかを判別するために NameHunter が使用するファイルのバージョンを示します。ヘッダーが欠落している場合、NameHunter はファイルが 4.2 より前のバージョン用であり、単一トークンのテキストのみが含まれていると見なしします。

4.2 より前のバージョンからマイグレーションしている場合、**convertvar** 変換ユーティリティを使用してファイルを更新できます。

[variants]

このセクションは、次の形式を使用して変形を特定します。

score,variant:variant,culture:culture

score 変形が一致したときに割り当てられるスコア。スコアは 0.0 から 1.0 の間である必要があります。

variant(s)

関連する名前トークンの変形のコロンで区切られたリスト (例: BOB:ROBERT)。

culture(s)

国/地域別情報コードのコロンで区切られたリスト。単一の変形項目に対して複数の国/地域別情報を指定できます。この値は、NameHunter および NameClassifier によって現在サポートされている国/地域別情報の 1 つである必要があります。

- 0 汎用
- 1 英語
- 2 アラビア語
- 3 中国語

- 4 ヒスパニック
- 5 韓国語
- 6 ロシア語
- 7 フランス語
- 8 ドイツ語
- 9 タイ語
- 10 インドネシア語
- 11 ヨルバ語
- 12 ペルシア語
- 13 パキスタン語
- 14 インド
- 15 日本語
- 16 アフガニスタン語
- 17 ベトナム語
- 18 ポーランド語
- 19 ポルトガル語
- 20 トルコ語
- 38 南西アジア
- 39 ヨーロッパ
- 40 ハン
- A すべて

次の例は、独自の名前変形ファイルを構成する場合に役立つ形式を示しています。

```
[version]
gnr 4.2.1
```

```
[variants]
0.95,ROBERT:ROB:ROBBY:BOB:BOBBY:BERT,0:1
0.95,ELIZABETH:ELIZBETH:LIZ:LIZZY:LIZZIE:LISA:LIBBY:LIBBIE,0:1
0.95,ELISABETH:LIESEL:LIESE:LIESCHEN:LIL,8
```

convertvar.exe 変換ユーティリティ:

convertvar.exe コマンド行ユーティリティは、4.2 より前の変形ファイルを変換します。4.2 より前のバージョンからマイグレーションしている場合、既存の名前変形ファイルで **convertvar.exe** 変換ユーティリティを実行する必要があります。

```
convertvar <input_pre4.2_filename> <output_converted4.2_filename>
```

用語

用語 とは、1 つの概念を示す名前トークンです。通常は組織名に使用されます。

用語テキストは、単一トークンまたは複数トークンのいずれかになります。通常、用語は複数トークン・テキストで構成されます。複数トークン用語は、2 つ以上の

語のシーケンス (空白文字を埋め込んだテキスト) であり、そのシーケンスで 1 つの概念を示します。用語ファイルでは、検索比較で使用する一連の複数トークン用語を定義します。NameHunter では、用語ファイルの各項目が 1 つの非スペース区切り語と同じように取り扱われます。例:

複数トークン用語

```
TRACTOR TRAILER,1.0,0
```

"tractor trailer" は、"tractor_trailer" と同じように取り扱われます。

用語ファイル:

用語ファイルには、組織用語を識別するために必要な情報がすべて入っています。

IBM では、デフォルトの組織用語ファイルを用意しています。組織用語は、データ・ディレクトリーにあり、`onterms.ibm` という名前になっています。独自の用語ファイルを用意して、デフォルトのリストを置き換えたり補足したりすることもできます。デフォルトの用語ファイルの情報をオーバーライドすることも可能です。

用語ファイルの項目には、変形を組み込みません。変形のある複数トークン用語は、組織変形ファイルで定義します。例えば、`real estate` という用語に `rl est` という変形があるとします。用語ファイルではそれぞれの値を定義し、組織変形ファイルではそれぞれの変形を定義します。

このファイルは、`.ini` ヘッダーを使用してセクションを分離します。

[version]

ファイルの最初のセクションでは、ファイルが適切な形式になっているかどうかを確認するために NameHunter が使用するファイルのバージョンを示します。

[terms]

このセクションでは、以下の形式を使用して用語を指定します。

```
termtext,weight,culture:culture
```

termtext

用語として取り扱うテキスト。

用語テキストは、単一トークンまたは複数トークンのいずれかになります。通常、用語は複数トークン・テキストで構成されます。複数トークン用語は、2 つ以上の語のシーケンス (空白文字を埋め込んだテキスト) であり、そのシーケンスで 1 つの概念を示します。

weight

用語が一致スコアに寄与する相対的な貢献度を指定します。

この値は、1.0 に設定する必要があります。

culture(s)

この用語に関連する国/地域別情報または民族グループを示す、コロンで区切られた国/地域別情報コードのリスト。この値は、NameHunter および NameClassifier によってサポートされている国/地域別情報の 1 つである必要があります。

0 汎用

- 1 英語
- 2 アラビア語
- 3 中国語
- 4 ヒスパニック
- 5 韓国語
- 6 ロシア語
- 7 フランス語
- 8 ドイツ語
- 9 タイ語
- 10 インドネシア語
- 11 ヨルバ語
- 12 ペルシア語
- 13 パキスタン語
- 14 インド
- 15 日本語
- 16 アフガニスタン語
- 17 ベトナム語
- 18 ポーランド語
- 19 ポルトガル語
- 20 トルコ語
- 38 南西アジア
- 39 ヨーロッパ
- 40 ハン
- A すべて

独自の用語ファイルを構成するときに役立つ形式を以下の例で示します。

```
[version]
gnr 4.2.1

[terms]
ANIMAL CLINIC,1.0,0
APPLIANCE REPAIR,1.0,0
AUTO SALES,1.0,0
AUTO REPAIR,1.0,0
AUTO SALES,1.0,0
BAY AREA,1.0,0
BEAUTY SALON,1.0,0
CHILD CARE,1.0,0
COMMUNITY CHURCH,1.0,0
ELEMENTARY SCHOOL,1.0,0
FARM BUREAU,1.0,0
FIRE DEPT,1.0,0
HAIR CARE,1.0,0
HEATING & AIR,1.0,0
HEATING & AIR COND,1.0,0
```

LAWN CARE,1.0,0
LITTLE LEAGUE,1.0,0
NAIL CARE,1.0,0
REAL ESTATE,1.0,0
SECURITY SYSTEMS,1.0,0

名前の正規化

個人名の場合、名前の正規化は、名前トークンの音ベースのスペルを生成する機能です。これにより、発音が同一のさまざまなスペルを、同一または類似の形式にマップします。正規化されたこうしたスペルの効果として、名前のスペルがかなり異なっているにもかかわらず、関連があると一般的に認識されている名前を **NameHunter** で特定できます。

例えば、**Layton** と **Leighton** の両方の名前は、**Laten** に正規化されます。最大正規化名前スコアの比較パラメーター (**REGULARIZE_SCORE_MAX**) によってスコアが制限されますが、それらの名前が比較されると完全一致が返されます。

組織名の場合、正規化によって数字と記号を、完全なスペル形式にマップします。例えば英語では、規則により、**162**、*one hundred sixty-two*、*one sixty-two*、および *one six two* はすべて互いに一致しているとされます。

名前の正規化は、名前の正規化規則ファイルによって主導されます。以下の名前の正規化規則ファイルが提供されています。

angloRegRule.ibm

英語の国/地域別情報によって特定される個人名用。これは汎用の国/地域別情報のために使用されることもあります。

angloOnRegRule.ibm

英語の国/地域別情報によって特定される組織名用。

chineseRegRule.ibm

中国語の国/地域別情報によって特定される個人名用。

chineseNativeOnRegRule.ibm

漢字で書かれた数字や他のテキストに関する特殊な規則も含む、中国語の国/地域別情報によって特定される組織名用。

chineseOnRegRule.ibm

中国語の国/地域別情報によって特定される組織名用。

farsiRegRule.ibm

ペルシア語の国/地域別情報によって特定される個人名用。

frenchRegRule.ibm

フランス語の国/地域別情報によって特定される個人名用。

genericOnRegRule.ibm

未指定 (未確定) の国/地域別情報によって特定される組織名用。

germanRegRule.ibm

ドイツ語の国/地域別情報によって特定される個人名用。

hispanicRegRule.ibm

ヒスパニックの国/地域別情報によって特定される個人名用。

hispanicOnRegRule.ibm

ヒスパニックの国/地域別情報によって特定される組織名用。

indianRegRule.ibm

インドの国/地域別情報によって特定される個人名用。

indoRegRule.ibm

インドネシア語の国/地域別情報によって特定される個人名用。

japaneseRegRule.ibm

日本語の国/地域別情報によって特定される個人名用。

japaneseNativeOnRegRule.ibm

漢字で書かれた数字や他のテキストに関する特殊な規則も含む、日本語の国/地域別情報によって特定される組織名用。

koreanRegRule.ibm

韓国語の国/地域別情報によって特定される個人名用。

koreanOnRegRule.ibm

韓国語の国/地域別情報によって特定される組織名用。

polishRegRule.ibm

ポーランド語の国/地域別情報によって特定される個人名用。

polishOnRegRule.ibm

ポーランド語の国/地域別情報によって特定される組織名用。

portugueseRegRule.ibm

ポルトガル語の国/地域別情報によって特定される個人名用。

portugueseOnRegRule.ibm

ポルトガル語の国/地域別情報によって特定される組織名用。

russianRegRule.ibm

ロシア語の国/地域別情報によって特定される個人名用。

russianOnRegRule.ibm

ロシア語の国/地域別情報によって特定される組織名用。

swasianRegRule.ibm

西南アジアの国/地域別情報によって特定される個人名用。
arabicRegRule.ibm を置き換えます。

thaiRegRule.ibm

タイ語の国/地域別情報によって特定される個人名用。

turkishRegRule.ibm

トルコ語の国/地域別情報によって特定される個人名用。

カスタムの国/地域別情報を作成するお客様は、独自のカスタム正規化規則ファイルを追加することができます。ただし、カスタム正規化ファイルの開発とサポートは、Global Name Management 製品の一部としては提供されません。

正規化を使用可能にした場合、パフォーマンスが低下します。名前がデータ・リストに追加され、正規化が正規形を見つけると、2番目の項目がデータ・リストに追

加されます。また、照会名が正規化されると、両方の形式がデータ・リストの各名前と比較されます。この機能を使用可能にした場合、検索時間がほぼ 2 倍になります。

アプリケーションへの NameHunter API の統合

アプリケーションに NameHunter API を統合するには、API の完全な定義が含まれる NameHunter.h ヘッダー・ファイルを使用します。

インクルード・ディレクトリーには、ConfigHandler.h という別のヘッダー・ファイルがあります。ただし、このファイルは一部のサンプル・アプリケーションでのみ使用されます。このファイルは使用できますが、NameHunter を使用する場合に必須ではありません。

NameHunter の実装は、1 つのオブジェクト・ライブラリーに含まれます。ただし、NameHunter はいくつかの IBM 共有コンポーネントを使用して、名前の音訳および正規化をサポートします。このほとんどは、IBM の International Components for Unicode (ICU) からのものです。ライブラリーは、次のとおりです。

NameHunter.lib

NameHunter API

NameTransliterator.lib

名前の音訳および正規化に使用するライブラリー

sicudata.lib

ICU のデータ・テーブル

sicui18n.lib

ICU の国際化対応ライブラリー

sicuuc.lib

ICU の共通の Unicode ライブラリー

実際のファイル名は、プラットフォームごとに異なります。前述の名前は Windows で使用され、Unix プラットフォームでは「libNameHunter.a」などの名前になります。ご使用のプラットフォームの正確なファイル名を取得するには、bin ディレクトリーを参照してください。

他のデータへのリンク

NameHunter は、名前データを格納して検索します。ほとんどのシステムでは、名前以外にも他の種類のデータを格納して使用します。例えば、住所、口座番号、体格、画像などです。このため、NameHunter は SearchList クラスの各項目に ID フィールドを指定します。

SearchList にレコードを追加する場合、データベースの索引など、ID フィールドのデータに索引/ポインターを指定できます。NameHunter の ID フィールドは最長 256 バイトで、ASCII 文字を任意に組み合わせて構成できます。

NameHunter API クイック・スタートの例

これらのクイック・スタートの例では、NameHunter の基本的な機能を示すいくつかの小さな実行プログラムが提供されます。

NameHunter クイック・スタートの例: 2 つの名前の一致

NameHunter API を使用した実行プログラムの簡単な例を以下に示します。この例では、2 つの名前を比較し、それらが一致するかどうかを報告します。

NameHunter インスタンスをコンパイルして作成し、NameHunter::nameMatch 関数を呼び出します。変形などの NameHunter の強力な機能のいくつかが使用可能になっていないことに注意してください。

```
#include <NameHunter.h>
#include <iostream>

using namespace LAS;
using namespace LAS::NH;
using namespace std;

int main(int argc, char* argv[])
{
    // NameHunter will throw (mostly at startup)
    try
    {
        // you always need a NameHunter instance
        NameHunter nh;

        // call nameMatch using the default compParms
        if (nh.nameMatch("jack", "johnson jr",
                        "john j", "de la jones" ))
            cout << "names match" << endl;
        else
            cout << "names don't match" << endl;
    }
    catch (const exception& e)
    {
        cerr << "Caught exception - " << e.what() << endl;
        return 1;
    }

    return 0;
}
```

NameHunter クイック・スタートの例: 2 つの名前のスコア

NameHunter::nameScore 関数を使用した 2 つの名前の間の類似性について報告する単純な例を以下に示します。特殊な機能は使用されていません。

```
#include <NameHunter.h>
#include <iostream>

using namespace LAS;
using namespace LAS::NH;
using namespace std;

int main(int argc, char* argv[])
{
    // NameHunter will throw (mostly at startup)
    try
    {
        // you always need a NameHunter instance
        NameHunter nh;

        char* gn1 = "jack";
        char* sn1 = "johnson jr";
        char* gn2 = "john j";
        char* sn2 = "de la jones";

        // call nameScore with the default compParms
        ScoreInfo score = nh.nameScore(gn1, sn1, gn2, sn2);
    }
}
```

```

    cout << "comparing " << gn1 << " " << sn1
         << " to " << gn2 << " " << sn2 << endl;
    cout << "  name score = " << score.name << endl;
    cout << "  gn score = " << score.gn << endl;
    cout << "  sn score = " << score.sn << endl;
}
catch (const exception& e)
{
    cerr << "Caught exception - " << e.what() << endl;
    return 1;
}

return 0;
}

```

NameHunter クイック・スタートの例: 異なる国/地域別情報、TAQ、および変形を使用した 2 つの名前のスコア

より現実的な例を次に示します。NameHunter の使用可能な国/地域別情報をすべて使用して、2 つの名前のスコアを決定します。また、TAQ (称号、接辞、修飾子) および変形 (John = Jack) も使用します。

```

#include <NameHunter.h>
#include <iostream>

using namespace LAS;
using namespace LAS::NH;
using namespace std;

int main(int argc, char* argv[])
{
    // NameHunter will throw (mostly at startup)
    try
    {
        // you always need a NameHunter instance
        NameHunter nh;

        // assuming that these files are in the path.
        // If they are not, an exception will be thrown.
        nh.loadTaqs("taq.ibm");
        nh.loadVariants("gnv.ibm", GivenName);
        nh.loadVariants("snv.ibm", SurName);

        char* gn1 = "jack";
        char* sn1 = "johnson jr";
        char* gn2 = "john j";
        char* sn2 = "de la jones";

        for (int i = 0; i < MaxCultureNum; ++i)
        {
            CompParms gnParms((Culture)i, GivenName);
            CompParms snParms((Culture)i, SurName);
            ScoreInfo score = nh.nameScore(gn1, sn1,
                                           gn2, sn2,
                                           &gnParms, &snParms);
            cout << "  for culture, " << nh.cultureName((Culture)i) << endl;
            cout << "  name score = " << score.name << endl;
            cout << "  gn score = " << score.gn << endl;
            cout << "  sn score = " << score.sn << endl;
        }
    }
    catch (const exception& e)
    {
        cerr << "Caught exception - " << e.what() << endl;
        return 1;
    }
}

```

```

    }
    return 0;
}

```

NameHunter クイック・スタートの例: 検索リストおよび検索

典型的なケースでは、名前のデータベースを NameHunter SearchList オブジェクトに格納して、照会名を使用したリストの検索に NameHunter Searcher オブジェクトを使用します。SearchList、Searcher のパラダイムの単純な例を次に示します。この例では、国/地域別情報、比較パラメーターの調整、またはその他の NameHunter の特別な機能は使用していないことに注意してください。

```

#include <NameHunter.h>
#include <iostream>

using namespace LAS;
using namespace LAS::NH;
using namespace std;

int main(int argc, char* argv[])
{
    // NameHunter will throw (mostly at startup)
    try
    {
        // you always need a NameHunter instance
        NameHunter nh;

        // assuming that these files are in the path
        nh.loadTaq("taq.ibm");
        nh.loadVariants("gnv.ibm", GivenName);
        nh.loadVariants("snv.ibm", SurName);

        // create a search list and add some names
        SearchList searchList(&nh);

        // add entries in the format GN, SN, ID where ID is
        // a field you can use to tie to you own data on
        // this name (e.g., date of birth, favorite color, etc.).
        searchList.add("john j", "de la jonson", "1");
        searchList.add("jack", "john sr", "2");
        searchList.add("j", "johnson", "3");
        searchList.add("george", "smith", "4");

        // create a searcher and resultList
        Searcher searcher(&nh);
        ResultList results;

        char* gn = "jack";
        char* sn = "johnson jr.";

        searcher.search(searchList,
                        results,
                        gn,
                        sn);

        // print the query name and all the matches.
        cout << "hits for, " << sn << ", " << gn << endl;
        for (size_t i = 0; i < results.size(); ++i)
            cout << " "
                 << results[i].sn << ", "
                 << results[i].gn << " - "
                 << results[i].score.name << endl;
    }
    catch (const exception& e)
    {

```

```

        cerr << "Caught exception - " << e.what() << endl;
        return 1;
    }

    return 0;
}

```

NameHunter のサンプル・アプリケーション

NameHunter の配布パッケージに、API のさまざまな使用方法を例示するサンプル・アプリケーションが付属しています。これらのサンプル・アプリケーションは、独自のアプリケーションの開発を開始するための基礎として使用できます。

NameHunter のサンプル検索アプリケーション

NameHunter 検索アプリケーションは、名前の 2 つのファイルを比較して、結果を別のファイルに書き込むコマンド行プログラムです。このアプリケーションを実行するには、コマンド・プロンプトを開き、/support/bin ディレクトリーに移動し、search を入力します。

グローバル・パラメーターの設定を行い、名または姓の比較パラメーターを設定して、search.config 構成ファイル内に次のデータ・ファイルの場所を指定します。また、この構成ファイルを使用して、TAQ ファイルおよび変形ファイルをロードし、正規化および音訳を構成することもできます。次のデータ・ファイルは、search.config 構成ファイルが呼び出すコマンドで区切られたテキスト・ファイルです。

queryFile

照会する名前が含まれます。

nameFile

検索対象の名前が含まれます。

resultFile

一致した名前のレコードが含まれます。

search.config 構成ファイルは、/support/data ディレクトリーに配置され、search サンプル・アプリケーションが使用するいくつかの他のファイルも一緒に配置されています。

NameHunter の why サンプル・アプリケーション

NameHunter の why サンプル・アプリケーションは、NameHunter が 2 つの名前に対して特定のスコアを示す理由を判別するコマンド行ユーティリティーです。このアプリケーションを実行するには、コマンド・プロンプトを開き、/support/bin ディレクトリーに移動し、why を入力します。

why.config 構成ファイルを変更して、NameHunter CompParm の設定を変更し、比較する名前を指定できます。構成ファイルを変更したら、ファイルを保存して、コマンド・プロンプトでピリオド (.) を入力します。why ユーティリティーが構成ファイルを再ロードして、アプリケーションを再起動せずに名前を比較します。名前の比較の結果は、NameHunter のソース出力にプリントされます。

比較パラメーターの変更

比較パラメーター (*CompParms*) は、名前オブジェクトの比較の指針およびスコアリングに使用される調整可能なパラメーターのセットです。名前オブジェクトは、指定したデータ・リスト上の単一の候補名に対する照会名の処理時に参照されます。

NameHunter 検索結果の統合と最適化の任務を負う技術者は、これらのパラメーターの 1 つ以上を調整することによって因果関係を理解する必要があります。

CompParms 設定を詳細にわたって十分に理解することにより、特定のアプリケーションおよび名前と関連データの特定のデータ・リストに対して最大限の効率で機能するように NameHunter を構成できるようになります。

CompParms 設定値の確認と設定は、NameHunter Developers Tool Kit に用意されている関連 API グループを使用して行います。NameHunter 機能が

NameHunter Distributed Search アプリケーションを介してアクセスされる場合は、XML パラメーター・メッセージ内に特定の設定値を含めるか、IBM デフォルトを使用するための国/地域別情報コードを指定するかのどちらかの方法によって、CompParms 設定値を調整することができます。

ここには、パラメーター設定とサンプル検索結果の例が多数記載されています。これらの結果は、NameHunter Distributed Search に付属のサンプル・データを使用した実際の検索から引用したものです。一部の例では、結果リストが扱いやすいサイズに圧縮されています。

NameHunter 比較パラメーターの概要

NameHunter 比較パラメーター (*CompParms*) は、抽象データ構造体を形成します。この抽象データ構造体は、NameHunter ベースの検索のターゲットとして指定されたデータ・リスト (データベースまたはレコードのファイル) 上の一連の各名前と照会名のペアワイズ比較のための永続ランタイム制御フレームワークになります。

CompParms 検索制御は、個人名の基本データ・モデルを軸に編成されます。広範囲の言語と国/地域別情報を入手先として収集された名前について、2 部構成の名前のモデルが確立されました。基礎となるこの名前モデルがあるので、名と姓にさまざまな NameHunter CompParms を使用できます。

NameHunter API には、定義済み CompParm 設定のパッケージがいくつか組み込まれており、それらの設定は名前を効果的に処理できることが著名グループによって示されています。これらの国/地域別情報パラメーターのパッケージは CompParms の代替デフォルト値になり、特定の国/地域別情報バックグラウンドに明確には関連付けられない名前に関して使用される汎用パッケージも含まれます。

compParms.config ファイルには、1 つ以上の国/地域別情報のデフォルト

CompParms のオーバーライドが含まれます。この構成ファイルには、個人名の名と姓の国/地域別情報値、および組織名の値のセット 1 つが含まれます。

Distributed Search はこの構成ファイルを読み取り、国/地域別情報の設定をコンパイルし、NameHunter の `overrideDefaultParms()` メソッドを呼び出して新しいデフォルト値を設定します。現行 CompParms のデフォルト値は、`getDefaultParms()` メソッドを呼び出すことによって取得できます。

compparms.config ファイルを NameWorks が参照することもあります。その場合はインスタンス・レベルで適用され、Embedded Search とペアワイズ比較 (Scoring.compare()) の両方が影響を受けます。

スコアリングと評価のプロセスは、以下の抽象化連続レベルで行われます。

- 氏名レベル
- 名前フィールド・レベル (名または姓)
- セグメント・レベル (トークンまたは名前句)

このグループ化は NameHunter の機能を使用するアプリケーションのソフトウェア設計と保守を容易にすることを目的としています。階層化 CompParm アーキテクチャーによって、NameHunter 名前処理シーケンスへのランタイム管理アクセスやランタイム・ユーザー・アクセスが簡素化されます。その結果、NameHunter 対応アプリケーションは、ユーザー要求の変化と、検索される名前のデータベースの (定性的または定量的) 変化の両方に対して、柔軟かつ効果的に対処できます。

代替解析スコア係数:

ユーザーは、代替解析スコア係数 (altScoreFactor) を 1.0 未満の値に設定すると、名前の代替解析に基づく一致結果にペナルティーを科すことができます。

名前「JOHN ELTON」と名前「ELTON JOHN」は一致することがあります。これは、名前「ELTON JOHN」が「JOHN」に名としてマークを付け、「ELTON」に姓としてマークを付ける代替解析を生成するからです。ユーザーは名前のオリジナルの順序の違いを比較スコアに反映させ、この比較が完全一致のスコア 1.0 を受け取らないようにすることができます。代替スコア係数は、NameHunter によって計算される氏名スコアに適用されます。

代替スコア係数は 0.00 と 1.00 の間の値でなければなりません。これは、以下の 3 つの方法のいずれかで設定できます。

- 下位の NameHunter コンポーネント (ibmgnr::hunter::Searcher::altScoreFactor()) により
- NameWorks の General 設定として。これは特定の Scoring オブジェクトに関連付けられたすべての Embedded Search データ・リストのスコアに影響します (NameWorks 構成ファイルを参照してください)。

```
[General]
DefaultAltScoreFactor=0.99
```

- NameWorks 検索戦略の定義の一部として (NameWorks 検索戦略の定義を参照)。

```
[Strategy:Somename]
AltScoreFactor=0.99"
```

短い名前のスコアリング・ロジック:

短い名前のコントロールを使用するなら、名前の長さに関連して差異が大きくなる場合でも短い名前で望む結果が得られます。

IBM InfoSphere Global Name Management に、短い名前のマッチング用の特殊なロジックが組み込まれました。このロジックでは、タイプミスと思われるものが実際には正当な個人名なのかどうか考慮されます。例えば「BRET」および「BETR」

は、「BERT」と異なります。文字の入れ替わりが 1 つありますが、「BRET」は正当な名前と考えられるのに対し、「BETR」はタイプミスと考えられます。IBM InfoSphere Global Name Management は、名前データ・アーカイブ (NDA) からの頻度情報を利用することで、2 つの短い名前の差異が、1 文字多いか (例:

「STEVE」と「STEEVE」)、1 文字違いか (例: 「STEVE」と「STEBE」)、または入れ替わっている 1 組の文字 (例: 「STEVE」と「SETVE」) である場合に、正当な名前なのか、それともタイプミスなのかを判別します。それから、名前のスコアリングを処理する特殊なロジックが適用されます。スコアリング・ロジックは、比較パラメーターの適用により変更することができます。名前が長くなればなるほど、タイプミスに起因すると見なせる名前間のスコアリングの差異が、はっきりしくなります。そのため、新しいスコアリング・ロジックは短い名前にのみ適用されます。なお、短いと定義されるのは 2 文字から 7 文字までです。8 文字からなる名前は、「MARJORIE」と「MARORIE」などのように、より短い名前と関連がある場合はこのロジックを適用することができます。2 文字の名前は、2 または 3 文字の名前と比較される場合にのみ、この新しいロジックでスコアリングされます。その理由は、一文字のトークンはイニシャルと見なされ、特殊なイニシャル・スコアリング・ロジックが適用されるためです。タイプミスによる差異が複数含まれる名前にも、新しいロジックが適用されません。そのため、「MARJORIE」と「MAROREI」は、通常のスコアリング・ロジックを使用してスコアリングされません。

短い名前のコントロールは、NameHunter API または NameHunter Distributed Search XML 属性によって指定できます。

NameHunter API

- CompParms::shortMinLength
- CompParms::shortMaxLength
- CompParms::shortNameScore
- CompParms::shortValidFactor
- CompParms::shortValidRatio
- CompParms::shortValidMax

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN
SHORT_MIN_LENGTH
- COMP_PARMS_GN
SHORT_MAX_LENGTH
- COMP_PARMS_GN
SHORT_NAME_SCORE
- COMP_PARMS_GN
SHORT_VALID_FACTOR
- COMP_PARMS_GN
SHORT_VALID_RATIO
- COMP_PARMS_GN
SHORT_VALID_MAX
- COMP_PARMS_SN
SHORT_MIN_LENGTH
- COMP_PARMS_SN
SHORT_MAX_LENGTH
- COMP_PARMS_SN
SHORT_NAME_SCORE
- COMP_PARMS_SN
SHORT_VALID_FACTOR
- COMP_PARMS_SN
SHORT_VALID_RATIO
- COMP_PARMS_SN
SHORT_VALID_MAX

shortMinLength および **shortMaxLength** パラメーターで、短い名前のロジックが適用される名前の文字数を制御します。2 から 7 までの値を指定する必要があります。

shortNameScore パラメーターは、1 つのタイプミスによる差異がある、マッチングする 2 つの短い名前 (「ANNA」と「ANNQ」、「IVAN」と「IVANM」、「KLAUS」と「KLUAS」など) に対して使用するスコアを指定します。有効な値は [0.00, 1.00] (両端を含む) です。すべての国/地域別情報におけるデフォルトは 0.97 です。

shortValidFactor パラメーターは、単一の差異がある名前を別個の正当な名前と見なせる場合に (「DORA」と「CORR」、「JUAN」と「JUANA」、「AMIR」と「MAIR」など)、**shortNameScore** に適用される係数を指定します。そのような場合にマッチ・スコアを求めるために、**shortNameScore** にこの係数が乗算されます。有効な値は [0.00, 1.00] (両端を含む) です。すべての国/地域別情報におけるデフォルトは 0.75 です。(したがって、スペルの違いが 1 つある、2 つの別個の正当な名前間のマッチ・スコアは、0.72 です。)

NDA において一方の短い名前の頻度カウントが、もう一方における構成可能な比率よりも大きい場合、または両方の頻度カウントが、構成可能なしきい値よりも大きい場合、それらの短い名前は別個の正当な名前と見なされます。これらの 2 つの

条件は、**shortValidRatio** および **shortValidMax** パラメーターによってそれぞれ制御されます。例えば、0.1 に設定されたデフォルトの **shortValidRatio** の場合、「JAMAM」は「JAMAL」と比較されると、NDA での「JAMAM」の出現回数が「JAMAL」の出現回数の 0.01 倍よりも小さいため、タイプミスとしてスコアリングされます。一方、「KAMAL」が「JAMAL」と比較された場合、NDA での「KAMAL」の出現回数は有効な比率よりも大きいため、これら 2 つは別個の正当な名前として扱われます。実際、「JAMAL」と「KAMAL」は両方とも、**shortValidMax** パラメーターのデフォルト値よりも多く NDA に出現します。この理由からも、これら 2 つは別個の正当な名前として扱われます。

名前しきい値:

名前しきい値制御は、NameHunter が名前検索処理時に参照する最上位比較パラメーターです。この値によって、データベース上の各候補名前が一致と見なされるために合致するか超えなければならない全体スコアが定義されます。

この制御は、0.00 から 1.00 までの範囲の値です。1.00 に設定すると、すべての一致結果は照会名と完全に同じでなければなりません。これにより、検索結果に含まれる一致するレコードの数は、相対的に少なくなることが予想できます。照会名とデータベース上の候補名の間になんらかの違いがあれば、突き合わせ処理はその候補レコードを拒否します。

NameHunter API または NameHunter Distributed Search XML 属性を通して名前しきい値を指定して、この機能をオンまたはオフにできます。

NameHunter API

- CompParms::nameThreshold

NameHunter Distributed Search XML 属性

- GENERAL_PARMS NAME_THRESH

以下の例は、サンプル・データベースを対象にサブミットされた検索要求に対して、NameHunter Distributed Search が提供した検索結果を示しています。名前しきい値制御は 3 つの異なるレベルに設定されています。

以下の検索結果は、名前しきい値制御を 0.80 に設定した場合に得られたものです。このしきい値レベルでは、データ・リスト上の 3 つの名前が一致結果として適格でした。

Query name = Johnson, Robert

GN	SN	名前スコア	SN スコア	GN スコア
Bob	Johnson	0.98	1.00	0.95
Robert	Johnston	0.90	0.82	1.00
Robert Adamson	Johnston	0.90	0.82	0.99
Michael Robert	Johnson	0.82	1.00	0.59

名前しきい値を 0.70 に変更すると、以下の結果が戻されます。

GN	SN	名前スコア	SN スコア	GN スコア
Bob	Johnson	0.98	1.00	0.95
Robert	Johnston	0.90	0.82	1.00

GN	SN	名前スコア	SN スコア	GN スコア
Robert Adamson	Johnston	0.90	0.82	0.99
Michael Robert	Johnson	0.82	1.00	0.59
Craig Robert	Johnston	0.74	0.82	0.63
Robert	Jackson	0.72	0.50	1.00
Robert	Swanson	0.72	0.50	1.00

名前しきい値を 0.50 に変更すると、以下の結果が戻されます。

GN	SN	名前スコア	SN スコア	GN スコア
Bob	Johnson	0.98	1.00	0.95
Robert	Johnston	0.90	0.82	1.00
Robert Adamson	Johnston	0.90	0.82	0.99
Michael Robert	Johnson	0.82	1.00	0.59
Craig Robert	Johnston	0.74	0.82	0.63
Robert	Jackson	0.72	0.50	1.00
Robert	Swanson	0.72	0.50	1.00
Henry R	Rokeby Johnson	0.58	0.64	0.50
Albert Leslie	Swanson	0.53	0.50	0.57

フィールド制御:

基本 CompParms には、照会名とデータベース上の名前名 (GN) 構成要素と姓 (SN) 構成要素を NameHunter がどのように処理してスコアリングを行うかを決定する 2 つの制御が組み込まれています。

GN データの言語、国/地域別情報、統計、計算に関するプロパティは、SN データのそれらとは大きく異なります。そのため、これらの制御は、広範囲にわたって異なるデータベースの内容、アプリケーション設計、ユーザー設定に NameHunter が対応できるような方法で、GN および SN 処理の調整と改良をサポートします。

フィールド情報が NameHunter 突き合わせプロセスに関与する仕方を、次の 2 つの基本因子が決定します。

しきい値

照会名とその比較先の名前の間の差分の最小許容度。比較の類似性スコアによって表されます。

重みづけ

名前全体比較にとっての GN または SN フィールドの重要度。氏名比較の合計スコアの計算への GN または SN フィールド・スコアの相対的寄与によって表されます。

フィールドしきい値:

フィールドしきい値は、NameHunter 検索プロセスが実行する上で、照会フィールドとデータベース・レコードのフィールド構成要素の間で判別されなければならない最低レベルの類似性を設定するのに使用します。

名前しきい値制御と同様に、この制御も 0.00 から 1.00 までの範囲の値です。1.00 に設定した場合、すべてのデータベース・ヒットは照会名と完全に一致しなければなりません。検索結果に含まれる一致するレコードの数は、相対的に少なくなることが予想できます。照会名とデータベース上の候補名の間になんらかの違いがあれば、突き合わせ処理はその候補レコードを拒否します。

フィールドしきい値は、NameHunter API または NameHunter Distributed Search XML 属性を通じて指定できます。フィールドしきい値が 1.00 に近いほど、比較対象の名前の類似性は高くなければなりません。設定値が低いほど、2 つの名前の相違が大きくなります。姓 (GN) と名 (SN) の相対的重み付けを決定するために、姓重み付け制御と名重み付け制御が組み合わされます。

NameHunter API

- CompParms::threshold

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN FIELD_THRESH
- COMP_PARMS_SN FIELD_THRESH

注: FIELD_THRESH パラメーターは、組織名のスコアリング・ロジックでは無視されます。組織名には NAME_THRESH 比較パラメーターを使用して、データベース上の各候補名前が一致と見なされるために合致するか超えなければならない全体スコアを定義するための値を設定します。

以下は、GN しきい値を 0.80 に設定し、SN しきい値を 0.30 に設定した場合に戻される結果です。この表で示されているとおり、照会と結果の名前の GN データの間には、緊密な対応が存在します。

Query name = Johnson, Robert

GN	SN	名前スコア	SN スコア	GN スコア
Bob	Johnson	0.98	1.00	0.95
Robert	Johnston	0.90	0.82	1.00
Robert Adamson	Johnston	0.90	0.82	0.99
Robert	Jackson	0.82	1.00	0.59
Robert	Swanson	0.72	0.50	1.00
Robert	Nelson	0.67	0.40	1.00

GN しきい値を 0.30、SN しきい値を 0.80 に設定すると、名フィールドの種類が増える結果になります。

GN	SN	名前スコア	SN スコア	GN スコア
Bob	Johnson	0.98	1.00	0.95
Robert	Johnston	0.90	0.82	1.00
Robert Adamson	Johnston	0.90	0.82	0.99
Michael Robert	Johnson	0.82	1.00	0.59
Craig Robert	Johnston	0.74	0.82	0.63
Robert	Jackson	0.72	0.50	1.00
Robert	Swanson	0.72	0.50	1.00

GN	SN	名前スコア	SN スコア	GN スコア
Roy	Johnson	0.72	1.00	0.36

フィールド重みづけ:

フィールド重みづけ制御を使用して、氏名スコアを計算する際の、名スコアまたは姓スコアのフィールド・スコア相互の相対的重要度を決定します。

例えば、GN フィールド重みづけと SN フィールド重みづけを両方とも 1.00 に設定すると、これらのフィールド・スコアは氏名スコアの計算に等しく寄与します。GN 重みづけを 0.00 に設定し、SN 重みづけを 1.00 に設定すると、GN スコアは氏名スコアの計算時に、考慮すべき要素から事実上除去されます。

多くの国/地域別情報では、GN には変形形式 (例えばニックネームや呼称) が許されますが、一方 SN では、かたくなに固定形式が保たれます。2 人を区別するうえで、SN の表記が変わることは、名が変化することよりもはるかに大きな意味を持ちます。名と姓が持つ他との区別を示す意義のこの違いは、GN フィールド重みづけを SN フィールド重みづけよりも相対的に小さくすることによって取り込まれます。

NameHunter API または NameHunter Distributed Search XML 属性を通してフィールド重みづけを指定して、この機能をオンまたはオフにできます。

NameHunter API では、GN と SN の相対的重みづけを決定するために、GN 重みづけ制御と SN 重みづけ制御が組み合わせられます。

NameHunter API

- CompParms::weight

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN FIELD_WEIGHT
- COMP_PARMS_SN FIELD_WEIGHT

以下は、GN 重みづけを 0.20 に設定し (GN はあまり重要でない)、SN 重みづけを 1.00 に設定した場合の結果です。

Query name = Johnson, Robert

GN	SN	名前スコア	SN スコア	GN スコア
Bob	Johnson	0.98	1.00	0.95
Michael Robert	Johnston	0.93	1.00	0.59
Roy	Johnson	0.89	1.00	0.36
Robert	Johnston	0.85	0.82	100
Robert Adamson	Johnston	0.90	0.82	0.99
Craig Robert	Johnston	0.82	1.00	0.59
Henry R	Rokeby Johnson	0.61	0.64	0.50

以下は、GN 重みづけを 0.90 に設定した場合 (GN が重要である場合) の結果です。名前全体スコアへの影響は、前の例とは著しく異なります。

GN	SN	名前スコア	SN スコア	GN スコア
Bob	Johnson	0.98	1.00	0.95
Robert	Johnston	0.91	0.82	1.00
Robert Adamson	Johnston	0.90	0.82	0.99
Michael Robert	Johnson	0.82	1.00	0.59
Robert	Jackson	0.72	0.50	1.00
Robert	Swanson	0.72	0.50	1.00
Craig Robert	Johnston	0.74	0.82	0.63
Roy	Johnson	0.72	1.00	0.36

欠落語幹係数:

2 つの名前フィールドを比較する際に NameHunter は、フィールド内の名前句の数に違いがあるかどうかを検査します。

違いがある場合 (例えば、一方のフィールドが「de la Cruz Beltran」で、もう一方が「de la Cruz」である場合) は、ペナルティーが適用されます。ペナルティーの量は、欠落語幹係数パラメーターに設定された値によって決まり、この値は、フィールドのスコアに係数として適用されます。

このパラメーターの値は、0.00 から 1.00 までの任意の値が可能ですが、通常は 0.98 に設定されます。

NameHunter API または NameHunter Distributed Search XML 属性を通して欠落語幹係数を指定して、この機能をオンまたはオフにできます。

NameHunter API

- CompParms::missingStemFactor

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN
MISSING_STEM_FACTOR
- COMP_PARMS_SN
MISSING_STEM_FACTOR

以下の結果は、次の照会名に対して欠落語幹係数を 1.00 に設定した場合に戻されます。ペナルティーが割り当てられないので、名前は完全一致になります。

Query name = Johnson, Robert

GN	SN	名前スコア	SN スコア	GN スコア
Robert	Johnston	0.90	0.82	1.00
Robert Adamson	Johnston	0.90	0.82	1.00

以下の結果は、欠落語幹係数を 0.90 に設定した場合に戻されます。Robert Adamson は Robert ほど一致しているわけではないため、通常より厳しいペナルティーが割り当てられています。

GN	SN	名前スコア	SN スコア	GN スコア
Robert	Johnston	0.90	0.82	1.00

GN	SN	名前スコア	SN スコア	GN スコア
Robert Adamson	Johnston	0.86	0.82	0.90

イニシャル制御:

イニシャルは通常、異なる名前がいくらかでも一致する可能性があるため、略さずに書かれた名前よりも特定のではありません。NameHunter 固有の制御を使用して、イニシャルに関する一致の値を決定できます。

3 つの NameHunter CompParms 制御で、イニシャルの処理方法を制御します。どのパラメーターも、GN と SN に適用できます。

イニシャル突き合わせフラグ

この論理値 (true/false) は、指示した名前フィールドに NameHunter のイニシャル処理特殊ロジックを適用するかどうかを決定します。C++ API フィールドは `CompParm::matchInitials` です。対応する XML 属性は、`COMP_PARM_GN_MATCH_INITIAL` および `COMP_PARM_SN_MATCH_INITIAL` です。

イニシャルとイニシャル・スコア

2 つのセグメントが両方ともイニシャルであり、両方がまったく同じ場合に割り当てられる一致スコア (0.00 から 1.00 までの範囲の値)。C++ API フィールドは `CompParm::initialOnInitialScore` です。対応する XML 属性は、`COMP_PARM_GN_INITIAL_ON_INITIAL_SCORE` および `COMP_PARM_SN_INITIAL_ON_INITIAL_SCORE` です。

イニシャル一致スコア

一方のセグメントがイニシャルであり、もう一方が複数文字トークンで、その先頭文字がイニシャルと同じ (例えば H と Harold) 場合に割り当てられる一致スコア (0.00 から 1.00 までの範囲の値)。C++ API フィールドは `CompParm::initialOnTokenScore` です。対応する XML 属性は、`COMP_PARM_GN_INITIAL_ON_TOKEN_SCORE` および `COMP_PARM_SN_INITIAL_ON_TOKEN_SCORE` です。

以下の表に示す結果は、次の設定が適用された場合に返されます。

- イニシャル突き合わせ = オン
- `InitialOnInitialScore` = 0.95
- `InitialOnTokenScore` = 0.85

Query name = Mathers, X

GN	SN	名前スコア	SN スコア	GN スコア
X	Mathews	0.81	0.71	0.95
Xavier	Mathews	0.77	0.71	0.85

以下の結果は、イニシャル突き合わせがオフの場合に返されます。

Query name = Mathers, X

GN	SN	名前スコア	SN スコア	GN スコア
X	Mathews	0.84	0.71	1.00

InitialOnInitialScore の NameHunter デフォルトは 1.00 です。

欠落名前制御:

NameHunter には、データが含まれていないフィールドのスコアリングをどのように行うかを制御するためのパラメーターが組み込まれています。この欠落名前制御を使用して、2 つの名前の対応する名前フィールド (名または姓) を比較するとき比較できるデータが一方 (または両方) にない場合の比較を調整することができます。

名前のさまざまなコレクションにおける共通問題の 1 つとして、レコードの名前データ・フィールドの 1 つが空の場合があるという点が挙げられます。この問題が起こる原因として、フィールドへのデータ配置が間違っている (例えば、名前全体が姓フィールドに入れられた場合)、名前の一部がレコードから欠落している (姓のみが記録されている場合など)、または個人に単一の名前しかない (例えば元インドネシア大統領の名前 Suharto のように) ということが考えられます。欠落しているまたは空の名前フィールドに NameHunter で対処するために、次の 2 つの制御ペアが使用されます。

- 部分名前制御
- 非存在名前制御

部分名前制御:

部分名前スコア (0.00 から 1.00 までの範囲の値) は、一方の名前フィールドが不明と指定されているときに割り当てられます。

シンボリック値の FNU (ファーストネーム不明) および LNU (ラストネーム不明) は、NameHunter に名または姓が欠落しているか不明であることを知らせるコードです。NameHunter は NFN をブランクの名、NLN をブランクの姓として扱います。名フィールドまたは姓フィールドがブランクか、NFN または NLN が指定されている場合、NameHunter は、その名前フィールドをデータ・リスト・レコードの非欠落名前フィールドと比較する際に、対応する部分名前スコアを使用します。

名前フィールドが NFN、NLN、または空に指定されていて、不明 (FNU または LNU) のフィールドと比較される場合、結果として生じるスコアは次の式を使用して決定されます。

$$(\text{UnknownScore} + 1)/2$$

NameHunter API または NameHunter Distributed Search XML 属性を通して部分名前制御を指定して、この機能をオンまたはオフにできます。

NameHunter API

- CompParms::noNameScore
- CompParms::nameUnknownScore

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN NO_NAME_SCORE
- COMP_PARMS_GN
NAME_UNKNOWN_SCORE
- COMP_PARMS_SN NO_NAME_SCORE
- COMP_PARMS_SN
NAME_UNKNOWN_SCORE

以下の結果は、noNameScore = 0.80 で UnknownScore = 0.85 の場合に生成されます。表の中のブランクの項目は、フィールドが空であることを示します。

Query name = Farris, Travis

GN	SN	名前スコア	SN スコア	GN スコア
FNU	Farris	0.93	1.00	0.85
	Farris	0.91	1.00	0.80
NFN	Farris	0.91	1.00	0.80

非存在名前制御:

非存在名前制御は、名前が不明である場合や、該当する名前フィールドが存在しない場合に役立ちます。

国/地域別情報によっては、GN または SN のどちらかの役割を果たすことができる単一の名前のみで個人を示すことが認められているため、名前が不明であるケースと、該当する名前フィールドが存在しないケースを区別しなければならない場合があります。非存在名前スコア (0.00 から 1.00 までの範囲の値) は、一方の名前がファーストネーム (GN) またはラストネーム (SN) のどちらかがないと明示的に指定された場合 (特殊シンボリック値 (ファーストネームなし (NFN) およびラストネームなし (NLN)) をそれぞれ使用) の一致スコアを設定するための手段となります。

非存在名前制御は、姓のみが含まれる 2 つの名前を NameHunter が比較するときなど、両方の名前とも一方の名前フィールドの値がない場合にも適用されます。

これらのパラメーターは、NameHunter C++ API または NameHunter Distributed Search XML 属性を通して指定できます。

NameHunter API

- CompParms::noNameScore
- CompParms::nameUnknownScore

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN NO_NAME_SCORE
- COMP_PARMS_SN NO_NAME_SCORE
- COMP_PARMS_GN
NAME_UNKNOWN_SCORE
- COMP_PARMS_SN
NAME_UNKNOWN_SCORE

以下の結果は、noNameScore = 0.80 で unknownNameScore = 0.85 の場合に返されます。このタイプの照会の結果は、広範囲に及ぶことがあります。その場合は、あらゆる名がブランクの照会名と一致します。

Query name = Farris,

GN	SN	名前スコア	SN スコア	GN スコア
FNU	Farris	0.97	1.00	0.93
	Farris	1.00	1.00	1.00
NFN	Farris	0.96	1.00	0.80
James	Farris	0.91	1.00	0.80

注: これらの例では、英語向けパラメーターが適用されています。名前 James Farris の姓スコア (1.00) と名スコア (0.80) の平均をとると、平均は 0.90 になります。しかし、英語向けパラメーターを適用すると姓スコアの重みづけは名スコアよりわずかに高いので、全体名前スコアは 0.91 になります。

セグメント・スコアリング方式:

名前はしばしば、例えば Kate Marie Smith や Ana Ramos Sanchez のように、1つのフィールドに1つより多い名または姓が含まれます。NameHunter セグメント・スコア・パラメーターは、フィールド内の個々の名前 (セグメント) それぞれのスコアからフィールド全体 (GN または SN フィールド) のスコアをどのように決定するかを制御します。

3つの異なるスコアリング・モードを指定することによって、特定の GN または SN 名前句 (セグメント) を、対応する GN または SN 名前フィールドの複合スコアに NameHunter がどのように結合するかを決定できます。これらのオプションは、NameHunter C++ API または NameHunter Distributed Search XML属性を通して指定します。

NameHunter API

- CompParm::scoreMode

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN SCORE_MODE
- COMP_PARMS_SN SCORE_MODE

3つのスコアリング・モードがあります。名前フィールドに Highest 設定を選択した場合は、より多くの一致が得られるようになり、一方 Lowest 設定を選択した場合は、逆の効果があります。

Lowest

NameHunter は、フィールド内の個々の名前すべてについて比較スコアを計算します。フィールド全体のスコアとして、最低比較スコアがフィールドに割り当てられます。例えば、Gina Marie を Ginny Marie と比較するとします。Marie/Marie 比較のスコアは 1.0 になり、一方 Gina/Ginny 比較のスコアは低くなります (例えば 0.67)。この GN フィールドのスコアとして最低スコア 0.67 が割り当てられます。このスコアリング・モードの効果は、フィールド内のどの単一セグメントも、フィールドしきい値以上の比

較スコアでなければならないという点です。したがって、このスコアリング・モードは最も厳密であり、照会と比較対象の名前の間に最高レベルの類似性を必要とします。

Average

名前フィールドの複合スコアの計算には、名前フィールド内のすべてのセグメント・スコアの単純平均がとられます。

Highest

NameHunter は、フィールド内の個々の名前すべてについて比較スコアを計算します。フィールド全体のスコアとして、最高比較スコアがフィールドに割り当てられます。前の例 (Gina Marie/Ginny Marie) でこのスコアリング・モードを使用したとすると、GN フィールドのスコアは 1.0 になります。しきい値以上でなければならないのは、単一のセグメント比較だけです。このモードは、照会名と比較対象の名前の間に最大度の多様性を許すので、最も緩やかです。

以下は、SN と GN の両方ともセグメント・スコア・モードを Highest に設定した場合の結果です。

Query name = Hamilton Connerly, Lucinda Anna

GN	SN	名前スコア	SN スコア	GN スコア
Lucinda Anna	Hamilton Connor	1.00	1.00	1.00
Patricia Ann	Hamilton	0.99	0.98	1.00
Hubert A	Hamilton	0.92	0.98	0.85
Wade A	Hamilton	0.92	0.98	0.85
Linda	Charlton	0.62	0.54	0.71

以下は、SN と GN の両方ともセグメント・スコア・モードを Average に設定した場合の結果です。

GN 名前句 (Lucinda~Lucinda) の最高一致のスコアと、SN 名前句 (Hamilton~Hamilton) の最高一致のスコアが、それぞれ GN 名前フィールド・スコアと SN 名前フィールド・スコアとして使用されます。missingStemFactor は、異なる数のトークンが比較されるスコアにおいて、わずかな違いを計算に含めます。

Query name = Hamilton Connerly, Lucinda Anna

GN	SN	名前スコア	SN スコア	GN スコア
Lucinda Anna	Hamilton Connor	1.00	1.00	1.00
Patricia Ann	Hamilton	0.99	0.98	1.00
Linda	Charlton	0.62	0.54	0.71

すべてのトークン比較が名前スコアに寄与するので、一部の一致結果がドロップアウトされました。その結果、Hubert~Lucinda が獲得するスコアが非常に低いために、Hubert A Hamilton はリストからドロップされています。

アンカー・セグメント制御:

アンカー・セグメント・パラメーターは、最適一致と見なすためには一致する名前句が複数の名前のフィールド内の左端 (最初) の位置になければならないか、右端 (最後) の位置になければならないか、あるいはフィールド内の位置は問わないかを決定します。このパラメーターを使用すると、フィールド内の順序をスコアリングに反映できるようになります。

順序は、スペイン語の姓で重要です。例えば、2 つの姓のうち左端が個人の家名です。一方、右端の姓は母親の家系をそのまま表しており、しばしば名前から省略されます。

選択可能なオプションとして、より重要なものを左端の名前とするか、右端の名前とするか、あるいはどちらの名前であるかを問わないことを選択できます。

世界の多くの地域で、人は複数の名 (GN) または姓 (SN) を持っています。これらの名前の持つ役割の基準となったり、どのような状況でどの名前が使用されるかを左右したりする慣習は、集団ごとに異なります。

例えば、英語を話す人の間では、「ファーストネーム」と「ミドルネーム」の少なくとも 2 つの名を持つのが一般的です。ミドルネームは、省略されたり、場合によってはイニシャルだけで表されたりする場合があります。したがって、ROBERT WILSON と ROBERT JAMES WILSON と ROBERT J. WILSON はすべて、同一個人を指していると思われる可能性があります。他の国/地域別情報でも、同様のパターンの包含、省略、構文の使い方が見られます。アンカー・セグメント制御は、SN と GN の両方の名前フィールドについて、NameHunter が複数セグメントの名前フィールドの適切な部分に重点を置けるようにするためのものです。この制御では、フィールドの最初 (左端) のセグメントをアンカー・セグメントと見なすか、最後 (右端) のセグメントをアンカー・セグメントと見なすか、あるいはどのセグメントも他のセグメントよりも重要であると見なさない (なし) かを定義します。

スペイン語においては、姓アンカー・セグメントの設定は通常、「最初」です。左端の姓が各父方の姓であり、2 番目の母方の姓はしばしば省略されるからです。しかし、ブラジル、ポルトガル、および一部のアフリカ諸国の国/地域別情報など、Lusophone (ポルトガル語を話す) 国/地域別情報では、逆のことが当てはまります。

アンカー・セグメント制御は、NameHunter API または NameHunter Distributed Search XML 属性を通して指定できます。

NameHunter API

- CompParms::anchorType

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN ANCHOR_MODE
- COMP_PARMS_SN ANCHOR_MODE

名前フィールド・レベルのスコアリングへのアンカー・セグメントの影響の大きさは、アンカー係数の値によって決まります。この値は、NameHunter C++ API または NameHunter Distributed Search XML 属性を通して設定できます。

NameHunter API

- CompParm::anchorFactor

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN ANCHOR_FACTOR
- COMP_PARMS_SN ANCHOR_FACTOR

この係数は 0.00 から 1.00 までの値を受け入れ、一致した名前句の 1 つがアンカー・セグメント位置で見つからなかった場合に NameParser 一致スコアに適用されます。

次の例を考えてみましょう。この例は、GN フィールドに名前句が 2 つあり、SN フィールドに名前句が 2 つある、スペイン語の典型的な名前の検索を示しています。

Query name = Figueroa Martin, Ana Maria

GN	SN	名前スコア	SN スコア	GN スコア
Ana	Figueroa	0.98	0.98	0.99
Maria	Figueroa	0.90	0.98	0.80
Juana	Figueredo	0.61	0.62	0.59

この検索の場合、GN と SN のアンカー・セグメントは両方とも「最初」に設定されており、GN アンカー係数は 0.90 に設定されていました。

最初の一致した名前 Ana Figueroa では、名の Ana と姓の Figueroa は両方とも、左端の位置にあります (アンカー・セグメントは最初の位置)。したがって、このどちらの名前にもアンカー・セグメント係数は適用されません。2 番目の一致した名前 Maria Figueroa では、名の Maria が、検索対象の Ana Maria の名の 2 番目の名前と一致します。しかし、照会名に含まれる名前 Maria は、左端アンカー位置にありません。したがって、Maria に関する一致は優先一致ではないため、アンカー係数の適用によってペナルティーが科せられます。その結果、Maria と Maria の比較はスペルが完全に一致しているにもかかわらず、Maria に関する一致には 0.90 の値が付けられます。このより低い値とその他のペナルティーのために、GN スコアは 0.80 に低下します。

今度は、指定の名前句を逆にしてみましょう。設定は同じままにしておきます。

Query name = Figueroa Martin, Maria Ana

GN	SN	名前スコア	SN スコア	GN スコア
Maria	Figueroa	0.98	0.98	0.99
Ana	Figueroa	0.90	0.98	0.80

最高ランキングの一致記録には「Maria」が含まれており、その一致 GN スコアは 0.99 となります。一致スコアが 0.99 になっているのは、アンカー・セグメント位置 (この場合は左端) での GN 名前句が、同様にアンカー・セグメント位置にある GN 名前句と一致したからです。その結果、GN アンカーは適用されず、Maria/Maria 突き合わせでの名前句の完全一致の仮スコア 0.99 が、変更されずに GN 名前フィールドの最終スコアになります。

今度は、GN フィールドに Ana が含まれるデータ・リスト・レコードとの一致が、前の例で Maria に適用されたのと同じスコア低下の対象となります。Ana はもうアンカー・セグメント位置になく、両方の名フィールドの構文上の対応する位置に現れないためです。

非所定位置セグメント制御:

NameHunter の非所定位置セグメント (OOPS) 制御を使用して、名前フィールド内の同じ構文位置にない名前セグメント間で一致を判別する場合の名前フィールド・レベルのスコアリングを調整します。

OOPS 係数は、2 つの名前句 (例えば JAMES ROBERT) を含んだ GN フィールドが、同じ名前句を 1 つ以上含んでいるもののそれらの位置が異なる (例えば ROBERT JOSEPH や JOSEPH ROBERT) GN フィールドと突き合わせられる場合に役立ちます。

比較される 2 つの名前の GN または SN フィールド内の一致する名前句を検出する最適な方法を判別する際に、NameHunter はしばしば、GN または SN 名前フィールド内の同じ構文位置にない 2 つの名前句の最適一致を識別します。これは一般に、GN フィールドを突き合わせる場合が該当します。ほとんどの国/地域別情報では、GN は複数の名前句から成り、多くの名前に非常に一般的な GN 名前句が 1 つ以上含まれるためです。

OOPS 係数は、0.00 から 1.00 までの範囲の値です。1.00 に近いスコアほど、非所定位置一致の理由で科せられるペナルティーは小さくなります。0.00 に近いスコアほど、非所定位置一致の理由でペナルティーが科せられます。例えば、OOPS 係数を 1.00 に設定した場合、ペアの名前句が名前フィールド内の異なる位置にあったとしても、名前句一致の仮一致スコアは保持されます。

OOPS 係数は、NameHunter API または NameHunter Distributed Search XML 属性を通して指定できます。また、フィールド変形一致が検出されたときに戻されるスコアを設定できます。

NameHunter API

- CompParms::oopsFactor

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN OOPS_FACTOR
- COMP_PARMS_SN OOPS_FACTOR

次の NameHunter 検索結果を見てみましょう。ここでは、GN の OOPS 係数は値 0.80 に設定されています。

Query name = Duval, James Robert

GN	SN	名前スコア	SN スコア	GN スコア
James	Duval	1.00	1.00	0.99
Jim	Duval	0.97	1.00	0.94
Robert James	Duval	0.91	1.00	0.80
Robert	Duval	0.91	1.00	0.79
Bob	Duval	0.89	1.00	0.75

Joseph の一致名前句が同じ位置で検出された場合、OOPS 係数は適用されません。完全一致突き合わせを表す名前解析仮スコア 1.00 (James~James - missingStemFactor のために実際は 0.99) は、最終 GN スコアでも影響を受けずにそのまま残っています。

しかし、Robert の一致名前句が異なる位置 (照会名では非左端、データ・リスト上の一致名では左端) で検出された場合は、GN OOPS 係数が適用され、名前解析仮スコア 1.00 が 0.80 に下げられます。

圧縮名前制御:

圧縮名前制御を使用して、2 つの名前を比較する際のセグメンテーションと空白配置での偶発的な相違を軽減します。

個人名の大規模なコレクションで起こる共通の問題として、空白 (ブランク) の不整合配置が挙げられます。ブランクはしばしば、より多くの文字をデータ入力形式に適合させる目的で、手動処理によって除去されます。また、ブランクを除去して、名前フィールド内の別個のトークンが単一のトークンに縮小されるようにする自動化データ処理システムも多数あります。

名前に含まれる不整合ブランクのもう 1 つの主な原因は、非ラテン・アルファベット書記体系からラテン・アルファベット化形式に名前を変換するときに適用される標準がさまざまであるという点です。名前のオリジナル形式が非英字書記体系 (アラビア語、中国語、韓国語など) で表されている場合、ラテン・アルファベット化された名前の中のブランクの配置は、多くの場合、ラテン・アルファベット化を行う個人や自動化処理の裁量に委ねられます。そのため、固有の形式でまったく同じように書かれた名前の 2 つのインスタンスが、ラテン・アルファベット化後は、大きく異なる 2 つのものになる可能性があります。

NameHunter CompParms 制御は、2 つの名前を比較する際のセグメンテーションと空白配置での偶発的な相違を軽減および克服するためのメカニズムを提供します。圧縮名前制御を使用すると、NameHunter は、名前フィールド (GN または SN) 内のすべての配置を、あたかも論理的には単一値を構成しているかのように見なせるようになります。この単一値に、NameHunter の標準の名前類似性技法によってスコアが付けられます。圧縮名前処理が活動化されると、NameHunter はあたかもブランクがすべて除去されたかのように GN 名前フィールドと SN 名前フィールドのスコアを計算し、そのスコアが NameHunter の標準スコアリング・メトリックによって計算された名前フィールド・スコアよりも大きければ、そのスコアを使用します。

圧縮名前制御は、NameHunter API または NameHunter Distributed Search XML 属性を通して指定できます。

NameHunter API

- CompParms::doCompressedName
- CompParms::compressedScoreMax
- CompParms::compressedScoreFactor

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN
DO_COMPRESSED_NAME
- COMP_PARMS_GN
COMPRESSED_SCORE_MAX
- COMP_PARMS_SN
DO_COMPRESSED_NAME
- COMP_PARMS_SN
COMPRESSED_SCORE_MAX

SN または GN フィールドの doCompressedName フラグが true に設定されていると、まずその名前フィールドのスコアリングが標準的な方法で行われ、次にそのフィールドの圧縮形式の値でスコアリングが再度行われます。圧縮名前スコアが通常のフィールド・スコアよりも高ければ、compressedScoreMax がフィールド・スコアとして使用されます。 compressedScoreMax のデフォルト値は 0.95 です。

圧縮後スコア係数 CompParms::compressedScoreFactor は主に、圧縮名前によって好ましくない一致結果が戻される可能性のある組織名と共に使用するためのものです。この係数は、比較的意味のない単一用語が高スコアになる一致結果を回避するために、圧縮名前スコアを低くします。例えば、ABC CONSTRUCTION と XYZ CONSTRUCTION の圧縮後スコアは、用語 CONSTRUCTION があるためになんまり高くなります。圧縮後スコア係数は、圧縮後スコアが取得されたときは常に適用されます。結果スコアは compressedScoreMax と照合され、この 2 つのスコアのうち小さい方が一致スコアと見なされます。圧縮後スコア係数のデフォルト値は、組織名の場合が 0.90、個人名の場合が 1.00 です。

NameHunter の圧縮名前制御は、名前のラテン・アルファベット書記体系規則がさまざまである言語で、空白の配置が異なっている名前のインスタンス間で突き合わせを行うための、効果的なメカニズムを提供します。例えば、Li Ping、Liping、Li-Ping など、ラテン・アルファベット化された中国語の名前では、空白が異なります。

doCompressedName=true の場合の次の照会を考えてみましょう。このフラグが true に設定されていなかったら、以下の結果のどれも戻されません。

Query name = Abdulsalah, Mohamed

GN	SN	名前スコア	SN スコア	GN スコア
Mohamed	Abdul Salah	0.97	0.95	1.00
Mohamed	Abdel Salah	0.90	0.82	1.00
Mohamed	Abd El Salah	0.90	0.82	1.00
Mohamed	Abdel Salam	0.90	0.64	1.00

追加の考慮事項

圧縮名前スコアリングを実装する際は、以下の動作を考慮に入れてください。

- NameHunter には、組織名、特に複数トークンの組織名が誤った高スコアや偽陽性にならないようにするためのメカニズムが組み込まれています。

- 複数の連続 TAQ は、スコアリングから除外されます。さらに、称号、修飾子、組織指定子、職業的修飾子などの特定の TAQ は、スコアリング時に考慮されません。
- すべての圧縮名前スコアを低くするために、`compressedScoreFactor` 比較パラメーターが適用されます。`compressedScoreFactor` のデフォルト値は、組織名の場合が 0.90、個人名の場合が 1.0 です。

圧縮名前制御の影響は、検索処理速度を上げるために NameHunter によってデータ・リストの早期出力オプション が使用されている間は、抑止できます。このオプションを選択すると、データ・リスト上の一致しそうな名前を除去するために、照会名とデータ・リスト上の名前のペアワイズ比較の早期に、さまざまな迅速な計算が適用されます。早期出力オプションを使用した場合は、有効な圧縮名前一致の認識、適用、スコアリングが行われないことがあります。圧縮名前一致になり得るものが NameHunter の検索結果に含まれていない場合は、検索処理を加速するための早期出力オプションが使用されているかどうかを確認してください。早期出力オプションを解除して、同じ検索を再試行してください。

左方バイアス制御:

NameHunter は、2 つの名前の類似性の計算で一般的末尾の影響を軽減するために使用できる左方バイアス制御を提供します。

西ヨーロッパおよび北米の多くの文化圏における名前は、ある種の特性を共有しており、その一部はそれぞれの祖先が話していたロマンス語、ゲルマン語、スラブ語の一般的特徴を受け継いでいます。これらの国/地域別情報の 1 つの特性であり、名前マッチング・アルゴリズムにとって重要なものとして、多くの名前が同じ末尾を共有するという点が挙げられます。そのため、このような名前の左端にある文字は、右方にある文字よりも特徴的である可能性があります。

例えば、英語を話す人たちの多くの姓は父系の情報を反映しています。これは、父方を通して起源をたどる個人の血統に関する情報です。末尾の `-SON` は、英語を話す人の姓にしばしば見られます。

JOHNSON、STEVENSON、ROBERTSON、JEFFERSON はその例です。

同じような現象は、ヨーロッパの他の文化圏でも見られます。例えばロシア語では、`-OV`、`-OVA`、`-SKI`、`-SKY` などの多くの典型的末尾が、ラテン・アルファベット化された名前において高いパーセンテージで見られます。

2 つの名前が一致するかどうかを NameHunter が判別する際、名前の右端にある文字の意義が、左端にある文字より低い可能性があります。この現象を左方バイアス と呼び、左方バイアス制御によって調整されます。この制御はフラグで行い、`true` に設定すると、あらかじめ決められた類似性計算が適用されます。この類似性計算は、対応する名前フィールド (GN または SN) 内の、名前語幹の初めの部分に共通してより多くの文字が含まれるセグメントの突き合わせをサポートします。

左方バイアス制御は、NameHunter API または NameHunter Distributed Search XML 属性を通して指定できます。

NameHunter API

- CompParms::leftBias

NameHunter Distributed Search XML 属性

- COMP_PARMS_GN LEFT_BIAS
- COMP_PARMS_SN LEFT_BIAS

次の例では、左方バイアス制御が有効になっており、名前それぞれの右方にある違いの影響を軽減しています。左方バイアス制御を有効にすると、名前の右半分の類似文字の意義の影響が小さくなるので、この制御を無効にした場合よりもスコアは低くなります。

Query name = Tarkovsky,Andrei

GN	SN	名前スコア	SN スコア	GN スコア
Andrei	Tankovsky	0.83	0.70	1.00
Andrei	Tchiakovsky	0.69	0.44	1.00

左方バイアス制御を無効にすると、すべての文字が等しく名前スコアに寄与します。したがって、名前それぞれの右にある共有文字シーケンス OVSKY が比較の類似性スコアを押し上げるので、戻される名前スコアは高くなります。

Query name = Tarkovsky,Andrei

GN	SN	名前スコア	SN スコア	GN スコア
Andrei	Tankovsky	0.88	0.80	1.00
Andrei	Tchiakovsky	0.79	0.63	1.00

デフォルトの比較パラメーター

NameHunter には、名前を比較するときを使用される、デフォルトの比較パラメーターが組み込まれています。これらのデフォルトの比較パラメーターは、外部構成ファイルによってオーバーライドすることができます。この構成ファイル内で、設定された値のみがオーバーライドされます。欠落したパラメーターまたは未指定のパラメーターは変更されません。

CompParms::overrideDefaultParms() メソッドを使用して、ある特定の国/地域別情報およびフィールド・タイプのデフォルトの比較パラメーターを設定します。

Distributed Search パラメーター・メッセージは、しきい値などのフィールドに適用できる因子を受け入れます。例えば、名前しきい値を変更して、データベース上の各候補名前が一致と見なされるために合致するか超えなければならない全体スコアを定義できます。比較パラメーター指定変更ファイルで nameThreshold=1.00 を指定した場合、名前しきい値は 1.00 に変更されます。1.00 に設定すると、すべての一致結果は照会名と完全に同じでなければなりません。これにより、検索結果に含まれる一致するレコードの数は、相対的に少なくなることが予想できます。

因子としきい値の有効範囲は、NameHunter では 0.00 から 1.00 までです。別々に送信されたパラメーターまたは検索要求に組み込まれたパラメーターに、変更が適用されます。

照会を制御するパラメーターは、以下のように特定の順序で適用されます。

- 照会で指定されたパラメーターのオーバーライド
- `compparms.config` ファイルから読み取られた、デフォルトのオーバーライド・パラメーター
- 国/地域別情報およびフィールド・タイプの、組み込みのデフォルト・パラメーター

NameHunter の比較パラメーター (組織名):

次の表は、組織名のデフォルト比較パラメーターを示しています。

	組織名
名前しきい値	0.55
重みづけ	1.00
左方バイアス	FALSE
イニシャル突き合わせ	TRUE
イニシャルとトークン	0.00
イニシャルとイニシャル	1.00
変形突き合わせ	TRUE
名前不明	0.40
名前なし	0.40
アンカー・タイプ	アンカーなし
アンカー係数	1.00
OOPS 係数	0.97
圧縮スコア実施	TRUE
圧縮スコア最大	0.95
圧縮スコア係数	0.90
スコア・モード	ScoreModeAverage
欠落語幹係数	0.95
フィールド変形突き合わせ	TRUE

組織名には、短い名前のスコアリング・パラメーターは使用できません。

NameHunter の比較パラメーター (未確定 - アラビア語):

次の表は、未確定、アフガニスタン語、英語、アラビア語の国/地域別情報コードに適用される名 (GN) フィールドと姓 (SN) フィールドのデフォルト・パラメーターを示しています。

	未確定 - 0		アフガニスタン語 - 16		英語 - 1		アラビア語 - 2	
	GN	SN	GN	SN	GN	SN	GN	SN
名前しきい値	0.60	0.60	0.65	0.65	0.65	0.65	0.65	0.65
フィールドしきい値	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
フィールド重みづけ	0.80	1.00	1.00	1.00	0.80	1.00	1.00	1.00

	未確定 - 0		アフガニスタン語 - 16		英語 - 1		アラビア語 - 2	
語幹欠落	0.99	0.98	0.99	0.98	0.99	0.98	0.99	0.98
変形突き合わせ	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
イニシャル突き合わせ	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
イニシャルとイニシャル	0.90	0.00	0.85	0.00	0.90	0.00	0.85	0.00
イニシャルとトークン	0.85	0.00	0.75	0.00	0.85	0.00	0.75	0.00
名前不明	0.60	0.60	0.60	0.75	0.60	0.60	0.75	0.75
名前なし	0.60	0.60	0.60	0.75	0.60	0.60	0.75	0.75
スコア・モード	AVG	AVG	AVG	AVG	AVG	AVG	AVG	AVG
アンカー・タイプ	NONE	NONE	FIRST	NONE	NONE	LAST	FIRST	NONE
アンカー係数	1.00	1.00	0.85	1.00	1.00	0.70	0.90	1.00
OOPS 係数	0.97	0.97	0.85	0.90	0.97	0.97	0.85	0.90
圧縮名前	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
圧縮後最大	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
圧縮スコア係数	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
左方バイアス	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
正規化最大	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
短い名前の最小長	2	2	2	2	2	2	2	2
短い名前の最大長	6	6	6	6	6	6	6	6
短い名前のスコア	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
短い名前の妥当性係数	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
短い名前の妥当性比率	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
短い名前の妥当性最大	200	200	200	200	200	200	200	200

NameHunter の比較パラメーター (中国語 - ドイツ語):

次の表は、中国語、ペルシア語、フランス語、ドイツ語の国/地域別情報コードに適用される名 (GN) フィールドと姓 (SN) フィールドのデフォルトの比較パラメーターを示しています。

	中国語 - 3		ペルシア語 - 12		フランス語 - 7		ドイツ語 - 8	
	GN	SN	GN	SN	GN	SN	GN	SN
名前しきい値	0.70	0.70	0.65	0.65	0.65	0.65	0.65	0.65
フィールドしきい値	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
フィールド重みづけ	0.80	1.00	1.00	1.00	0.80	1.00	0.80	1.00
語幹欠落	0.70	0.70	0.99	0.98	0.99	0.98	0.99	0.98
変形突き合わせ	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
イニシャル突き合わせ	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
イニシャルとイニシャル	0.85	0.00	0.85	0.00	0.85	0.00	0.85	0.00
イニシャルとトークン	0.85	0.00	0.75	0.00	0.75	0.00	0.75	0.00
名前不明	0.65	0.65	0.60	0.75	0.65	0.65	0.60	0.60
名前なし	0.65	0.65	0.60	0.75	0.65	0.65	0.60	0.60
スコア・モード	LOW	AVG	AVG	AVG	AVG	AVG	AVG	AVG
アンカー・タイプ	NONE	NONE	FIRST	NONE	NONE	LAST	NONE	LAST
アンカー係数	1.00	1.00	0.90	1.00	1.00	0.85	1.00	0.85
OOPS 係数	0.85	0.75	0.85	0.90	0.97	0.97	0.97	0.97
圧縮名前	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
圧縮後最大	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
圧縮スコア係数	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
左方バイアス	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
正規化最大	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
短い名前の最小長	2	2	2	2	2	2	2	2
短い名前の最大長	6	6	6	6	6	6	6	6
短い名前のスコア	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
短い名前の妥当性係数	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
短い名前の妥当性比率	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

	中国語 - 3		ペルシア語 - 12		フランス語 - 7		ドイツ語 - 8	
短い名前の 妥当性最大	200	200	200	200	200	200	200	200

NameHunter の比較パラメーター (スペイン語 - 日本語):

次の表は、スペイン語、インド語、インドネシア語、日本語の国/地域別情報コードに適用される名 (GN) フィールドと姓 (SN) フィールドのデフォルトの比較パラメーターを示しています。

	スペイン語 - 4		インド語 - 14		インドネシア語 - 10		日本語 - 15	
	GN	SN	GN	SN	GN	SN	GN	SN
名前しきい 値	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
フィールド しきい値	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
フィールド 重みづけ	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00
語幹欠落	0.99	0.98	0.99	0.98	0.99	0.98	0.99	0.98
変形突き合 わせ	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
イニシャル 突き合わせ	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
イニシャル とイニシャ ル	0.85	1.00	1.00	0.00	0.85	0.00	0.80	0.00
イニシャル とトークン	0.85	0.85	0.85	0.00	0.75	0.00	0.80	0.00
名前不明	0.60	0.60	0.55	0.50	0.70	0.70	0.55	0.55
名前なし	0.60	0.60	0.55	0.50	0.70	0.70	0.55	0.55
スコア・モ ード	AVG	AVG	AVG	AVG	AVG	AVG	AVG	AVG
アンカー・ タイプ	NONE	FIRST	NONE	NONE	NONE	NONE	NONE	NONE
アンカー係 数	1.00	0.70	1.00	1.00	1.00	1.00	1.00	1.00
OOPS 係数	0.90	0.80	0.90	0.90	0.97	0.97	0.90	0.75
圧縮名前	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
圧縮後最大	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
圧縮スコア 係数	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
左方バイア ス	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
正規化最大	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
短い名前の 最小長	2	2	2	2	2	2	2	2

	スペイン語 - 4		インド語 - 14		インドネシア語 - 10		日本語 - 15	
短い名前の最大長	6	6	6	6	6	6	6	6
短い名前のスコア	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
短い名前の妥当性係数	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
短い名前の妥当性比率	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
短い名前の妥当性最大	200	200	200	200	200	200	200	200

NameHunter の比較パラメーター (韓国語 - ロシア語):

次の表は、韓国語、パキスタン語、ポーランド語、ポルトガル語、およびロシア語の国/地域別情報コードに適用される名 (GN) フィールドと姓 (SN) フィールドのデフォルトの比較パラメーターを示しています。

	韓国語 - 5		パキスタン語 - 13		ポーランド語 - 18		ポルトガル語 - 19		ロシア語 - 6	
	GN	SN	GN	SN	GN	SN	GN	SN	GN	SN
名前しきい値	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
フィールドしきい値	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
フィールド重みづけ	0.80	1.00	1.00	1.00	0.80	1.00	0.80	1.00	0.80	1.00
語幹欠落	0.85	0.75	0.99	0.98	0.99	0.98	0.99	0.98	0.99	0.98
変形突き合わせ	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
イニシャル突き合わせ	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
イニシャルとイニシャル	0.85	0.00	0.85	0.00	1.00	0.00	1.00	0.00	0.80	0.00
イニシャルとトークン	0.85	0.00	0.75	0.00	0.85	0.00	0.85	0.00	0.85	0.00
名前不明	0.60	0.60	0.60	0.75	0.65	0.65	0.65	0.65	0.65	0.65
名前なし	0.60	0.65	0.60	0.75	0.65	0.65	0.65	0.65	0.65	0.65
スコア・モード	AVG	AVG	AVG	AVG	AVG	AVG	AVG	AVG	AVG	AVG
アンカー・タイプ	NONE	NONE	FIRST	LAST	NONE	NONE	NONE	NONE	FIRST	NONE

	韓国語 - 5		パキスタン語 - 13		ポーランド語 - 18		ポルトガル語 - 19		ロシア語 - 6	
アンカー 係数	1.00	1.00	0.85	0.85	1.00	1.00	1.00	1.00	0.80	1.00
OOPS 係 数	0.85	0.75	0.85	0.90	0.90	0.80	0.90	0.80	0.90	0.85
圧縮名前	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
圧縮後最 大	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
圧縮スコ ア係数	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
左方バイ アス	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
正規化最 大	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
短い名前 の最小長	2	2	2	2	2	2	2	2	2	2
短い名前 の最大長	6	6	6	6	6	6	6	6	6	6
短い名前 のスコア	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
短い名前 の妥当性 係数	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
短い名前 の妥当性 比率	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
短い名前 の妥当性 最大	200	200	200	200	200	200	200	200	200	200

NameHunter の比較パラメーター (タイ語 - ヨルバ語):

次の表は、タイ語、トルコ語、ベトナム語、およびヨルバ語の国/地域別情報コードに適用される名 (GN) フィールドと姓 (SN) フィールドのデフォルトの比較パラメーターを示しています。

	タイ語 - 9		トルコ語 - 20		ベトナム語 - 17		ヨルバ語 - 11	
	GN	SN	GN	SN	GN	SN	GN	SN
名前しきい 値	0.60	0.60	0.65	0.65	0.70	0.70	0.60	0.60
フィールド しきい値	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
フィールド 重みづけ	1.00	1.00	0.80	1.00	0.80	1.00	1.00	1.00
語幹欠落	0.99	0.98	0.99	0.98	0.99	0.98	0.99	0.98

	タイ語 - 9		トルコ語 - 20		ベトナム語 - 17		ヨルバ語 - 11	
変形突き合わせ	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
イニシャル突き合わせ	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
イニシャルとイニシャル	0.90	0.00	1.00	0.00	0.85	0.00	0.85	0.00
イニシャルとトークン	0.85	0.00	0.85	0.00	0.85	0.00	0.85	0.00
名前不明	0.50	0.50	0.60	0.60	0.65	0.65	0.55	0.55
名前なし	0.50	0.50	0.60	0.60	0.65	0.65	0.55	0.55
スコア・モード	AVG	AVG	AVG	AVG	LOW	AVG	AVG	AVG
アンカー・タイプ	NONE	NONE	NONE	LAST	NONE	NONE	NONE	NONE
アンカー係数	1.00	1.00	1.00	0.70	1.00	1.00	1.00	1.00
OOPS 係数	0.90	0.75	0.97	0.97	0.85	0.80	0.97	0.97
圧縮名前	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
圧縮後最大	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
圧縮スコア係数	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
左方バイアス	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
正規化最大	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
短い名前の最小長	2	2	2	2	2	2	2	2
短い名前の最大長	6	6	6	6	6	6	6	6
短い名前のスコア	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
短い名前の妥当性係数	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
短い名前の妥当性比率	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
短い名前の妥当性最大	200	200	200	200	200	200	200	200

NameHunter の比較パラメーター (グループの国/地域別情報):

次の表は、すべての有効なグループの国/地域別情報コードに適用される名 (GN) フィールドと姓 (SN) フィールドのデフォルトの比較パラメーターを示しています。

西南アジア

アフガニスタン語、アラビア語、ペルシア語、パキスタン語、およびトルコ語の国/地域別情報が含まれるグループの国/地域別情報。

ヨーロッパ

英語、フランス語、ドイツ語、スペイン語、およびポルトガル語の国/地域別情報が含まれるグループの国/地域別情報。

ハン 中国語、韓国語、ベトナム語の国/地域別情報が含まれるグループの国/地域別情報。

	西南アジア - 38		ヨーロッパ - 39		ハン - 40	
	GN	SN	GN	SN	GN	SN
名前しきい値	0.65	0.65	0.65	0.65	0.70	0.70
フィールドしきい値	0.49	0.49	0.49	0.49	0.49	0.49
フィールド重みづけ	1.00	1.00	0.80	1.00	0.80	1.00
語幹欠落	0.99	0.98	0.99	0.98	0.70	0.70
変形突き合わせ	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
イニシャル突き合わせ	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
イニシャルとイニシャル	0.85	0.00	0.85	0.00	0.85	0.00
イニシャルとトークン	0.75	0.00	0.75	0.00	0.85	0.00
名前不明	0.60	0.75	0.60	0.60	0.60	0.60
名前なし	0.60	0.75	0.60	0.60	0.60	0.60
スコア・モード	AVG	AVG	AVG	AVG	LOW	AVG
アンカー・タイプ	FIRST	NONE	NONE	NONE	NONE	NONE
アンカー係数	0.90	1.00	1.00	1.00	1.00	1.00
OOPS 係数	0.85	0.90	0.90	0.75	0.85	0.75
圧縮名前	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
圧縮後最大	0.95	0.95	0.95	0.95	0.95	0.95
圧縮スコア係数	1.00	1.00	1.00	1.00	1.00	1.00
左方バイアス	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
正規化最大	0.98	0.98	0.98	0.98	0.98	0.98
短い名前の最小長	2	2	2	2	2	2
短い名前の最大長	6	6	6	6	6	6
短い名前のスコア	0.97	0.97	0.97	0.97	0.97	0.97
短い名前の妥当性係数	0.75	0.75	0.75	0.75	0.75	0.75

	西南アジア - 38	ヨーロッパ - 39	ハン - 40
短い名前の妥当性比率	0.01	0.01	0.01
短い名前の妥当性最大	200	200	200

NameHunter の音訳規則セットの構成

NameHunter で追加的な音訳規則セットを使用するには、それらを使用するよう NameHunter、NHServer、および Distributed Search プロセスを構成する必要があります。

音訳規則ファイルを使用するための NameHunter の構成

IBM InfoSphere Global Name Management には、NameHunter の音訳関数と連動する規則ファイルが含まれています。規則ファイルを使用する前に、音訳をオンにしてから、特定の規則ファイルをロードするよう関数を呼び出す必要があります。

始める前に

- transliterate 関数を使用して、音訳をオンにします。
- 使用する規則ファイルがパス内にあることを確認します。

このタスクについて

NameHunter インスタンスをインスタンス化した後で、次の関数を呼び出します。

```
void NameHunter::loadTransRules(const std::string& ruleFileName);
```

ruleFileName を、適切な規則ファイルの場所で置き換えます。

注: 最適な結果を得るために、規則ファイルはパス内にある必要があります。規則ファイルがパス内にない場合、NameHunter は例外をスローします。

音訳規則ファイルの呼び出しの例

例えば、アラビア語の規則ファイルを呼び出すには、次の関数を呼び出します。

```
void NameHunter::loadTransRules(const std::string& arabicTransRule.ibm);
```

音訳規則ファイルを使用するための Distributed Search プロセスの構成

IBM InfoSphere Global Name Management には、音訳機能を変更する規則ファイルが含まれています。規則ファイルの名前と場所を含むよう Distributed Search プロセスの構成ファイルを変更した後、Distributed Search プロセスは、これらの規則ファイルによって扱われるスクリプトに記述された名前を突き合わせるできるようになります。

手順

1. Distributed Search プロセスの構成ファイルを開きます。デフォルト・ファイルは `dsconfig.ini` です。

2. [searcherCommon] セクションの下で、含める音訳ファイルを指定します。以下の例では、韓国語と日本語の音訳ファイルが含まれています。

```
[searcherCommon]
```

```
koreanTransFile=koreanTransRule.ibm  
japaneseTransFile=japaneseTransRule.ibm
```

Distributed Search を使用した名前の検索

NameHunter Distributed Search を使用すると、追加の検索サーバーをサポートするためにプロセッサ・リソースを追加することにより、許可される同時検索の数、および検索される名前の数 (最大 2 億) を増やすことができます。

次の図に、Distributed Search のハイレベルな設計を示します。(Distributed Search の一部ではない) 複数のクライアントが、TCP/IP を利用した XML によって中央の通信ミドルウェア・プロセスと通信します。ミドルウェア・プロセスはクライアント要求をキューに入れ、検索サーバーのセットに送信します。ミドルウェア・プロセスは検索サーバーからの応答を管理し、要求元のクライアントに 1 つの集約した応答を返します。クライアントには、要求への応答を生成するために複数のサーバーが寄与していることは分かりません。

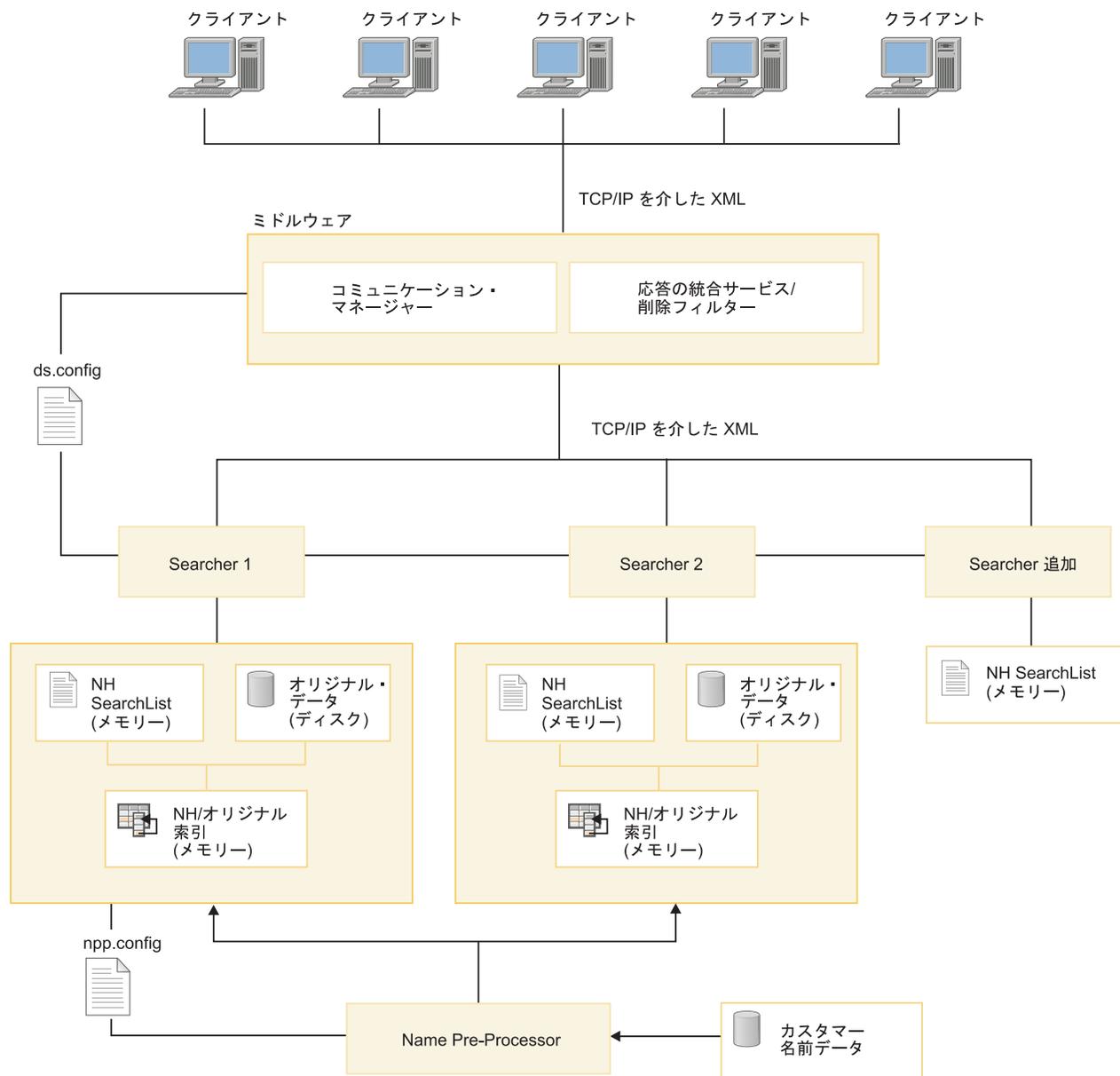


図 4. Distributed Search のハイレベルな設計

各 Distributed Search サーバーは、大きな名前データ・リストの一部を管理します。各 Distributed Search サーバーが管理するデータ・リストは、メモリーに常駐する検索リストと、ユーザーが指定するオリジナルの名前データのディスク・ベースのリポジトリへの索引で構成されます。追加要求が作成された場合など、Distributed Search セッション中に名前の結合リポジトリに行く必要のある更新を処理するために、通常は 1 台のサーバーが割り振られます。

各サーバーが管理するデータは、Name Preprocessor という Distributed Search のユーティリティーによって作成されます。名前データのコンマで区切られたファイルが指定されると、Name Preprocessor は、解析、分類、正規化、重複の削除を行い、すべての関与する Distributed Search サーバーが共有するために大きな名前ファイルを小さな部分に区分します。Name Preprocessor からの出力はバイナリーで、システム固有です。

重要: Name Preprocessor ユーティリティは、Distributed Search プロセスが実行されているのと同じオペレーティング・システム (OS) およびアーキテクチャーを持つシステムで実行する必要があります。Name Preprocessor が別のオペレーティング・システムで実行されている場合、Distributed Search は失敗します。

Name Preprocessor の概要

Name Preprocessor は、カスタマー名前レコードのコンマ区切りファイルを、Distributed Search が必要とするファイルに変換するユーティリティです。

Name Preprocessor は、名前分析関数 (音訳、解析、および分類) と NameHunter の内部プリプロセッシングのステップ (正規化とクレンジング) を結合して、Distributed Search プロセスへの入力として適切な名前のセットを生成します。

重要: Name Preprocessor ユーティリティは、Distributed Search プロセスが実行されているのと同じオペレーティング・システム (OS) およびアーキテクチャーを持つシステムで実行する必要があります。Name Preprocessor が別のオペレーティング・システムで実行されている場合、Distributed Search は失敗します。

Name Preprocessor のデータ・ファイルの概要

コンマで区切られたテキスト・ファイルがいくつか、Name Preprocessor によって消費および生成されます。Name Preprocessor は、情報を均一な断片に分割し、重複を取り除くので、Distributed Search および Embedded Search のプロセスが受け入れ可能な形式の単一のデータ項目になります。

カスタマー・データ・ファイルは、Name Preprocessor への入力であり、そこで解析、分類、正規化、音訳、および重複の確認が行われます。この最初のパスでは、一時データ・ファイルと呼ばれる、一時的なコンマ区切りのファイルが生成されます。一時データ・ファイルが小さな断片に分割される非重複化プロセスの結果、次の 3 つの出力ファイルが生成されます。それらは、Distributed Search および Embedded Search プロセスに提供されるデータ構造に区分されます。

- NH の検索リスト
- オリジナル・データ
- NH のオリジナル索引

データを Name Preprocessor に入力することは、Distributed Search を使用する場合のデフォルトの設定ではありません。別のアプリケーションを使用して情報を既に処理した場合は、Name Preprocessor をバイパスし、名前データを使用して Searcher プロセスを直接ロードできます。Name Preprocessor を使用して情報を処理する場合は、Name Preprocessor の構成ファイル `npp.config` に、処理するファイルを指定します。

名前のプリプロセッシングの流れ

次のような入力名前レコードがあるとします。

Jirí, Válek, 1234

このレコードに対して、Name Preprocessor で次のような処理ステップを実行します。

1. 名前音訳コンポーネントが、拡張 ASCII 文字を削除して次のフォームを生成します。
JIRI, VALEK, 1234
2. NameParser コンポーネントが、氏名を名 (GN) コンポーネントと姓 (SN) コンポーネントに解析し、カスタマー・データで元々提供された名前の順序が間違っていないかを検証します。この名前に対する 2 番目の、代わりにの解析順序が生成されます。
Valek, Jiri, 1234
3. NameClassifier は、SN と GN のそれぞれがヨーロッパの国/地域別情報 (国/地域別情報コード 39) に最適に関連付けられることを決定します。
4. 国/地域別情報を基本としたスペルの差の主なパターンを平滑化する言語規則 (名前の正規化) を、オプションで入力名に適用できます。

このため、Name Preprocessor で、オリジナルの入力名から次の一時的な CSV 形式の出力レコードを生成します。

```
1234,P,JIRI,VALEK,39,39,Jirí,Válek,0,0  
1234,P,VALEK,JIRI,39,39,Jirí,Válek,1,0
```

オリジナルの名前データは変更されません。Name Preprocessor によって実行される処理の 2 番目のパス中に、暫定ファイルのソート、重複除外、および Distributed Search サーバーと Embedded Search サーバーが必要とするデータ構造への区分が実行されます。

Embedded Search を使用した名前リストの前処理:

Embedded Search は、NameHunter の検索機能を、IBM NameWorks が初期化するとき、別のステップではなく名前データが前処理される単一プロセスで IBM NameWorks に公開します。

Embedded Search は、名前の前処理を行って、管理のオーバーヘッドを減らし、全体的なパフォーマンスを向上させます。これにより、アプリケーションはデータを前処理しなくても検索を開始できます。

Embedded Search で使用する名前データ・ファイルは、コンマ区切り値 (CSV) の形式で、各レコードに 3 個から 10 個のフィールドが含まれたものを指定する必要があります。これらのデータ・ファイルは、ASCII または UTF-8 のエンコードが想定されています。その他のエンコードのテキストでは、エラー状態が発生し、レコード番号と名前レコードの内容を含む BADRECORD 例外 (GODW033E) がスローされます。

重要: Embedded Search と Distributed Search は、両方とも Name Preprocessor の暫定データ・ファイルから読み取ります。Name Preprocessor を使用して名前データを前処理する場合、すべてのデータ・フィールドを指定する必要があるのに対し、Embedded Search を使用して名前を前処理する場合は、データ・フィールドはオプションです。

Name Preprocessor のカスタマー・データ・ファイル:

カスタマー・データ・ファイルは、Name Preprocessor ユーティリティーへの入力です。このファイルは、Distributed Search に渡せるように小さな断片に分割されている暫定データ・ファイルを生成するために処理されます。

カスタマー・データ・ファイルは、次のフィールドを含むコンマで区切られたファイルです。

レコード ID

このレコードに関連付けられる追加データ (データのオリジナル・ソースを指す ID など)。最大長は 256 バイトです。

カテゴリー

名前に指定する名前カテゴリー。ブランクのままの場合、Name Preprocessor がカテゴリーの判別を試みます。有効値は以下のとおりです。

- B = 個人と組織の両方
- O = 組織名
- P = 個人名

無効な値の場合、エラーが報告され、問題となっている名前レコードは破棄されます。

スクリプト・タイプ

名前テキストの文字タイプを次のように指定します。1 から 8 までの数値で特定の文字タイプを示し、0 で特定の文字タイプがないことを示し、ブランクで文字タイプの判別を Name Preprocessor によって試みる必要があることを示します。有効値は以下のとおりです。

- 0 = 特定の文字なし。
- 1 = 漢字 (中国語)
- 2 = 漢字/仮名文字 (日本語)
- 3 = デーバナーガリー文字 (インド)
- 4 = キリル文字 (ロシア語)
- 5 = ラテン文字 (ヨーロッパ、英語)
- 6 = ハングル文字 (韓国語)
- 7 = アラビア文字 (アラビア語、西南アジア)
- 8 = ギリシャ文字 (ギリシャ語)

無効な値の場合、エラーが報告され、問題となっている名前レコードは破棄されます。

姓または氏名

名前レコードに 3 つのフィールドのみが含まれている場合、このフィールドは氏名として解釈されます。名前レコードが組織名として分類された場合 (データ・ファイルで直接的に、または IBM NameWorks によって間接的に)、名フィールドと名の国/地域別情報フィールドは無視されます。その他のインスタンスでは、名前は音訳、解析、分類、およびデータ・リストへの追加が行われます。

名 その前のフィールドが姓エレメントとして解釈された場合は名エレメントとして解釈されます。追加のフィールドが存在しない場合、名エレメントおよび姓エレメントは音訳、解析、分類、およびデータ・リストへの追加が行われます。

姓の国/地域別情報コード

姓の国/地域別情報コード。このフィールドがブランク、または値 -1 を含む場合、IBM NameWorks は姓エレメントを分類して、その分類の結果を姓の国/地域別情報コードとして使用します。このフィールドに無効値が含まれている場合、エラー状態が報告されます。

名の国/地域別情報コード

名の国/地域別情報コード。このフィールドがブランク、または値 -1 を含む場合、IBM NameWorks は名エレメントを分類して、その分類の結果を名の国/地域別情報コードとして使用します。このフィールドに無効値が含まれている場合、エラー状態が報告されます。

オリジナルの姓

このフィールドが設定されている場合、オリジナルの姓エレメント。このフィールドには、組織名の元のフルネームが入ります。

オリジナルの名

オリジナルの名エレメント。

代替解析のフラグ

その前のフィールドによって判別された内容にかかわらず、現在のレコードがオリジナルの名前の代替解析であることを示します (0 = いいえ、1 = はい)。

Name Preprocessor の暫定データ・ファイル:

暫定データ・ファイルは、Name Preprocessor ユーティリティーの最初のパスの出力である、一時的な、コンマで区切られたファイルです。このファイルは、Name Preprocessor をカスタマー・データ・ファイルに適用 (解析、分類など) した結果として生成されるため、ファイルには Distributed Server への入力を生成するのに必要とされる情報がすべて含まれています。

重要: Embedded Search と Distributed Search は、両方とも Name Preprocessor の暫定データ・ファイルから読み取ります。Name Preprocessor を使用して名前データを前処理する場合、すべてのデータ・フィールドを指定する必要があるのに対し、Embedded Search を使用して名前を前処理する場合は、データ・フィールドはオプションです。

名前の名前カテゴリー (個人または組織のいずれか) に基づき、さまざまなフィールドが組み込まれます。名前が組織名であると判別された場合、名前の前処理中に、以下のフィールドのみが組み込まれます。

組織名の暫定データ・フォーマット

- 補足データ
- カテゴリー
- スクリプト・タイプ
- 名前

- 国/地域別情報コード
- オリジナルの名前

暫定データ・フォーマットの主な違いは、組織名が単一フィールドとして格納されます。

データ・ファイルには、以下のフィールドのうち、任意のものが含まれます。指定されたフィールドの数によって、どの前処理ステップを実行するかが決定されます。

補足データ

通常は、1 つ以上のデータベースで名前レコードに関するその他の情報を取得するのに使用できるレコード ID。ただし、このフィールドは、名前レコードに関する追加情報を含むことができます。

カテゴリー

名前レコードが、個人 (P) 名か、組織 (O) 名か、それともその両方 (B) として分類される必要があるかを示す単一の文字。このフィールドが空の場合、名前が IBM NameWorks によって分類されてからその後の処理が適用されます。無効な文字を指定すると、エラー条件が発生します。

スクリプト・タイプ

名前テキストの文字タイプを次のように指定します。1 から 8 までの数値で特定の文字タイプを示し、0 で特定の文字タイプがないことを示し、ブランクで文字タイプの判別を Name Preprocessor によって試みる必要があることを示します。

- 0 = 特定の文字なし。
- 1 = 漢字 (中国語)
- 2 = 漢字/仮名文字 (日本語)
- 3 = デーバナーガリー文字 (インド)
- 4 = キリル文字 (ロシア語)
- 5 = ラテン文字 (ヨーロッパ、英語)
- 6 = ハングル文字 (韓国語)
- 7 = アラビア文字 (アラビア語、西南アジア)
- 8 = ギリシャ文字 (ギリシャ語)

姓または氏名

名前レコードに 3 つのフィールドのみが含まれている場合、このフィールドは氏名として解釈されます。名前レコードが (データ・ファイルで直接、または IBM NameWorks によって間接的に) 組織名として分類された場合、追加のフィールドは無視されます。その他のインスタンスでは、名前は音訳、解析、分類、およびデータ・リストへの追加が行われます。

名 その前のフィールドが姓エレメントとして解釈された場合は名エレメントとして解釈されます。このフィールドがブランクであり、名前レコードが (データ・ファイルで直接、または IBM NameWorks によって間接的に) 組織として分類された場合、追加のフィールドは無視されます。追加のフィール

ドが存在しない場合、名エレメントおよび姓エレメントは音訳、解析、分類、およびデータ・リストへの追加が行われます。

姓の国/地域別情報コード

姓の国/地域別情報コード。このフィールドがブランク、または値 -1 を含む場合、IBM NameWorks は姓エレメントを分類して、その分類の結果を姓の国/地域別情報コードとして使用します。このフィールドに無効値が含まれている場合、エラー状態が報告されます。

名の国/地域別情報コード

名の国/地域別情報コード。このフィールドがブランク、または値 -1 を含む場合、IBM NameWorks は名エレメントを分類して、その分類の結果を名の国/地域別情報コードとして使用します。このフィールドに無効値が含まれている場合、エラー状態が報告されます。

オリジナルの姓

オリジナルの姓エレメント。このフィールドには、組織名の元のフルネームが入ります。

オリジナルの名

オリジナルの名エレメント。

代替解析のフラグ

その前のフィールドによって判別された内容にかかわらず、現在のレコードがオリジナルの名前の代替解析であることを示します (0 = いいえ、1 = はい)。

Name Preprocessor の検索リスト・ファイル:

NH 検索リスト・ファイルは、非重複化された名前情報を含む Distributed Search への入力です。

検索リスト、オリジナル・データ、および NH オリジナル索引ファイルは、暫定データ・ファイルが小さな断片に分割される非重複化プロセスの結果です。暫定ファイルの各分割によって、次の 3 つのファイルが生成されます。

検索リスト・ファイルは、コンマ区切りファイルで、固有の名前モードが有効 (npp 構成ファイルで createOrig=true) などのとき、次の 2 つの例外を除いて「暫定データ」ファイルと同じです。

- 「補足データ」フィールドに索引値が含まれ、これは固有の名前と一致するすべてのレコードを見つけるために使用されます。
- 追加の値 (numOrig) が最後のフィールドとして存在し、固有の名前と関連付けられている元のレコードの数を示します。

補足データ

通常は、1 つ以上のデータベースで名前レコードに関するその他の情報を取得するのに使用できるレコード ID。ただし、このフィールドは、名前レコードに関する追加情報を含むことができます。

カテゴリー

名前レコードが、個人 (P) 名か、組織 (O) 名か、それともその両方 (B) として分類される必要があるかを示す単一の文字。このフィールドが空の場合

合、名前が IBM NameWorks によって分類されてからその後の処理が適用されます。無効な文字を指定すると、エラー条件が発生します。

スクリプト・タイプ

名前テキストの文字タイプを示します。1 から 8 までの整数で特定の文字タイプを示し、0 で特定の文字タイプがないことを示します。

姓または氏名

名前レコードに 3 つのフィールドのみが含まれている場合、このフィールドは氏名として解釈されます。名前レコードが (データ・ファイルで直接、または IBM NameWorks によって間接的に) 組織名として分類された場合、追加のフィールドは無視されます。その他のインスタンスでは、名前は音訳、解析、分類、およびデータ・リストへの追加が行われます。

名 その前のフィールドが姓エレメントとして解釈された場合は名エレメントとして解釈されます。このフィールドがブランクであり、名前レコードが (データ・ファイルで直接、または IBM NameWorks によって間接的に) 組織として分類された場合、追加のフィールドは無視されます。追加のフィールドが存在しない場合、名エレメントおよび姓エレメントは音訳、解析、分類、およびデータ・リストへの追加が行われます。

姓の国/地域別情報コード

姓の国/地域別情報コード。このフィールドがブランク、または値 -1 を含む場合、IBM NameWorks は姓エレメントを分類して、その分類の結果を姓の国/地域別情報コードとして使用します。このフィールドに無効値が含まれている場合、エラー状態が報告されます。

名の国/地域別情報コード

名の国/地域別情報コード。このフィールドがブランク、または値 -1 を含む場合、IBM NameWorks は名エレメントを分類して、その分類の結果を名の国/地域別情報コードとして使用します。このフィールドに無効値が含まれている場合、エラー状態が報告されます。

オリジナルの姓

オリジナルの姓エレメント。このフィールドは、組織名の元のフルネームであると判別されます。

オリジナルの名

オリジナルの名エレメント。

代替解析のフラグ

その前のフィールドによって判別された内容にかかわらず、現在のレコードがオリジナルの名前の代替解析であることを示します (0 = いいえ、1 = はい)。

numOrig

固有の名前と関連付けられている元のレコードの数を示します。

Name Preprocessor のオリジナル・データ・ファイル:

オリジナル・データ・ファイルは、前処理の結果を表示する 2 つの追加のフィールドがあるカスタマー・データです。

検索リスト、オリジナル・データ、および NH のオリジナル索引ファイルは、暫定ファイルが小さな断片に分割される非重複化プロセスの結果です。暫定ファイルの各分割によって、次の 3 つのファイルが生成されます。

オリジナル・データ・ファイルには、次のフィールドがあります。

SN 姓 (前処理済み)。

GN 名 (前処理済み)。

custID

このレコードに関連付けられる追加データ (データのオリジナル・ソースを指す ID など)。最大長は 256 バイトです。

altParseFlag

この名前が代替名前解析を生成したか? (0 = いいえ、1 = はい)

regFlag

この名前に正規化が必要だったか? (0 = いいえ、1 = はい)

Name Preprocessor のオリジナル索引ファイル:

オリジナル索引ファイルは、NH の検索リストとオリジナル・データ・ファイルを結び付けます。

検索リスト、オリジナル・データ、および NH のオリジナル索引ファイルは、暫定ファイルが小さな断片に分割される非重複化プロセスの結果です。暫定ファイルの各分割によって、次の 3 つのファイルが生成されます。

Distributed Search は、オリジナル索引ファイルを使用して、NameHunter の結果からオリジナル名前データを取得します。オリジナル索引ファイルには、次の 2 つのフィールドがあります。

nhID NH 検索リストに格納されるときに、前処理されたユニークな各名前に割り当てられたユニークな数値 ID。

origOffset

最初のカスタマー・レコードを指すオリジナル・データ・ファイルへのオフセット。

Name Loader ユーティリティを使用した名前リストの管理:

Enterprise Name Search の名前リストの管理は、Name Loader ユーティリティを使用して行えます。Name Loader ユーティリティは、独立した Java ベースのプログラムです。これは、名前リストから未加工のソース名を取り、IBM NameWorks を使用してその名前を分析し、名前および名前分析の結果を ENS データベースに保管します。

ENS の Name Loader ユーティリティは、Distributed Search で使用される Name Preprocessor ユーティリティと機能の点で似ていますが、それによって変わるものではありません。Name Loader ユーティリティは、名前リスト・ファイルまたは外部データ・ソース (「乗客」リストや「ウォッチ・リスト」など) から名前をロードします。「ソース名」は、元の解析とスクリプトを含む、最初に受け取った名前を指します。検索の結果はその形式で表示されます。

ソース名は必要に応じて、IBM NameWorks によって分析、解析、および音訳されます。この結果として生成される名前を、「検索」名と呼びます。検索名は、内部形式 (大文字のローマ字) で ENS データベースに保管され、検索で使用するために IBM NameWorks にロードされます。ENS は、両方の形式をデータベース表に保持し、その間のマッピングを管理します。

さらに、Name Loader ユーティリティーを使用して名前を再ロードまたは削除することもできます。Name Loader を使用して名前をリストからロード、再ロード、および削除する際は、検索セルをアクティブのままにすることができます。

名前リスト・ファイルからのソース名の読み取り

名前リスト・ファイルは、CSV (コンマ区切り値) 形式のフラット・テキスト・ファイルです。名前リストの各行は、1 つのソース名を表します。CSV ファイルには、各行に関する以下の情報が含まれます。

表 10. 入力名に関する CSV ファイル内のフィールド

フィールド	意味	例	デフォルト	備考
0	姓	"SMITH"	空白	姓または名のいずれかを空白にすることができます。ただし、両方を空白にすることはできません。
1	名	"JOHN"	空白	姓または名のいずれかを空白にすることができます。ただし、両方を空白にすることはできません。
2	idData	300011	デフォルトなし	必須。データ・ソース内で固有でなければならない ID。
3	姓の国/ 地域別情報	3	空白	オプション
4	名の国/ 地域別情報	3	空白	オプション
5	カテゴリ	0	空白	オプション。名前のタイプ (個人名、組織名、または指定なし) を示します。

Name Loader Analyzer コンポーネント

Name Analyzer コンポーネントは、各入力ソース名に対して以下の分析を実行します。

- 名前を音訳することにより、各ソース名を処理する。
- フラット CSV ファイルの「カテゴリ」要素が空白の場合に、名前を名前タイプにカテゴリ化する。
- 「カテゴリ」が「個人」の場合に、名前の元の形式に対して最適な解析を見つける。
- ソース名に基づいて、音訳された 1 つ以上の検索名を見つける。

ソース名と検索名

ソース名 とは、名前リストから最初にロードされる名前のことです。一方、検索名は、IBM NameWorks によって実行される分析の結果として生成される名前を指します。スキーマは、ソース名と検索名の両方を保管します。

ソース名は ENS_SOURCE_NAME テーブルに置かれます。検索名は ENS_SEARCH_NAME テーブルに置かれます。Enterprise Name Search は、その 2 つの間のマッピングを管理します。パフォーマンス上の理由により、名前をスキーマ・テーブルに書き込む際には複数のスレッドを使って書き込みを行います。各スレッドは、バッチ・データベース更新を使用して複数のレコードを追加または更新します。スレッドの数およびバッチ・サイズは、Name Loader ユーティリティーの構成ファイルで構成することができます。

2 つの検索名が同一でも国/地域別情報が異なる場合、それらの検索名はそれぞれ別々に保管されます。例えば、韓国語の国/地域別情報が指定されている「June Park」は、英語の国/地域別情報が指定されている「June Park」とは別に保管されます。一致規則がこれら 2 つの国/地域別情報で異なっているからです。

2 つの検索名が同一で代替解析フラグが異なる場合も、それらの検索名はそれぞれ別々に保管されます。例えば、ソース名「Elton John」はソース名「John Elton」と異なります。「Elton John」の名前検索を要求すると両方のソース名が検出されて報告されますが、代替解析に基づいているため、ソース名 John Elton はより低い一致スコアで報告されます。

loader.config ファイル

Name Loader ユーティリティーには loader.config という名前の独自の構成ファイルがあります。これは、Enterprise Name Search で使用される IBM NameWorks インスタンスの動作を指定します。Name Loader ユーティリティーに関するすべてのオプションと構成の詳細が、このファイルで指定されます。

Name Preprocessor 構成ファイル

Name Preprocessor の入力、ステップ、および出力は、構成ファイル npp.config によって制御されます。このファイルは、標準的な Microsoft Windows の .ini ファイルの形式になっています。

入力名の NameCategory は、名前がどのデータ・リスト (組織または個人) に追加されるかを判別します。NameCategory が指定されていない場合、Name Preprocessor は NameSifter を使用して NameCategory を判別します。NameCategory が不明であり、ふるい分けのオプションが false に設定されている (doCategorize=false) 場合、Name Preprocessor は、エンティティー関連付けのない名前を個人名として処理します。

外部トークン・ファイルを指定するには、カスタム解析トークンを npp.config ファイルの [Custom Tokens] セクションに追加します。

npp.config ファイルのサンプル

```
[npp]
inFile=names.txt
interimFile=names.npp.txt
```

```

nppOutFile=names.npp.out
origDataFile=names.orig.dat
origIndexFile=names.orig.idx

sifterRulesFile=SifterRules.ibm

arabicTransFile=arabicTransRule.ibm
cyrillicTransFile=cyrillicTransRule.ibm
greekTransFile=greekTransRule.ibm
koreanTransFile=koreanTransRule.ibm
latinTransFile=latinTransRule.ibm
japaneseTransFile=japaneseTransRule.ibm
chineseTransFile=chineseTransRule.ibm

reportIncrement=100000
doFullName=true
doCategorize=false
doTransliterate=true
doParse=true
doClassify=true
doNhClean=true
parseThreshold=0.5

maxRecsPerSplit=2000000
createNpp=true
createNh=true
numNppOutFiles=2
deleteTempFiles=true
deleteNppFile=false
createOrig=true

logDebug=cout
logError=cout
logEvent=cout

maxGnCacheSize=4000000
maxOnCacheSize=0
maxSnCacheSize=4000000

```

直接ロードされる **Distributed Search** プロセス

名前の前処理がバイパスされるように、前処理されていない名前データを使用して Distributed Search プロセスを直接ロードできます。この場合、NH の検索リストのみが処理に必要です。Name Preprocessor が作成するオリジナル・データおよび NH のオリジナル索引は使用されません。

このモードでは、NH 検索リスト・ファイルは以下の形式になります。これらの CSV 形式のファイルのすべてのフィールドは、引用符で囲まれたフィールドをサポートし、これにより入力ファイルでコンマを使用できます。

個人名

```
ID,category,SN,GN,cultureSN,cultureGN,origSN,origGN,altParse,0,count
```

組織名

```
ID,category,ON,origON,altParse,0,count
```

ID 前処理されていないユニークな各名前に割り当てられた、ユニークな数値 ID

category

名前のカテゴリー。P (個人名) または O (組織名)

SN 姓
GN 名
ON 組織名

cultureSN
姓の国/地域別情報コード

cultureGN
名の国/地域別情報コード

origSN
オリジナルの姓

origGN
オリジナルの名

altParse
名前が代替名前解析から生成されたかどうかを示します (0 = いいえ、1 = はい)。名前が代替解析であることがわかっている場合は、値 1 を指定します。

count この ID に対して参照しているレコード数を示します。全検索モードで実行している場合、各 ID が 1 つのレコードに結び付けられるので、カウントはゼロになります。固有の名前モードで実行している場合、各レコードを複数の ID に参照付けることができるので、カウントはゼロよりも大きくなります。

Name Preprocessor の実行

Name Preprocessor をコマンド・プロンプトから実行できます。

始める前に

ファイル名がコマンド行で指定されていない場合、Name Preprocessor は起動時に現行ディレクトリーのデフォルトの構成ファイル `npp.config` を開きます。通常、このディレクトリーから Name Preprocessor が呼び出されます。例:

```
C:> npp
```

Name Preprocessor を実行するときに、コマンド行で別のディレクトリーの構成ファイルを指定できます。例:

```
C:> npp /usr/GNR/data/npp.config
```

Name Preprocessor 構成の考慮事項

Name Preprocessor の構成方法を決定するには、使用するデータおよびコンピューティング環境で、何か最も適切に動作するかを調べるための考慮事項およびいくつかの実験が必要です。

データに対してどのようなタスクを実行するかに応じて、次のようなさまざまな考慮事項があります。

音訳 名前を基本的なラテン・アルファベットの 26 文字のみを使用するように変換します。音訳では、アクセントが付いたラテン・アルファベットからアクセントを削除し、アラビア語、中国語、キリル文字、ギリシャ語、仮名 (日本語の平仮名と片仮名)、およびハングル (韓国語) の文字で記述された名前

を、ラテン・アルファベットの文字を使用した対応する形式に変換します。処理されるデータが既に基本的な 26 個のラテン・アルファベットを使用している場合、音訳を使用不可にして、前処理の時間を短縮できます。

解析 単一の名前を、名 (GN) コンポーネントと姓 (SN) コンポーネントに分割します。解析では、NameParser の信頼性がしきい値より小さい場合に、名前の追加の解析を生成します。通常、名前データを解析すると、名前の数が 10% 増えます。既に GN と SN に分割されている名前を指定することによって、解析を使用不可にして前処理の時間を節約できます。

分類 姓および名に国/地域別情報コードを割り当てるには、NameClassifier を実行します。名前を分類しないままにしておくと、検索時間が遅くなり、正規化が使用不可になるため、分類を完了しておくことを強くお勧めします。

正規化

IBM の正規化規則がその国/地域別情報に対して存在する場合は、入力名の正規形を生成します。正規化は、検索の品質を大幅に向上しますが、入力名が多くなるため、検索プロセスが遅くなります。

NhClean

NameHunter のクリーン関数は、スペースおよび大文字の A から Z を除くすべての文字を削除します。クリーニングは高速で、重複する項目の数を増やすため、Distributed Search がロードする必要があるレコードの数が減少します。

サイジング

同時に実行する Searcher の数は、名前の数、使用可能にしているオプション、および利用できるプロセッサ数の関数になります。選択しているオプションによって、基本的なパフォーマンスが決定され、必要に応じて使用する Searcher の数に影響を与えます。次の例は、サイジングの考慮事項の説明に役立ちます。

- 50,000,000 個の入力名で開始します。
- 解析により 5,000,000 個の名前が追加され、合計が 55,000,000 個になります。
- 正規化により 20,000,000 個の名前が追加され、新しい合計が 75,000,000 になります。
- Name Preprocessor の非重複化によって、固有の名前の数が 50,000,000 個に減らされます。これは、標準的な減少ですが、これより大きくも小さくもなります。
- Distributed Search で利用できるプロセッサが 8 個あるとします。これらのうちの 1 つは追加用に予約されているため、50,000,000 個の固有の名前を 7 つに分割でき、1 つのプロセッサにつき約 7,000,000 個の名前が割り当てられます。このため、構成項目 **numNppOutFiles** を 7 に設定します。

numNppOutFiles 構成オプションを設定することによって、Name Preprocessor を使用してデータ・リストを必要な数のサブリストに分割できます。

Distributed Search のパフォーマンスおよび構成の概要

このセクションでは、Distributed Search の構成、調整、および実行方法に関する情報を提供します。

Distributed Search は、メモリーとプロセッサに高負荷をかけます。 Distributed Search プロセスは、要求を処理する際にコンピューターの処理能力を 100% 消費します。このため、Distributed Search を少なくとも検索プロセスと同じ数のプロセッサを使用するよう構成してください。コミュニケーション・マネージャーは、サポートする検索の数が妥当であれば、専用プロセッサを必要としません。

Distributed Search が照会を処理できる速度は、並行して処理されている検索プロセスの数に直接比例します。メッセージ管理のコストがますます重要になってきているため、検索の数が増え、各検索によってサポートされるデータのサイズが小さくなると、パフォーマンス上の利点は小さくなります。

Distributed Search のメモリー

各 Distributed Search プロセスは、すべての名前データをメモリーにロードする必要があります。検索プロセスがディスクにページングする必要がある場合、パフォーマンスが低下します。

各検索プロセスによって消費されるメモリーの量の概算は、入力名の平均の長さ、正規化を使用するかどうか、および Name Preprocessingによってどのように名前が前処理されたかなど、多くの要因によって影響されます。多くの場合、Searcher にロードされる名前の量がわかっていると、次の経験法則を使用して必要なメモリーを見積もることができます。

$$100 \text{ MB} + \text{NumberOfNames} * 300 = \text{memory required}$$

100 MB は、NameHunter サポート・ファイルをすべてロードする場合に必要なメモリー量です。300 バイトは、名前ごとに必要とされるストレージを控えめに見積もった場合の値です。このため、1000 万個の名前をロードしている場合、必要なメモリーは次のようになります。

$$100 \text{ MB} + 10,000,000 * 300 = 3.1 \text{ GB}$$

名前データに必要なメモリーに加えて、NameHunter が検索中に一致候補を格納するためのメモリーも必要になります。1 回の検索で、一致しきい値を上回るスコアで検出された一致候補ごとに、最大 1000 バイトが必要です。上位 50 個のみを返す場合でも、類似性スコアが最も高い名前を特定するためにすべての名前を検査する必要があります。40,000 以上の一致候補を追跡する必要がある場合があります。Distributed Search サーバーは現時点で単一スレッドであるため、どの時点でも、必要な一致候補のリストは 1 つだけです。検索結果の収集のためにさらに 100 MB 以上を見込んでおくことをお勧めします。

Distributed Search のトランザクション速度

100 万個の名前を超えるサイズでは、検索の実行にかかる時間の方が、XML メッセージの管理にかかる時間より重要です。プロセッサおよびメモリーの割り振りが十分だと想定した場合、1 つの Distributed Search プロセスが、100 万個の名前に対して 1 秒当たり 10 個の照会を実行できます。この式は、線形的に変化します。

1000 万個の名前の場合、1 つの検索プロセスは、1 秒当たり 1 個の照会を実行できます。システム・パフォーマンスは、データの構成および処理能力に応じて異なります。

例えば、1 億個の名前の名前リストは、名前の前処理の後に 6400 万個の固有の名前に減少します。8 個のプロセッサを搭載したサーバーでは、このリストを 800 万個の名前のサブセットに分割できます。この設定によって、Distribute Search は、1 秒当たり 1 個より多い照会をサポートできます。プロセッサが 16 個ある場合、サブセットは 400 万個の名前まで減少し、トランザクション速度は 1 秒当たり 2 個の照会より少し良くなります。

Distributed Search の構成ファイルおよび設定

Distributed Search は、通信を管理する 1 つのプロセス (commgr)、および検索と更新を実行する 1 つ以上のプロセス (Searcher) から構成されています。Searcher はそれぞれランタイム言語サポート・ファイルのセットへのアクセスを必要とし、それぞれが名前データの各部分を見つけられる場所を認識している必要があります。Distributed Search に付属のデフォルトの構成ファイルの名前は、ds.config です。このファイルには、新しいセッションが開始されるときに各 Searcher プロセスが必要とする構成設定が含まれます。これらの設定は、すべての Searcher によって共有されるものと、Searcher プロセスの特定のインスタンスのみによって使用されるものがあります。

ds.config ファイルのサンプル

デフォルトの構成ファイルのデフォルトの内容を次に示します。これらの設定は、3 つの検索プロセスがある Distributed Search システムを構成します。最初の 2 つのプロセスは、オリジナル・データ・ファイルの部分を共有し、3 つ目はデータなしで実行しますが、追加を処理するよう構成されています。

ファイル名の項目は、パスとともに完全修飾ファイル名を指定した (例: %user%GNR\data%names.txt) 場合を除き、ファイルが現在のディレクトリーにあることを想定しています。ブール値の項目はすべて、次の入力を true として受け入れます。

- true または t
- yes または y

項目は大/小文字の区別がなく、その他の値は false と見なされます。

```
[commgr]
listenPort=2345
ipv6=
sleepMsec=10
waitConnectSec=5
heartbeatSec=60
msgBuffSize=1000000
logDebug=
logError=cout
logEvent=cout
logMessage=
numSearchers=3

[searcherCommon]
compparmsDefaults=compparms.config
```

```
ibmTaqFile=taq.ibm
ibmGnvFile=gnv.ibm
ibmSnvFile=snv.ibm
ibmOnvFile=onv.ibm
ibmOnTermFile=terms.ibm

custTaqFile=
custGnvFile=
custSnvFile=
custOnvFile=
custOnTermFile=

arabicTransFile=arabicTransRule.ibm
cyrillicTransFile=cyrillicTransRule.ibm
greekTransFile=greekTransRule.ibm
koreanTransFile=koreanTransRule.ibm
latinTransFile=latinTransRule.ibm
japaneseTransFile=japaneseTransRule.ibm

afghaniRegFile=swasianRegRule.ibm
angloRegFile=angloRegRule.ibm
arabicRegFile=swasianRegRule.ibm
chineseRegFile=chineseRegRule.ibm
europeanRegFile=
farsiRegFile=farsiRegRule.ibm
frenchRegFile=frenchRegRule.ibm
genericRegFile=angloRegRule.ibm
germanRegFile=germanRegRule.ibm
hanRegFile=
hispanicRegFile=hispanicRegRule.ibm
indianRegFile=indianRegRule.ibm
indonesianRegFile=indoRegRule.ibm
japaneseRegFile=japaneseRegRule.ibm
koreanRegFile=koreanRegRule.ibm
pakistaniRegFile=swasianRegRule.ibm
russianRegFile=russianRegRule.ibm
southwestAsianRegFile=swasianRegRule.ibm
thaiRegFile=thaiRegRule.ibm
vietnameseRegFile=
yorubanRegFile=

genericOnRegFile=genericOnRegRule.ibm

isUnique=true

doTransliterate=true
doRegularize=true
doOnToPNListSearch=false
doCompressedBitsig=true

defaultMaxResults=100
defaultAltScoreFactor=0.97

numRecords=100000
reportIncrement=100000

allowFnuLnu=false
allowFnuInit=false
allowInitLnu=false
allowInitInit=false

[searcher1]
hostname=localhost
port=2346
ipv6=
doAdds=false
nameFile=names.nh.txt.1
```

```
origDataFile=names.orig.dat.1
origIndexFile=names.orig.idx.1
logDebug=
logError=cout
logEvent=cout
logMessage=
```

```
[searcher2]
hostname=localhost
port=2347
ipv6=
doAdds=false
nameFile=names.nh.txt.2
origDataFile=names.orig.dat.2
origIndexFile=names.orig.idx.2
logDebug=
logError=cout
logEvent=cout
logMessage=
```

```
[searcher3]
hostname=localhost
port=2348
ipv6=
doAdds=true
nameFile=
origDataFile=
origIndexFile=
logDebug=
logError=cout
logEvent=cout
logMessage=
```

Distributed Search のコミュニケーション・マネージャー用構成オプション:

Distributed Search のコミュニケーション・マネージャー・プロセスに対して、いくつかの構成オプションが存在します。

以下の構成オプションによって、コミュニケーション・マネージャー (commgr) が制御されます。

listenPort

Distributed Server と接続するためにクライアントによって使用される TCP/IP ポート。

ipv6 呼び出しプロセスとの通信に IPv6 プロトコルを使用すべきかどうかを示すオプション項目。この設定は、NameWorks 構成データなどにある、呼び出しプロセスで使用されるものと一致する必要があります。存在しないか、無効になっている場合は、IPv4 プロトコルが使用されます。

sleepMsec

メッセージ処理のループの間でコミュニケーション・マネージャーがスリープする時間数 (ミリ秒)。デフォルトで適切に動作しますが、10 またはそれ以上の任意の値が受け入れられます。

waitConnectSec

commgr による検索プロセスへの接続の試行間の間隔。

heartBeatSec

commgr と検索プロセスの間のハートビート・メッセージ間の秒数。ハートビート・メッセージは、非アクティブの期間中に接続を最新の状態に保ちます。

msgBuffSize

メッセージの送受信に使用される TCP/IP バッファのサイズ (バイト)。デフォルトの 1,000,000 は、十分なサイズより大きくなっています。

logDebug、logError、logEvent、logMessage

Distributed Search には、構成ファイルで制御できる 4 つのログ・レベルがあります。各項目は、以下のものを認識します。

- **filename** – ログ項目を配置するファイルの名前。
- **cout、cerr、clog** – 標準出力ストリーム。
- **blank** – このログが必要でないことを Distributed Search に通知しません。

logDebug

開発中に使用されるデバッグ・メッセージを表示します。実動環境では、このログへの出力は非常に少なくなります。

logError

無効なユーザー入力、システム・エラー、および通信の問題などのエラー状態を表示します。

logEvent

開始状態、メッセージの送信および受信などの重要なイベントをログに記録します。

logMessage

すべての着信および発信メッセージをログに記録します。これにより多くの出力が生成されるため、通常の操作では使用不可にしてください。

numSearchers

実行している検索の数。

Distributed Search の共通構成オプション:

すべての Distributed Search プロセスで、いくつかのオプションが共通です。これらの共通のオプションは、[searcherCommon] セクションにあります。

以下の項目は、すべての Distributed Search プロセスの [searcherCommon] で共通です。

表 11. Distributed Search の共通オプション

項目	説明
ibmTaqFile	IBM 提供の称号、接辞、修飾子 (TAQ) ファイル (taq.ibm) の場所。
ibmGnvFile	IBM 提供の名の変形ファイル (gnv.ibm) の場所。
ibmSnvFile	IBM 提供の姓の変形ファイル (snv.ibm) の場所。
ibmOnvFile	IBM 提供の組織の変形ファイル (onv.ibm) の場所。
ibmOnTermFile	IBM 提供の用語ファイル (terms.ibm) の場所。

表 11. *Distributed Search* の共通オプション (続き)

項目	説明
custOnTermFile	カスタム用語ファイルの場所。
custTaqFile	カスタムの称号、接辞、修飾子 (TAQ) ファイルの場所。
custGnvFile	カスタムの名の変形ファイルの場所。
custSnvFile	カスタムの姓の変形ファイルの場所。
arabicTransFile	IBM のアラビア語の音訳規則ファイル (arabicTransRule.ibm) の場所。
cyrillicTransFile	IBM のキリル文字の音訳規則ファイル (cyrillicTransFile.ibm) の場所。
greekTransFile	IBM のギリシャ語の音訳規則ファイル (greekTransFile.ibm) の場所。
latinTransFile	IBM のラテン語の音訳規則ファイル (latinTransFile.ibm) の場所。
angloRegFile	IBM の英語の正規化規則ファイルの場所 (angloRegRule.ibm)。
arabicRegFile	IBM のアラビア語の正規化規則ファイルの場所 (arabicRegRule.ibm)。
chineseRegFile	IBM の中国語の正規化規則ファイルの場所 (chineseRegRule.ibm)。
farsiRegFile	IBM のペルシア語の正規化規則ファイルの場所 (farsiRegRule.ibm)。
frenchRegFile	IBM のフランス語の正規化規則ファイルの場所 (frenchRegRule.ibm)。
germanRegFile	IBM のドイツ語の正規化規則ファイルの場所 (germanRegRule.ibm)。
hispanicRegFile	IBM のヒスパニックの正規化規則ファイルの場所 (hispanicRegRule.ibm)。
indianRegFile	IBM のインド語の正規化規則ファイルの場所 (indianRegRule.ibm)。
indonesianRegFile	IBM のインドネシア語の正規化規則ファイルの場所 (indonesianRegRule.ibm)。
japaneseRegFile	IBM の日本語の正規化規則ファイルの場所 (japaneseRegRule.ibm)。
koreanRegFile	IBM の韓国語の正規化規則ファイルの場所 (koreanRegRule.ibm)。
russianRegFile	IBM のロシア語の正規化規則ファイルの場所 (russianRegRule.ibm)。
southwestasianRegFile	IBM の南西アジアの正規化規則ファイルの場所 (southwestasianRegRule.ibm)。
thaiRegFile	IBM のタイ語の正規化規則ファイルの場所 (thaiRegRule.ibm)。
genericRegFile	IBM の汎用の正規化規則ファイルの場所 (genericRegRule.ibm)。
isUnique	入力データが Name Preprocessor によって非重複化されていてオリジナル・データ・ファイルがあるかどうかを Distributed Search に通知します。
doTransliterate	入力名に対して音訳が使用可能であるかどうかを示します。
doRegularize	入力名が正規化される必要があるかどうかを示します。
defaultAltScoreFactor	名前の突き合わせに代替解析を使用するときに適用されるペナルティ係数のデフォルト値。この値は検索メッセージおよびパラメーター・メッセージによってオーバーライドできます。

表 11. *Distributed Search* の共通オプション (続き)

項目	説明
defaultMaxResults	検索要求によって返される結果のデフォルトの最大数。この値は検索およびパラメーター・メッセージによってオーバーライドできます。
allowFnuLnu	姓および名がない照会が許可されるかどうかを制御します (First Name Unknown、Last Name Unknown)。この照会によってシステムで応答が多くなりすぎるが多いため、通常はこれは false です。
allowFnuInit	単一の姓のイニシャル、およびブランクの名の照会が許可されるかどうかを制御します。この照会によってシステムで応答が多くなりすぎるが多いため、通常はこれは false です。
allowInitLnu	単一の名のイニシャル、およびブランクの姓の照会が許可されるかどうかを制御します。この照会によってシステムで応答が多くなりすぎるが多いため、通常はこれは false です。
allowInitInit	単一の名のイニシャル、および単一の姓のイニシャルの照会が許可されるかどうかを制御します。この照会によってシステムで応答が多くなりすぎるが多いため、通常はこれは false です。

Distributed Search の個別の検索に固有のオプション:

いくつかの構成オプションは、個別の検索に固有です。構成ファイルの各ヘッダーには、最後に固有の番号が付いています (例: search1)。これは、構成項目が適用される検索プロセスを示します。各検索で、以下のセクションのいずれかを指定する必要があります。

以下の項目は、個別の検索に固有です。

hostname

Searcher を実行しているマシンのホスト名。

port コミュニケーション・マネージャーとの通信に使用されるポート番号。

ipv6 コミュニケーション・マネージャーとの通信に IPv6 プロトコルを使用すべきかどうかを示すオプション項目。存在しないか、無効になっている場合は、IPv4 プロトコルが使用されます。

doAdds

この検索プロセスは追加要求に応答するか? 1 つの検索のみでこの項目セットを true に設定する必要があります。

nameFile

前処理される名前データのロケーション。理想的には、これは Name Preprocessor によって生成されます。

origDataFile

nameFile に関連付けられた、Name Preprocessor からのオリジナル・データ・ファイルの出力。

origIndexFile

origDataFile に関連付けられた、Name Preprocessor からのオリジナル・データ索引の出力。

logDebug、logError、logEvent、logMessage

Distributed Search には、構成ファイルで制御できる 4 つのログ・レベルがあります。各項目は、以下のものを認識します。

- **filename** – ログ項目を配置するファイルの名前。
- **cout**、**cerr**、**clog** – 標準出力ストリーム。
- **blank** – このログが必要でないことを Distributed Search に通知しません。

logDebug

開発中に使用されるデバッグ・メッセージを表示します。実動環境では、このログへの出力は非常に少なくなります。

logError

無効なユーザー入力、システム・エラー、および通信の問題などのエラー状態を表示します。

logEvent

開始状態、メッセージの送信および受信などの重要なイベントをログに記録します。

logMessage

すべての着信および発信メッセージをログに記録します。これにより多くの出力が生成されるため、通常の操作では使用不可にしてください。

Distributed Search の実行

Distributed Search を実行する前に、操作する名前データを Name Preprocessor で処理できます。そのプロセスの一部として、オリジナル名前データ・ファイルが必要な数の部分に分割できます。または、Name Preprocessor を使用しないでオリジナル名前データ・ファイルを手動で分割することもできます。各部分は、パフォーマンスと応答時間の最適化のために、Distributed Search プロセスの専用インスタンス、および専用プロセッサが必要です。

始める前に

オリジナル・データ・ファイルが Name Preprocessor によって分割、または手動で分割された後で、Distributed Search サーバー・セッションの開始後の名前データ・リストへの名前の追加をサポートする必要があるかどうかを判別する必要があります。オリジナル名前データ・ファイルの手動による分割が実行された場合、新しい名前 (複数可) を部分の 1 つに追加してから、Distributed Search サーバーを再始動できます。通常、名前は Distributed Search セッションを終了せずに追加メッセージを使用して追加する必要があります。この場合、検索の 1 つのインスタンスを追加トランザクションのために構成する必要があり、この追加インスタンスに対して別個のプロセッサを割り振る必要があります。これらのリソースおよびランタイム・サポートの決定は、Distributed Search 構成ファイルに設定として登録されます。

このタスクについて

Distributed Search を実行するには、以下のステップを実行します。

手順

1. 検索 ID および (オプションで) 構成ファイルを指定することにより、Distributed Search プロセスの構成済みの各インスタンスを開始します。検索 ID は、Distributed Search 構成ファイルのヘッダーの数値に対応します。例えば、1 は [searcher1] の構成を使用するため、構成ファイルのヘッダーのセクションは、searcher 1 ds.config を読み取ります。
2. コマンド行からコミュニケーション・マネージャーを開始して、(オプションで) 構成ファイルを指定します。例: commgr ds.config。

次のタスク

すべての検索プロセスは、コミュニケーション・マネージャー (commgr) の開始前にロードおよび初期化を完了しておく必要があります。次のコンソール・メッセージが表示されたら、検索プロセスの準備ができています。

```
waiting for connection on port <ポート番号>
```

検索プロセスの場合と同様、構成ファイルの引数はオプションであり、指定されない場合は、デフォルトで現行ディレクトリーの ds.config に設定されます。コミュニケーション・マネージャーが 1 つ以上の Searcher との接続に失敗した場合、メッセージの送信を続行し、そのメッセージは次の形式でコンソール・デバイスに送付されます。

```
06:25:53 could not connect to server - localhost:2348
```

commgr は、関連するすべての検索プロセスが正常に接続するまで、ユーザーの接続要求を拒否します。構成されたすべての検索プロセスに接続した後で、commgr は次の形式でコンソール・デバイスにメッセージを発行します。

```
06:25:54 connected to all searchers, waiting for requests
```

(ログ・イベントやエラーなど) デフォルトのロギング・オプションを使用して構成したとき、Distributed Search プロセスはセッションの初期化の際に次の出力をコンソールにプリントします。

```
>searcher 1
11:13:26
searcher1.hostname ----- localhost
searcher1.port ----- 2346
searcher1.doAdds ----- 0
searcher1.isUnique----- 1
searcher1.doRegularize----- 1
searcher1.doTransliterate---- 0
searcher1.nameFile ----- names1m.nh.txt.1
searcher1.numRecords----- 400000
searcher1.reportIncrement---- 100000
searcher1.defaultMaxResults-- 100
searcher1.ibmTaqFile----- taq.ibm
searcher1.ibmGnvFile----- gnv.ibm
searcher1.ibmSnvFile----- snv.ibm
searcher1.ibmBnvFile----- bnv.ibm
searcher1.custTaqFile-----
searcher1.custGnvFile-----
searcher1.custSnvFile-----
searcher1.arabicTransFile --- arabicTransRule.ibm
searcher1.cyrillicTransFile - cyrillicTransRule.ibm
searcher1.greekTransFile ---- greekTransRule.ibm
searcher1.latinTransFile ---- latinTransRule.ibm
searcher1.angloRegFile ----- angloRegRule.ibm
```

```

searcher1.arabicRegFile ----- swAsianRegRule.ibm
searcher1.germanRegFile -----
searcher1.indianRegFile -----
searcher1.russianRegFile ----
searcher1.thaiRegFile -----
searcher1.origDataFile----- names1m.orig.dat.1
searcher1.origIndexFile----- names1m.orig.idx.1
searcher1.logDebug -----
searcher1.logError ----- cout
searcher1.logEvent ----- cout
searcher1.logMessage -----
11:13:26 loading taqs from taq.ibm
11:13:26 loading gnv from gnv.ibm
11:13:30 loading snv from snv.ibm
11:13:33 loading Arabic trans file from arabicTransRule.ibm
11:13:35 loading Cyrillic trans file from cyrillicTransRule.ibm
11:13:35 loading Greek trans file from greekTransRule.ibm
11:13:35 loading Latin trans file from latinTransRule.ibm
11:13:35 loading Anglo reg file from angloRegRule.ibm
11:13:35 loading Arabic reg file from swAsianRegRule.ibm
11:13:35 loading names from names1m.nh.txt.1
11:13:35 reserving space for 400000 records
11:13:38 -- num loaded = 100000
11:13:42 -- num loaded = 200000
11:13:46 -- num loaded = 300000
11:13:51 -- num loaded = 400000
11:13:56 -- num loaded = 500000
11:13:57 numLoaded      = 533025
11:13:57 loading orig data
num read = 100000
num read = 200000
num read = 300000
num read = 400000
num read = 500000
11:13:58 waiting for a connection on port 2346
11:13:58 connected, waiting for data...

```

次の例は、commgr のセッションの正常な開始からのコンソールの出力がどのようなものかを示したものです。

```

>commgr
11:13:58
commgr.listenPort ----- 2345
commgr.sleepMsec ----- 10
commgr.logDebug -----
commgr.logError ----- cout
commgr.logEvent ----- cout
commgr.logMessage -----
commgr.numSearchers ----- 3
11:13:58 listening on port, 2345
11:13:58 connected to searcher localhost:2346
11:14:14 could not connect to server - localhost:2347
11:14:15 could not connect to server - localhost:2347
11:14:15 connected to searcher localhost:2347
11:14:15 connected to searcher localhost:2348
11:14:15 waiting for requests...

```

Distributed Search の構成オプションおよび考慮事項

Distributed Search は、さまざまな種類の運用設定で動作するように設計されています。

Distributed Search は、32 個または 64 個のプロセッサを持つ専用サーバーと同様の容易さで、シングル・プロセッサ・マシンで運用できます。これは、単純化されたメッセージング・システム (TCP/IP 接続) および利用可能なプロセッサに

検索を拡大する機能のためです。重複した名前を取り除くことと、元の名前データにそれらを残しておくことの間で選択することは、Distributed Search のパフォーマンスおよび検索結果の品質に重要な影響を与えます。最適な出力は、元の名前データ・レコード自体の性質および特性に左右されるため、そのような選択に対する一貫性のあるベスト・プラクティスはありません。

手動で処理するか、または Name Preprocessor ユーティリティを使用して元の名前データ・レコードの重複を取り除く場合、各検索要求の動作で、候補名数が少なくなり、検索の応答時間およびトランザクションのスループット・レベルがそれに対応して向上します。Name Preprocessor は、データのクリーニングおよび重複の除去によって、1 億の名前が含まれている名前リストを、約 3300 万の名前まで減らすことができます。

クレンジングおよび重複除外プロセスによって達成される減少の程度は、名前データごとに異なります。データ・リストには、重複名を多く含むものがあれば、固有の名前を多く含むものもあります。ノイズ、形式の不整合、およびその他のランダムまたは国/地域別情報に基づく要因も、クレンジングおよび重複除外された名前リストの最終的なサイズに影響を与えます。

検索時間の短縮は、各検索プロセスの後にオリジナル・データ要求トランザクションがいくつか続く必要があると、相殺されます。このトランザクションは、重複除外プロセス中に割り当てられ、内部生成された固有の名前 ID と、オリジナル名前データからの 1 つ以上の実際の名前 ID のセットを交換します。ただし、大幅に数が減らされた名前の検索で短縮された時間は、オリジナル・データ要求トランザクションによって発生したオーバーヘッドの影響を緩和します。このオーバーヘッドまたは重複する名前の特定および削除に必要なプリプロセッシング時間が、許容できる制限を超えた場合、Distributed Search は、手動または Name Preprocessor のいずれかで小さなセグメントに分割されたオリジナル名前データを処理できますが、これにより並列化が増加し、各検索要求に対して使用するプロセッサのリソースを増やすことができます。

前処理された名前レコードによって、共通名に対する検索結果を大幅に改善することもできます。検索要求に共通名が含まれる場合、正確な名前の一致により、わずかにスペルが異なる有効で役立つ可能性がある名前の一致が削減されます。名前の大きなデータ・リストが Distributed Search によって検索されている場合は常に、多数の名前の一致が検索結果から無視されないようにするために、この結果を考慮する必要があります。

NameHunter Distributed Search XML インターフェース

Distributed Search は、XML インターフェースを使用して、名前データ・リストを操作して検索します。すべてのメッセージは、クライアントからの要求と Distributed Search からの応答で構成されています。

以下の XML 要求および応答の例には、コードのサンプルと、キー・エレメントの記述がある関連付けられた応答が含まれています。エレメントのいくつかは、情報提供のみを目的としており、説明されていません。例えば、次のようなバージョン情報は、すべてのメッセージ・ヘッダーにあります。

```
<NHServerMessage protocol_version="4.1">
```

Search Results 応答を例外として、Distributed Search からの応答は、関連付けられた要求が正常に処理されたことの単純な確認応答です。

Distributed Search とクライアント・アプリケーションの間の要求はすべて、NULL で終了する必要があります。

Distributed Search XML 要求

Distributed Search XML 要求によって、名前データ・リストを操作して検索できます。すべての要求は、要求の出力を記述した Distributed Search からの応答を受け取ります。

追加要求:

追加要求を使用すると、名前をメモリーに常駐する検索可能な名前に追加するよう Distributed Search に命令できます。正常に完了すると、関連付けられた追加要求メッセージに示された要求 ID 値を含む応答メッセージが返され、クライアント・プロセスが要求と応答のペアを作ることができます。

名前付きエンティティ・カテゴリー (NEC) フィールドは、名前のタイプと、Distributed Search の検索リストを更新するときどのフィールドを使用するかを決定します。

- O – 組織名が NAME フィールドを使用します。
- P – 個人名が SN フィールドおよび GN フィールドを使用します。
- A – 3 つのフィールドすべてが使用され、2 つの名前が追加されます。

追加要求のサンプル

```
< NHServerMessage protocol_version="4.1">
  <BASIC_REQUEST_INFO request_type="A" request_id="-1"/>
  <NAME_TO_ADD
    SN="Acme Light Industries, Inc."
    GN=""
    NAME="Acme Light Industries, Inc."
    NEC="A"
    NAME_ID="123456"
    CULTURE_SN="4"
    CULTURE_GN="4"
    ALT_PARSE="N" />
  <DATA_LIST_NAME value="" />
</ NHServerMessage>
```

BASIC_REQUEST_INFO

属性	必須?	制限
request_type	はい	「A」である必要があります。
request_id	いいえ	Distributed Search が応答で戻す数値。Distributed Search が数値を生成するには、-1 を使用します。

NAME_TO_ADD

属性	必須?	制限
SN	はい ¹	姓。最大 128 文字。
GN	はい ¹	名。最大 128 文字。
NAME	はい ²	名前。最大 128 文字。組織名などのフィールド化されていない名前の保持に使用されません。
NEC	はい	名前付きエンティティ・カテゴリ (NEC)。有効値は以下のとおりです。 O = 組織名 P = 個人名 A = すべて
NAME_ID	はい	追加される名前の ID。最大 256 文字の英数字。ID はユニークである必要はありません。
CULTURE_SN	いいえ	姓の国/地域別情報コード。
CULTURE_GN	いいえ	名の国/地域別情報コード。
ALT_PARSE	いいえ	名前の代替解析を実行するかどうか。(Y または N)

¹NEC が P または A のときにのみ必須。

²NEC が O または A のときにのみ必須。

DATA_LIST_NAME

属性	必須?	制限
value	はい	Distributed Server では現在実装されていません。

削除要求:

削除要求を使用すると、Distributed Server から名前 (複数可) を削除できます。この要求は、指定されたお客様提供の Name_ID を持つ Distributed Server の各レコードを削除します。正常に完了した場合は、送信者のオリジナル要求 ID を含む成功の応答メッセージが返されます。

削除要求のサンプル

```
<NHServerMessage protocol_version="4.2">
  <BASIC_REQUEST_INFO request_type="D" request_id="-1"/>
  <RECORD_ID_TO_DELETE value="123456"/>
  <DATA_LIST_NAME value=""/>
</NHServerMessage>
```

BASIC_REQUEST_INFO

属性	必須?	制限
request_type	はい	D である必要があります。
request_id	いいえ	Distributed Search が応答で戻す数値。Distributed Search が数値を生成するようにするには、-1 を使用します。

RECORD_ID_TO_DELETE

属性	必須?	制限
value	はい	削除対象のお客様指定の Name_ID。この Name_ID のすべての名前が削除されます。

DATA_LIST_NAME

属性	必須?	制限
value	いいえ	Distributed Search では現在実装されていません。

オリジナル・データ要求:

オリジナル・データ要求を使用すると、このマッチングに関連付けられている名前のオリジナル・データ・レコードをすべて送信するよう Distributed Search に命令できます。この要求は、Name Preprocessor を使用して名前データを非重複化する場合にのみ必要です。正常に完了した場合は、送信者のオリジナル要求 ID を含む検索結果メッセージが返されます。

オリジナル・データ要求のサンプル

```
<NHServerMessage protocol_version="4.2">
  <BASIC_REQUEST_INFO request_type="G" request_id="-1"/>
  <DATA_LIST_NAME value=""/>
  <NAME_ID value="123456"/>
</NHServerMessage>
```

BASIC_REQUEST _INFO

属性	必須?	制限
request_type	はい	G である必要があります。
request_id	いいえ	Distributed Search が応答で戻す数値。Distributed Search が数値を生成するようにするには、-1 を使用します。

DATA_LIST_NAME

属性	必須?	制限
value	はい	Distributed Server では現在実装されていません。

NAME_ID

属性	必須?	制限
value	はい	オリジナル・データを取得する対象となる名前の NameHunter ID。この値は、オリジナル名前リストが Name Preprocessor によって重複除外された場合に、検索結果メッセージ内に返される NAME_ID です。

検索要求:

検索要求を使用すると、名前を検索するよう Distributed Search に命令できます。正常に完了した場合は、送信者のオリジナル要求 ID を含む検索結果の応答が返されます。

オプションで、検索要求は、要求された検索に対して 1 つ以上のパラメーター設定を入力するよう指定する場合があります。検索要求を使用すると、パラメーター・グループ (GENERAL_PARMS、COMP_PARMS_GN、COMP_PARMS_SN、および COMP_PARMS_ON) それぞれにおいて、いずれかまたはすべての設定を指定できます。比較パラメーター (CompParm) のオーバーライドおよび調整が、検索名の GnCulture、SnCulture、または OnCulture フィールドに適用されます。国/地域別情報が指定されていない場合は、オーバーライドおよび調整がデフォルトの国/地域別情報 (CultureUnknown) に適用されます。

名前付きエンティティ・カテゴリー (NEC) フィールドは、照会名のタイプ、および検索を実行するときどのフィールドを使用するかを決定します。

- O – 組織名が NAME フィールドを使用します。
- P – 個人名が SN フィールドおよび GN フィールドを使用します。

検索要求のデフォルトの NEC は、P (個人名) です。NEC の A (すべて) は検索要求では許可されておらず、IBM NameWorks によって P にマップされます。

次の例は、パラメーターなしで組織名を使用する標準的な検索を示したものです。

検索要求のサンプル (パラメーターの指定なし)

```
<NHServerMessage protocol_version="4.1">
  <BASIC_REQUEST_INFO request_type="S" request_id="-1"/>
  <DATA_LIST_NAME value="data_list_1"/>
  <SEARCH_NAME NAME="Acme Light Industries, Inc."
    NEC="0"
    Srch Opt=3/>
</NHServerMessage>
```

次の検索要求には、組織名の調整係数が含まれています。調整係数は、パラメータに `_ADJ` を追加して、調整に有効な値を指定することによって指定されます。

検索要求のサンプル (パラメーターの指定あり)

```
<NH_SERVER_MESSAGE protocol_version="false">
  <BASIC_REQUEST_INFO request_type='S' request_id='-1' />
  <DATA_LIST_NAME value='' />
  <SEARCH_NAME NAME="Kidder Byron Licensed Land Surveyor"
    NEC="0" CULTURE_SN="" CULTURE_GN=""
    SRCH_OPT="2" />
  <GENERAL_PARMS SHOULD_USE_INDEX="Y" NAME_THRESH="0.75"
    MAX_RETURN_NAMES="50" RETURN_ORIG_NAMES="N" />
  <COMP_PARMS_ON INITIAL_TOKEN_SCORE_ADJ="0.98"
    INITIAL_INITIAL_SCORE_ADJ="1.0"
    NAME_UNKNOWN_SCORE_ADJ="0.98"
    NO_NAME_SCORE_ADJ="0.97"
    ANCHOR_FACTOR_ADJ="0.98"
    OOPS_FACTOR_ADJ="1.05"
    COMPRESSED_SCORE_MAX_ADJ="1.0"
    FIELD_THRESH_ADJ="0.97"
    FIELD_WEIGHT_ADJ="0.98"
    MISSING_STEM_FACTOR_ADJ="0.98"
    FIELD_VARIANT_SCORE_ADJ="1.05" />
</NH_SERVER_MESSAGE>
```

次の検索要求には、個人名の具体的な設定および調整係数が含まれています。

検索要求のサンプル (パラメーターの指定あり)

```
<NHServerMessage protocol_version="4.1">
  <BASIC_REQUEST_INFO request_type="S" request_id="-1" />
  <DATA_LIST_NAME value="data_list_1" />
  <SEARCH_NAME SN="Freeman"
    GN="Harlow J"
    CULTURE_SN="1"
    CULTURE_GN="1" />
  <GENERAL_PARMS
    NAME_THRESH="0.600000"
    MAX_RETURN_NAMES="15"
    RETURN_ORIG_NAMES="N" />
  <COMP_PARMS_GN
    SCORE_MODE="1"
    OOPS_FACTOR="0.600000"
    NO_NAME_SCORE="0.750000"
    NAME_UNKNOWN_SCORE="0.750000"
    MISSING_TAQ_FACTOR="0.970000"
    MISSING_STEM_FACTOR="0.950000"
    MATCH_VARIANTS="Y"
    MATCH_INITIALS="Y"
    LEFT_BIAS="N"
    INITIAL_TOKEN_SCORE="0.800000"
    INITIAL_INITIAL_SCORE="0.900000"
    FIELD_WEIGHT="0.800000"
    FIELD_THRESH="0.500000"
    DO_COMPRESSED_SCORE="Y"
    COMPRESSED_SCORE_MAX="0.950000"
    DIF_TAQ_FACTOR="0.990000"
    CULTURE="4"
    ANCHOR_TYPE="1"
    ANCHOR_FACTOR="0.600000"
    MATCH_FIELD_VARIANTS="Y"
    FIELD_VARIANT_SCORE="0.95" />
  <COMP_PARMS_SN
    SCORE_MODE="0"
```

```

OOPS_FACTOR="0.750000"
NO_NAME_SCORE="0.750000"
NAME_UNKNOWN_SCORE="0.700000"
MISSING_TAQ_FACTOR="0.980000"
MISSING_STEM_FACTOR="0.960000"
MATCH_VARIANTS="Y"
MATCH_INITIALS="Y"
LEFT_BIAS="N"
INITIAL_TOKEN_SCORE="0.850000"
INITIAL_INITIAL_SCORE="0.900000"
FIELD_WEIGHT="1.000000"
FIELD_THRESH="0.550000"
DO_COMPRESSED_SCORE="Y"
COMPRESSED_SCORE_MAX="0.950000"
DIF_TAQ_FACTOR="0.990000"
CULTURE="4"
ANCHOR_TYPE="1"
ANCHOR_FACTOR="0.800000"
MATCH_FIELD_VARIANTS="Y"
FIELD_VARIANT_SCORE="0.95"/>

```

</NHServerMessage>

BASIC_REQUEST_INFO

属性	必須?	制限
request_type	はい	S である必要があります。
request_id	いいえ	Distributed Search が応答で戻す数値。Distributed Search が数値を生成するようにするには、-1 を使用します。

DATA_LIST_NAME

属性	必須?	制限
value	はい	Distributed Search では現在実装されていません。

SEARCH_NAME

属性	必須?	制限
SN	はい ¹	姓。最大 128 文字。
GN	はい ¹	名。最大 128 文字。
NAME	はい ²	名前。最大 128 文字。組織名などのフィールド化されていない名前の保持に使用されます。
NEC	はい	名前付きエンティティ・カテゴリー。有効値は以下のとおりです。 O = 組織名 P = 個人名

属性	必須?	制限
CULTURE_GN	いいえ	名の国/地域別情報コード。 このフィールドが指定されていない場合、現在のパラメーターが使用されます。
CULTURE_SN	いいえ	姓の国/地域別情報コード。 このフィールドが指定されていない場合、現在のパラメーターが使用されます。
SRCH_OPT	いいえ	1 = 個人名リストのみの検索 2 = 組織リストのみの検索 3 = 個人名と組織名の両方のリストの検索

¹NEC が P のときにのみ必須。

²NEC が O のときにのみ必須。

シャットダウン要求:

シャットダウン要求を使用すると、Distributed Search のすべての Searcher およびコミュニケーション・マネージャー (commgr) の制御されたシャットダウンを開始できます。キューに入れられたトランザクションは、シャットダウンが実施される前に完了します。

有効な要求には、Distributed Search 構成ファイルで指定されたパスワードに一致する、大/小文字の区別があるパスワードが含まれます。commgr のシャットダウンの直前に、送信者のオリジナル要求 ID を含む成功の応答が、要求元のクライアント・プロセスに返されます。このメッセージは、標準的な Distributed Search 運用デプロイメントで管理者またはサポート担当者の使用に制限されることを意図しています。

シャットダウン要求のサンプル

```
<NHServerMessage protocol_version="4.2">
  <BASIC_REQUEST_INFO request_type="X" request_id="-1"/>
  <SHUTDOWN_PASSWORD value="NHSERVER"/>
</NHServerMessage>
```

BASIC_REQUEST _INFO

属性	必須?	制限
request_type	はい	X である必要があります。
request_id	いいえ	Distributed Search が応答で戻す数値。Distributed Search が数値を生成するようにするには、-1 を使用します。

SHUTDOWN_PASSWORD

属性	必須?	制限
value	いいえ	使用されていません。

状況要求:

状況要求を使用すると、現在の処理状況およびキュー状況を報告するよう Distributed Search に命令できます。 Distributed Search は、送信者のオリジナル要求 ID を含む状況応答を使用して応答します。

状況要求のサンプル

```
<NHServerMessage protocol_version="4.2">
  <BASIC_REQUEST_INFO request_type="T" request_id="-1"/>
</NHServerMessage>
```

BASIC_REQUEST_INFO

属性	必須?	制限
request_type	はい	T である必要があります。
request_id	いいえ	Distributed Search が応答で戻す数値。 Distributed Search が数値を生成するようにするには、-1 を使用します。

更新要求:

更新要求を使用すると、Distributed Search セッション中に、メモリーに常駐する Distributed Search 名前リスト (データ・リスト) 内に既に存在する名前を更新できます。正常に完了した場合は、送信者のオリジナル要求 ID を含む成功の応答メッセージが返されます。

この要求は、RECORD_ID_TO_UPDATE 属性で指定された値に一致する NAME_ID フィールドを持つシステム内のすべてのレコードを変更します。更新要求は、削除要求を実行してから追加要求を実行するため、遅くなる場合があります。複数のレコードが同じ NAME_ID を共有している場合、それらのレコードが影響を受ける可能性があります。

名前付きエンティティ・カテゴリー (NEC) フィールドは、名前のタイプと、Distributed Search の検索リストを更新するときどのフィールドを使用するかを決定します。

- O – 組織名が NAME フィールドを使用します。
- P – 個人名が SN フィールドおよび GN フィールドを使用します。
- A – すべて。つまり 3 つのフィールドすべてが使用され、2 つの名前が追加されます。

更新要求のサンプル

```
< NHServerMessage protocol_version="4.1">
  <BASIC_REQUEST_INFO request_type="U" request_id="-1"/>
  <UPDATED_NAME
    SN="Acme Light Industries Inc."
    GN=""
```

```

NAME="Acme Light Industries Inc."
NEC="A"
  NAME_ID="123456"
  CULTURE_SN="1"
  CULTURE_GN="1"
  ALT_PARSE="N" />
<RECORD_ID_TO_UPDATE value="123456"/>
<DATA_LIST_NAME value=""/>
</ NHServerMessage>

```

BASIC_REQUEST_INFO

属性	必須?	制限
request_type	はい	U である必要があります。
request_id	いいえ	Distributed Search が応答で戻す数値。Distributed Search が数値を生成するようにするには、-1 を使用します。

UPDATED_NAME

属性	必須?	制限
SN	Yes ¹	姓。最大 128 文字。
GN	Yes ¹	名。最大 128 文字。
NAME	はい ²	名前。最大 128 文字。組織名などのフィールド化されていない名前の保持に使用されません。
NEC	はい	名前付きエンティティ・カテゴリ。有効値は以下のとおりです。 O = 組織名 P = 個人名 A = すべて
NAME_ID	はい	更新される名前の ID。最大 256 文字の英数字。ID はユニークである必要はありません。
CULTURE_SN	いいえ	姓の国/地域別情報コード。
CULTURE_GN	いいえ	名の国/地域別情報コード。
ALT_PARSE	いいえ	これは名前の代替解析か? (Y または N)

¹NEC が P のときにのみ必須。

²NEC が O のときにのみ必須。

RECORD_ID_TO_UPDATE

属性	必須?	制限
value	はい	更新される名前。最大 256 文字。NAME_ID はこのフィールドに一致している必要があります。

DATA_LIST_NAME

属性	必須?	制限
value	いいえ	Distributed Search では現在実装されていません。

Distributed Search XML 応答

Distributed Search XML 要求の実行が処理された後で、要求が正常に完了したか、または失敗したかの指示を記述した応答メッセージを受け取ります。また、状況要求を実行した後で、状況応答を受け取ることもできます。

エラー応答:

エラー応答は、要求の処理中に Distributed Search でエラーが発生した場合に返されます。この応答は、エラーが発生した送信者のオリジナル要求 ID を示します。具体的なエラー・コードおよびメッセージは、別のセクションで説明します。

エラー応答のサンプル

```
<NHServerMessage server_version="4.2" >
  <REQUEST_ID value="4070517"/>
  <ERROR supplied_data_list_name=""
    severity=""
    error_msg="Searcher 3 is not responding"
    error_code="GNRDS-123"/>
</NHServerMessage>
```

REQUEST_ID

属性	必須?	制限
value	N/A	エラーの原因となった要求の ID。

ERROR

属性	必須?	制限
severity	N/A	Distributed Search では現在実装されていません。
error_msg	N/A	エラーの記述。
error_code	N/A	エラー・メッセージ固有のコード。

検索結果の応答:

検索結果の応答は、検索要求またはオリジナル・データ要求を実行した後に返されます。応答は、送信者のオリジナル要求 ID を提供することによって要求を示します。この応答のすべてではありませんが、ほとんどのフィールドが、両方の要求によって使用されます。

次の例は、2 つの一致する名前を含む検索結果の応答を示しています。

検索結果の応答のサンプル

```
<NHServerMessage protocol_version="4.1">
  <SEARCH_RESULTS>
    <NAME
      SN="FREEMAN"
      GN="HARLOW J"
      NAME=""
      NEC="P"
      NAME_ID="123456"
      SN_SCORE="1.000000"
      GN_SCORE="1.000000"
      FULL_NAME_SCORE="1.000000"
      NAME_CNT="1"
      IS_REG="N"
      ALT_PARSE="N"/>
    <NAME
      SN=""
      GN=""
      NAME="Acme Light Industries Inc."
      NEC="0"
      NAME_ID="73"
      SN_SCORE="0.7700"
      GN_SCORE="0.85000"
      FULL_NAME_SCORE="0.82000"
      NAME_CNT="5"
      IS_REG="N"
      ALT_PARSE="N"/>
  </SEARCH_RESULTS>
  <REQUEST_ID value="9070517"/>
</NHServerMessage>
```

REQUEST_ID

属性	必須?	制限
value	N/A	検索要求の ID。

SEARCH_RESULTS_NAME

属性	必須?	制限
SN	N/A	姓。最大 60 文字。
GN	N/A	名。最大 60 文字。
NAME	いいえ	名前。最大 128 文字。組織名などのフィールド化されていない名前の保持に使用されません。

属性	必須?	制限
NEC	いいえ	名前付きエンティティ・カテゴリ。有効値は以下のとおりです。 O = 組織名 P = 個人名
NAME_ID	N/A	固有の名前の場合、この値は NH_ID です。オリジナルの名前の場合、この値はカスタマー ID です。NAME_ID は最大 256 文字です。
SN_SCORE	N/A	姓のスコア。0 から 1.00
GN_SCORE	N/A	名のスコア。0 から 1.00
FULL_NAME_SCORE	N/A	氏名のスコア。0 から 1.00
NAME_CNT	N/A	この固有の名前が適用されるオリジナルの名前の数。オリジナルの名前が返される時は、1 に設定します。
ALT_PARSE	N/A	これは代替名の解析の結果か? オリジナルの名前にのみ使用されます。
IS_REG	N/A	これは名前の正規化の結果か? オリジナルの名前にのみ使用されます。

状況応答:

状況応答は、状況要求から現在の処理およびキューの状況を返し、関連付けられた状況要求からの要求 ID を含みます。

状況応答のサンプル

```
<NHServerMessage server_version="4.2" >
  <REQUEST_ID value="4070517"/>
  <STATUS_MESSAGE
    value="Server running with 0 searches running
    and 0 queued\"/>
</NHServerMessage>
```

REQUEST_ID

属性	必須?	制限
value	N/A	関連付けられた状況要求の ID。

STATUS_MESSAGE

属性	必須?	制限
value	N/A	Distributed Server の状態を記述するテキスト。

成功の応答:

成功の応答は、メッセージが正常に受信および処理されたことを示す、Distributed Search によって使用される確認です。応答は、送信者のオリジナル要求 ID を提供することによって、正常に完了したトランザクションを示します。

成功の応答のサンプル

```
<NHServerMessage protocol_version="4.2">  
  <REQUEST_ID value="1070555"/>  
</NHServerMessage>
```

REQUEST_ID

属性	必須?	制限
value	N/A	発信元の要求メッセージの ID。

Enterprise Name Search を使用した名前の検索

Enterprise Name Search (ENS) は、企業における大量かつ大規模な名前検索を非常に大きな名前リストで分散するためのインフラストラクチャーを提供します。ENS は、効率的な名前検索機能や名前リスト管理のため、また名前検索処理の構成、管理、およびモニターのために、IBM NameWorks を活用します。ENS は、パフォーマンス上の必要に応じた水平スケーリング、高可用性およびフェイルオーバーを備えたデプロイメント、サード・パーティー・ソフトウェアおよび既存のレガシー・システムとのクライアント統合に必要な SOAP および REST の両インターフェースのために追加オプションを提供します。

Web サービス API によって独自のクライアント・プログラムから ENS の検索機能を使用するか、あるいは ENS で提供されるグラフィカル・ユーザー・インターフェースを使用することができます。Web サービスを使用する予定の場合でも、GUI では Web サービスを使用して検索を実行するため、検索を試したり、ENS の機能と動作を理解したりする上で GUI は役立ちます。

Enterprise Name Search ユーザー・セキュリティの管理

ENS は、WebSphere Liberty アプリケーション・サーバーを取り込んでその Web コンポーネントをホストし、WebSphere Liberty のセキュリティの実装を基にして認証や許可を行います。

デフォルトでは、WebSphere Liberty はユーザー、グループ、パスワードに関する情報を単純にファイル・ベースで格納します。ENS では、この情報が含まれるファイルは *ensroot/ibm-home/wlp/users.xml* にあります。このファイルを編集して、ユーザーを追加、削除、管理できます。この操作はシステムが稼働しているときに行うことができます。

さらに複雑なセキュリティ要件を持つクライアントは、ENS のインストール後に、単純なファイル・ベースの格納の代わりに LDAP ユーザー・レジストリーを使

用するよう WebSphere Liberty の構成を変更できます。これは、GNM 資料の範囲外ですが、WebSphere Liberty セキュリティーの資料で説明されています。

また、ユーザーはパスワード変更 GUI 画面で自分の ENS パスワードを変更できます。

wspswd ユーティリティーのコマンド行構文

wspswd ユーティリティーは、ens.passwd パスワード・ファイルを管理するために使用するコマンド行ユーティリティーです。パスワード・ファイルには、各 Enterprise Name Search ユーザーに割り当てられているユーザー名、パスワード、およびセキュリティー・ロール・グループのリストが含まれます。

wspswd ユーティリティーを使用するには、<install>/bin ディレクトリーに移動し、コマンド行からコマンドを入力します。wspswd ユーティリティーの構文は次のとおりです。

- Linux の場合:

```
./wspswd optional_path_location_of_the_password_file options
```

- Windows の場合:

```
wspswd.bat optional_path_location_of_the_password_file options
```

password_file へのパスが指定されていない場合、デフォルトの <install>/ibm-home/ens ディレクトリーに存在する ens.passwd ファイルになります。

コマンド・オプション

-c username

指定されたユーザー名で新規ユーザーを作成し、そのユーザーのパスワードを生成します。

-c username password

指定されたユーザー名とパスワードでユーザーを作成します。

-d username

指定されたユーザー名とパスワードを持つユーザーを削除します。ユーザーを削除することで、ユーザー名とパスワードが削除され、そのユーザーが所属するすべてのセキュリティー・ロール・グループからユーザーが削除されます。

-l

Enterprise Name Search のすべての現行ユーザーをユーザー名順にリストします。

-p username

指定されたユーザーの新規パスワードを生成します。

-p username password

指定されたユーザーについて、パスワードを指定された値に変更します。

-a username grp1 grp2 ...

ユーザーをセキュリティー・ロール・グループに割り当てます。セキュリティー・ロール・グループによって、どの ENS コンポーネントにユーザーがアクセスできるかと、ユーザーが実行できる機能のタイプが決まります。

-r username grp1 grp2 ...

指定されたユーザーに割り当てられているセキュリティー・ロール・グループをリセットします。

-u username grp1 grp2 ...

ユーザーを 1 つ以上のセキュリティー・ロール・グループから削除します。

注: ユーザー、パスワード、およびロールには、a から Z までの文字 (大/小文字) と 0 から 9 を使用する必要があります。

例

Windows 環境で、ユーザーを作成し、パスワードを指定し、セキュリティー・ロール・グループを新規ユーザーに割り当てするには、以下のようになります。

1. 「ConsoleUser1」というユーザー名を持つユーザーを作成し、「ens1234」というパスワードを割り当てます。

```
wspswd.bat -c ConsoleUser1 ens1234
```

2. セキュリティー・ロール・グループ「admin」をユーザーに割り当てます。

```
wspswd.bat -a ConsoleUser1 admin
```

ユーザーの作成とセキュリティー・グループの割り当て

`ensroot/ibm-home/wlp/users.xml` ファイル内で `<user>` 要素や `<member>` 要素を追加したり変更したりすることで、ユーザーと権限の定義や管理ができます。

このタスクについて

`users.xml` ファイルの先頭付近に「List users here」というコメントが付いたセクションがあります。

```
<!-- List users here -->
```

```
<user name="ensadmin" password="{aes}AICwWs/XAgx70bztj0s09zkS31vubq0rERfNM/o81K"/>
```

ユーザーを作成するには、このセクションに行を追加します。インストール時に、ここで作成した 2 つの項目を削除したり変更したりしないでください。新しいユーザーを追加する場合には、以下のようなプレーン・テキストの初期パスワードを付与できます。

```
<user name="smith" password="changeMe"/>
```

ユーザー名とパスワードには、文字 (a-z と A-Z) と数字 (0-9) のみ使用します。

「ユーザー・パスワードの変更またはリセット」に記述されているように、ユーザー項目を追加した後で、ユーザーは自分のパスワードを変更する必要があります。[このトピックへのリンク]

`<user>` 要素を追加するだけでなくユーザーのグループ・メンバーシップを設定するには、1 つ以上のグループ内のユーザーをリストし、新しいユーザーが実行できる操作を指定する必要があります。ENS では、ユーザーは以下のグループに入ることができます。

表 12.

グループ	権利
admins	<ul style="list-style-type: none"> すべてのコンソール操作を使用して、システムの状態を表示したり変更したりします。 名前リストを検索したり管理したりします。 ダッシュボードを使用します。
console_users	<ul style="list-style-type: none"> すべてのコンソール操作を使用して、システムの状態を表示したり変更したりします。 ダッシュボードを使用します。
dashboard_users	<ul style="list-style-type: none"> ダッシュボードを使用して、システムの状態を表示します。
managers_all_lists	<ul style="list-style-type: none"> 名前リストを検索したり管理 (追加/削除) したりします。
managersNNN* (managers101 など)	<ul style="list-style-type: none"> NameLoader でのリストの作成時にこの manager ロールが指定された名前リストを検索したり管理したりします。
searchers_all_lists	<ul style="list-style-type: none"> 名前リストを検索します。
searchersNNN * (searchers101 など)	<ul style="list-style-type: none"> NameLoader でのリストの作成時にこの searcher ロールが指定された名前リストを検索します。 きめ細かく制御するために使用します。

「admins」グループのメンバーシップにより、すべての ENS 操作へのアクセス権が提供されます。このグループのユーザーを他のグループに追加する必要はありません。

「managers_all_lists」グループのメンバーシップにより、すべての名前リスト内のユーザー検索を行ったり、すべての名前リストを管理したりできます。このグループのユーザーを他の searchers グループや managers グループに追加する必要はありません。

「searchers_all_lists」グループのメンバーシップにより、すべての名前リスト内のユーザー検索を行うことができます。このグループのユーザーを他の searchers グループに追加する必要はありません。

「searchersNNN」グループと「managersNNN」グループを使用すると、特定のリストへのアクセス権をきめ細かく制御できます。ほとんどの ENS インストール済み環境でこの機能は使用しません。ユーザーをグループに入れるには、users.xml で該当する <group> 要素を見つけて、そのユーザーの <member> 要素を追加します。例えば、以下のようにして、既存のユーザー「smith」を console_users グループのメンバーにすることができます。

```
<group name="console_users">
<member name="smith"/>
</group>
```

<member name="smith"> 要素を削除すると、「smith」を console_users グループから削除できます。

ユーザー・パスワードの変更またはリセット

Enterprise Name Search にアクセスするためのユーザー・パスワードは、users.xml ファイルにより管理されます。

このタスクについて

管理者は、users.xml 項目にユーザー項目を追加して新しいユーザーを作成するとき、以下のような初期プレーン・テキスト・パスワードを付与できます。

```
<user name="jdoe" password="changeMe"/>
```

その後ユーザーは、<http://host:port/changePassword.jsp> のパスワードの変更ページを (実行中の ENS サーバー上で) 参照し、自身のパスワードを変更できます。古い (初期) パスワードを使用してログインするよう求めるプロンプトが表示され、それから新しいパスワードを 2 度入力できるフォームが表示されます。入力して「パスワードの変更 (Change Password)」をクリックすると、users.xml ファイルが更新され、(AES エンコードされた) 新しいパスワードが格納されます。

```
<user name="jdoe" password="{aes}ACdUKKZ0yQi09b4jY83d19UNGUmBbdT9znIT8kmUPmu3"/>
```

パスワードは、長さ 5 文字から 30 文字の間でなければなりません。ユーザーのパスワードとユーザー名は同じであってはなりません。

ユーザーがパスワードを忘れた場合、管理者は

```
<user>
```

項目に上記と同じようなプレーン・テキストをもう一度付与して、パスワードをリセットすることができます。

```
<user name="jdoe" password="changeMe"/>
```

この後、再度ユーザーが上記のように変更する必要があります。

Enterprise Name Search 機能へのユーザー・アクセスの削除

特定のグループからユーザーを削除する (一部の ENS 機能に対するアクセスを取り消す) には、users.xml ファイルを編集し、該当するグループからそのユーザーの <member...> ノードを削除します。

このタスクについて

ENS からユーザーを完全に削除するには、すべてのグループから、その <user> ノードを削除します。

特定のリストに対する細かいアクセス制御

ENS の users.xml ファイルには「searchers_all_lists」というグループがあり、このグループのメンバーは、すべての名前リストで検索を実行できます。

「managers_all_lists」というグループもあり、このグループのメンバーは、すべてのリストで検索を実行できるうえに、名前を追加、削除、および取得して管理することもできます。

このタスクについて

インストール済み環境の多くは、これらの汎用的なサーチャー・グループおよびマネージャー・グループ (より汎用性の高い「admins」グループもあります) によるアクセス制御モデルで十分です。

クライアントによっては、特定のユーザーが検索または管理できる名前リストを限定するために、より細かいモデルを使用する必要がある場合があります。ENS は、番号が付いた「searchersNNN」および「managersNNN」グループによって、これをサポートします。

NameLoader を使用して ENS に名前リストを追加するときには、そのリストのサーチャー・ロールの名前を指定します (「searcher101」など)。関連付けられているグループ (searchers101) のユーザーに、そのリスト内の名前を検索する許可が与えられます。

同様に、そのリストのマネージャー・ロールの名前も指定します (「manager101」など)。関連付けられているグループ (managers101) のユーザーに、Web サービスを使用してそのリストの名前を追加および削除する許可が与えられます。

users.xml ではグループの名前は複数形 (searchers101) で表されていますが、NameLoader の対応するロールの名前は単数形 (searcher101) であることに注意してください。

これらの細かいロールはオプションです。NameLoader で名前を追加するときにはロールを指定する必要がありますが、汎用的な「searchers_all_lists」および「managers_all_lists」グループで十分であると判断した場合には、これらの細かいロールを使用する必要はありません。

デフォルトでは、users.xml に含まれているグループは、searchers101 から searchers120 と managers101 から managers120 です。より広い範囲のグループが必要な場合は (最大範囲は searcher1 から searcher500 と manager1 から manager500)、users.xml ファイルに <group> 要素を追加し、applications.xml ファイルに <security-role> 要素を追加して定義することができます。どちらのファイルも *ensroot/ibm-home/wlp* フォルダーにあります。詳しくは、これらのファイルのコメントを参照してください。

ENS で検索 GUI か Web サービスのいずれかを使用して名前を検索した場合、検索許可を持つ名前リストからの結果のみが表示されます。例えば、CUSTOMERS 名前リストに searcher101 ロールが関連付けられ、EMPLOYEES 名前リストに searcher102 ロールが関連付けられ、ユーザー「jdoe」が searchers101 グループの唯一のメンバーであるとし、この場合は以下ようになります。

- ENS 検索 GUI か Web サービスのいずれかを使用してユーザー jdoe が「John Smith」を検索すると、CUSTOMERS リストからの結果のみが表示され、EMPLOYEES からの結果は表示されません。
- ユーザー jdoe が ENS 検索 GUI を使用するときに表示される検索可能な名前リストのリストには、CUSTOMERS が含まれますが、EMPLOYEES は含まれません。
- ユーザー jdoe が「get namelists」Web サービスを使用するとき生成されるリストには、CUSTOMERS が含まれますが、EMPLOYEES は含まれません。

NameLoader ユーティリティーを使用した名前リストの管理

Enterprise Name Search の名前リストは、NameLoader ユーティリティーを使用して管理できます。NameLoader ユーティリティーは、Java ベースの独立したプログラムです。これは、名前リストから未加工のソース名を取り込み、IBM NameWorks を使用してそれらの名前を分析し、名前および名前の分析結果を ENS データベースに保管します。

ENS の NameLoader ユーティリティーは、機能的には Distributed Search で使用する Name Preprocessor ユーティリティーと似ていますが、これに取って変わるものではありません。NameLoader ユーティリティーは、名前リスト・ファイルから名前をロードし、「ソース名」と「検索名」という 2 つの異なる形式でデータベースに保管します。

ソース名とは、最初に受け取った名前であり、これには最初の解析とスクリプトも含まれます。NameLoader は、GNM を使用してこれらのソース名を分析して、分析済みの形式、つまり検索名を生成します。この分析には、カテゴリー (個人または組織)、国/地域別情報の判別、および (個人名の場合には) 名前と姓を分ける解析が含まれます。NameLoader は、ソース名、検索名、および、ソース名と検索名のマッピングを保管するために、データベース・レコードを検索または追加します。

NameLoader ユーティリティーを使用して名前を再ロードまたは削除することもできます。ENS セルがアクティブであるときに、NameLoader を使用して名前をリストからロードしたり、再ロードしたり、削除したりできます。ただし、効率的な削除操作 (「すべてクリア」) のほとんどは、セルが非アクティブでなければ実行できません。

名前リスト・ファイルからのソース名の読み取り

名前リスト・ファイルは、CSV (コンマ区切り値) 形式のフラット・テキスト・ファイルです。ローマ字以外を使用する場合、ファイルは UTF-8 エンコードを使用する必要があります。名前リストの各行は、1 つのソース名を表します。通常は、以下の情報が含まれます。

表 13. 入力する名前のための CVS ファイル内のフィールド - 通常形式

フィールド	意味	例	デフォルト	備考
0	姓	"SMITH"	空白	姓または名のいずれかを空白にすることができます。ただし、両方を空白にすることはできません。
1	名	"JOHN"	空白	姓または名のいずれかを空白にすることができます。ただし、両方を空白にすることはできません。
2	idData	300011	デフォルトなし	必須。データ・ソース内で固有でなければならない ID。

表 13. 入力する名前のための CVS ファイル内のフィールド - 通常の形式 (続き)

フィールド	意味	例	デフォルト	備考
3	姓の国/ 地域別情報	3	ブラン ク	オプション
4	名の国/ 地域別情報	3	ブラン ク	オプション
5	カテゴリー	0	ブラン ク	オプション。名前のタイプ (個人名、組織名、または指定なし) を示します。個人の場合は「p」、組織の場合は「o」を使用します。代わりに、個人に数字の「0」、組織に「1」を使用することもできますが、文字を使用する方が混乱せずに済む可能性があります。

通常の形式を使用した有効な入力行の例:

Smith,John,x101,1,1,p - John Smith、外部 ID は x101、姓名の国/地域別情報は英語圏。個人名。

John Smith,,x102,1,1,p - 姓と名ではなく単一フィールドによる名前。コンマが 2 つ続いていることに注意してください。

Smith,John,x103,, - 国/地域別情報とカテゴリーが省略されています。GNM が分析によって両方を判別します。

Smith,John,x104 - 同上。末尾のフィールドを省略する場合は、コンマは不要です。

■,■,■,x105,,,p - 名前にローマ字以外のスクリプトが使用されている可能性があることを示しています。

Smith Corp,,x106,,,o - sn フィールドに組織名が含まれている (コンマが 2 つ続いていることに注意してください)。明示的な「o」カテゴリー。

Smith Corp,,x107 - 組織名。カテゴリーは記されていません。GNM が分析によって判別します。

名前リスト・ファイルからのソース名の読み取り - 拡張された入力形式

ほとんどの場合には、上記の通常の入力形式で十分ですが、ローマ字以外のスクリプトを入力として使用する場合のために、別の形式も用意されています。ほとんどのスクリプト・タイプは自動認識が可能ですが、ENS と GNM は、入力名の中国語繁体字と漢字を自動的に区別することができません。

仮名なしの中国語文字だけを使用して作成された日本の組織名が名前リストに含まれている場合、GNM はそれを日本語の名前として自動的に識別することができません。そのような名前を含む名前リストの場合は、必要に応じてスクリプト・タイプを明示的に指定できる、新しい拡張ファイル形式を使用する必要があります。こ

の形式は、フィールドの順序が異なり、スクリプト・タイプを明示的に指定できるフィールドが追加されています。この形式を使用する場合、CSV ファイルの各行には以下の情報が含まれます。

このようなファイルを使用して NameLoader を実行する場合には、NameLoader コマンド行に「-extendedInputFormat」を指定するか、ローダーの構成ファイルに「extendedInputFormat=true」を指定する必要もあります。

表 14. 入力する名前のための CSV ファイル内のフィールド - 拡張形式 - 個人名のデータ行

フィールド	意味	例	デフォルト	備考
0	外部 ID	300011	デフォルトなし	必須。データ・ソース内で固有でなければならない ID。
1	カテゴリー	0	ブランク	オプション。名前のタイプ (個人名、組織名、または指定なし) を示します。個人の場合は「p」、組織の場合は「o」を使用します。代わりに、個人に数字の「0」、組織に「1」を使用することもできますが、文字を使用する方が混乱せずに済む可能性があります。
2	スクリプト・タイプ		ブランク	個人名の場合、このフィールドは使用せずに空白のままにする必要があります。
3	姓または完全な個人名	「Smith」または「John Smith」	ブランク	姓または名のいずれかをブランクにすることができます。ただし、両方をブランクにすることはできません。
4	名	"JOHN"	ブランク	姓または名のいずれかをブランクにすることができます。ただし、両方をブランクにすることはできません。
5	姓の国/地域別情報	1	ブランク	オプション
6	名の国/地域別情報	1	ブランク	オプション

表 15. 入力する名前のための CSV ファイル内のフィールド - 拡張形式 - 組織名のデータ行

フィールド	意味	例	デフォルト	備考
0	外部 ID	300011	デフォルトなし	必須。データ・ソース内で固有でなければならない ID。

表 15. 入力する名前のための CSV ファイル内のフィールド – 拡張形式 – 組織名のデータ行 (続き)

フィールド	意味	例	デフォルト	備考
1	カテゴリー	0	ブランク	オプション。名前のタイプ (個人名、組織名、または指定なし) を示します。個人の場合は「p」、組織の場合は「o」を使用します。代わりに、個人に数字の「0」、組織に「1」を使用することもできますが、文字を使用する方が混乱せずに済む可能性があります。
2	スクリプト・タイプ			(オプション) スクリプト・タイプを示す明示的なヒント。ほとんどの場合、ブランクのままです。現在、このフィールドの用途は、CHINESESCRIPT または JAPANESESCRIPT を指定して、中国語繁体字と漢字の組織名の入力を区別することだけです。その他のスクリプト・タイプは、名前のテキストから自動的に判別されます。
3	組織名	「Smith Corp」	デフォルトなし	ブランクにすることはできません
4	組織名の国/地域別情報	1	ブランク	オプション

NameLoader アナライザー・コンポーネント

Name Analyzer コンポーネントは、各入力ソース名に対して以下の分析を実行します。

- 名前を音訳することにより、各ソース名を処理する。
- フラット CSV ファイルの「カテゴリー」要素がブランクの場合に、名前を名前タイプにカテゴリー化する。
- 「カテゴリー」が「個人」の場合に、名前の元の形式に対して最適な解析を見つける。
- ソース名に基づいて、音訳された 1 つ以上の検索名を見つける。

ソース名と検索名

ソース名 とは、名前リストから最初にロードされる名前のことです。一方、検索名は、IBM NameWorks によって実行される分析の結果として生成される名前を指します。スキーマは、ソース名と検索名の両方を保管します。

ソース名は ENS_SOURCE_NAME テーブルに置かれます。検索名は ENS_SEARCH_NAME テーブルに置かれます。Enterprise Name Search は、その 2 つの間のマッピングを管理します。パフォーマンス上の理由により、名前をスキーマ・テーブルに書き込む際には複数のスレッドを使って書き込みを行います。

各スレッドは、バッチ・データベース更新を使用して複数のレコードを追加または更新します。スレッドの数およびバッチ・サイズは、NameLoader ユーティリティーの構成ファイルで構成することができます。

2 つの検索名が同一でも国/地域別情報が異なる場合、それらの検索名はそれぞれ別々に保管されます。例えば、韓国語の国/地域別情報が指定されている「June Park」は、英語の国/地域別情報が指定されている「June Park」とは別に保管されます。一致規則がこれら 2 つの国/地域別情報で異なっているからです。

2 つの検索名が同一で代替解析フラグが異なる場合も、それらの検索名はそれぞれ別々に保管されます。例えば、ソース名「Elton John」はソース名「John Elton」と異なります。「Elton John」の名前検索を要求すると両方のソース名が検出されて報告されますが、代替解析に基づいているため、ソース名 John Elton はより低い一致スコアで報告されます。

loader.config ファイル

NameLoader ユーティリティーには、loader.config という専用の構成ファイルがあります。これにより、Enterprise Name Search で使用する IBM NameWorks インスタンスの動作を指定します。NameLoader ユーティリティーのオプションと構成の詳細はすべて、このファイル内に指定されます。

NameLoader コマンド

NameLoader ユーティリティーは、Enterprise Name Search で使用する名前リストを管理するために使用されます。OS コマンド・プロンプトで単一コマンドを入力して特定の操作を行うことにより、NameLoader を実行します。コマンド行オプションおよび NameLoader 構成ファイルに指定する設定によって、NameLoader の実行内容を制御することができます。

NameLoader ユーティリティーを実行するには、<install path>/bin/ ディレクトリに移動し、以下のアクションまたはパラメーターを 1 つ以上指定して **nameLoader** を入力します。

表 16. ENS NameLoader コマンド

アクション	影響	例
-load	(デフォルト) CSV ファイルからソース名をロードし、それらを処理して検索名にし、以下の表で項目を追加または更新します。 <ul style="list-style-type: none"> ens_source_name ens_search_name ens_search_source_name ens_name_list オプションで、リビジョンをトラッキングするために、レコードを ens_search_name_revs テーブルおよび ens_search_name_adds テーブルに追加することも行います。	デフォルト設定を使用して単一の名前リストをロードします。 <pre><install path>/bin/nameLoader -load -nlc EMPLOYEES -nld "Current employees" -in employees.csv</pre> 独自のローダー構成ファイルを使用して単一の名前リストをロードします。 <pre><install path>/bin/nameLoader -load -nlc EMPLOYEES -nld "Current employees" -in employees.csv myLoaderConfigFile.config</pre> 注: 上記のどの例においても、最後のコマンド行引数としてご自分の構成ファイルの 1 つを指定することができます。

表 16. ENS NameLoader コマンド (続き)

アクション	影響	例
-reload	クリアとロードの組み合わせ。	デフォルト設定を使用して、単一の名前リストをクリアして再ロードします。 <pre><install path>/bin/nameLoader -reload -nlc EMPLOYEES -nld "Current employees" -in employees.csv</pre>
-clear	特定の名前リストの ens_name_list および ens_source_name 項目をデータベースから削除します。 ens_search_name からの項目の削除は行われません。	単一の名前リストをクリアします。 <pre><install path>/bin/nameLoader -clear -nlc EMPLOYEES</pre>
-clearAll	ens_name_list 、 ens_source_name 、および ens_search_name からすべての項目を削除します。	すべての名前リストをクリアします。 <pre><install path>/bin/nameLoader -clearAll</pre>
-usage	NameLoader コーティリティーのすべてのオプションを表示します。	NameLoader コーティリティーの追加オプションを表示します。 <pre><install path>/bin/nameLoader -usage</pre>

インストーラーによって作成されるデフォルトの loader.config ファイルには、各項目について説明するコメントがあります。大量の名前を処理する場合は、おそらく、このファイルに含まれているパフォーマンス関連のパラメーターを調整する必要があります。これにより、NameLoader のパフォーマンスが大きく変化する可能性があります。各パフォーマンス・パラメーターの説明については、ファイルのコメントを参照してください。

NameLoader を実行するときには、最後のコマンド行引数として NameLoader 構成ファイルを指定できます。これを省略すると、デフォルトの loader.config ファイルが使用されます。

ほとんどのオプションおよび構成の詳細 (Global Name Management 構成ファイルのパスを含む) は、NameLoader 構成ファイル内に指定できます。あるいは、通常は名前リストによって異なる詳細をコマンド行引数として指定することもできます。そうすることで、単一の構成ファイルを複数の名前リスト用に使用することができます。

無効な引数または「-usage」引数を指定して NameLoader を実行すると、有効なコマンド行引数の要約が表示されます。

NameLoader 構成ファイル

NameLoader の機能は、構成ファイルとコマンド行引数の組み合わせによって制御されます。NameLoader 構成ファイルは、Global Name Management 構成ファイルとは別のものです。後者は、NameWorks の動作を指定します。このように構成ファイルが分かれているので、さまざまな名前リスト用の複数の NameLoader 構成ファイルで、1 つの共通の Global Name Management 構成ファイルを共有することができます。

ENS をインストールすると、インストール済み環境に適した設定を持つデフォルトの NameLoader 構成ファイルが生成されます。このファイルは、*install_path/data/loader.config* に置かれます。別の構成ファイルを指定せずに nameLoader コマンドを実行すれば、このデフォルトの構成ファイルが使用されま

す。このファイルを編集してその設定を調整することも、別の目的のために独自の構成ファイルを作成することもできます。デフォルトの構成ファイルには、各設定について説明するコメントがあります。

NameLoader を実行するときには、コマンド行引数に NameLoader 構成ファイルを指定します。それ以外のオプションおよび構成の詳細 (Global Name Management 構成ファイルのパスを含む) はすべて、NameLoader 構成ファイル内に指定できます。あるいは、通常は名前リストによって異なる詳細をコマンド行引数として指定することもできます。そうすることで、単一の構成ファイルを複数の名前リスト用に使用することができます。

ENS に付属するサンプル loader.config ファイルには、各項目について説明するコメントがあります。「-usage」引数または無効な引数を指定して NameLoader を実行すると、NameLoader のコマンド行引数に関する情報が表示されます。

表 17. ENS NameLoader 構成ファイルの項目

NameLoader 構成ファイルのキー	型	代表的な値	備考	対応するコマンド行オプション
nameListCode	ストリング	CUSTOMERS	外部名前リストの名前。名前リストごとにローダー構成ファイルを作成する場合を除き、通常はコマンド行引数で指定します。	-nameListCode または -nlc
nameListDescription	ストリング	"Our customers"	データ・ソースの説明。名前リストごとにローダー構成ファイルを作成する場合を除き、通常はコマンド行引数で指定します。	-nameListDescription または -nld
nameListManagerRole	ストリング	manager107	このリストに変更を加えるために必要な ENS ユーザー・ロール。指定できる値は "manager0" から "manager400" です。	-managerRole または -mr
nameListSearcherRole	ストリング	このリスト内を検索するために必要な ENS ユーザー・ロール。デフォルトで用意されている範囲は、「searcher101」から「searcher120」です。*	このリスト内を検索するため、またはそれが存在することを確認するために必要な ENS ユーザー・ロール。有効な値は "searcher0" から "searcher400" です。	-searcherRole または -sr
inFileName	ストリング	project1/ badGuys.in.txt	入力として使用される CSV ファイルのパス。	-inFileName または -in
jdbcDriver	ストリング	com.ibm.db2. jcc.DB2Driver	JDBC ドライバー・クラスの名前。	
dbUrl	ストリング	jdbc:db2:ENS	データベースの URL。	
dbSchema	ストリング	ens	データベース・スキーマ名。	

表 17. ENS NameLoader 構成ファイルの項目 (続き)

NameLoader 構成ファイルのキー	型	代表的な値	備考	対応するコマンド行オプション
dbUserId	ストリング	yourDBUserId	データベース・アクセスのためのユーザー ID。コマンド行引数で指定して、構成ファイルに保管しないようにすることができます。どちらでも指定しない場合、コンソールにこの入力を求めるプロンプトが出されます。	-databaseUserId または -dbu
dbPassword	ストリング	yourDBPassword	データベース・アクセスのためのパスワード。コマンド行引数で指定して、構成ファイルに保管しないようにすることができます。どちらでも指定しない場合、コンソールにこの入力を求めるプロンプトが出されます。	-databasePassword または -dbp
doFullName	ブール値	true	ロード時に単一フィールド・ソース名 (姓はあるが名はない) を特定の名および姓に解析するかどうかを制御します。	
parseThreshold	0 から 100 の整数	65	doFullName が true で単一フィールド名が解析される際に、ソース名に対する代替解析の使用を制御します。	
doCategorize	ブール値	false	カテゴリー (個人/組織) を持たない単一フィールド・ソース名がロード時に NameWorks によってカテゴリー化されるかどうかを制御します。	
gnrConfigFile	ストリング	/abc/gnr.config	解析と分析を制御する Global Name Management 構成ファイルのパス。	
numThreadsAnalyzer**	1 から 64 の整数	4	名前の分析に使用されるスレッドの数。パフォーマンスのために調整されます。	
numThreadsWriter**	1 から 64 の整数	4	データベースまたは出力ファイルに名前を書き込むために使用されるスレッドの数。パフォーマンスのために調整されます。	
dbBatchSize**	1 から 256 の整数	32	データベースに書き込むときに一時に処理される名前の数。パフォーマンスのために調整されます。	

表 17. ENS NameLoader 構成ファイルの項目 (続き)

NameLoader 構成ファイルのキー	型	代表的な値	備考	対応するコマンド行オプション
threadQueueSize**	1 から 9999 の整数	250	分析または出力のためにキューで保留される名前の数。パフォーマンスのために調整されます。	

* - 特定のリストに対する細かいアクセス制御で説明している細かいアクセス制御を使用する場合。101 から 120 までの範囲では十分でない場合は、このセクションの説明に従って範囲を拡大できます。

** - これらの各パフォーマンス・パラメーターの調整のヒントについては、インストール済み環境にある loader.config ファイルの対応するコメントを参照してください。

名前リスト

NameLoader 構成ファイルおよびコマンド行引数を使用して、名前リストについての次の情報を指定します。

- 名前リスト・コード
- 説明
- サーチャー・ロール
- マネージャー・ロール

指定された名前リスト・コードを持つ項目が ENS 名前リスト・テーブル (ens_name_list) 含まれる場合、その項目は更新されます。含まれない場合は追加されます。

Name Loader ユーティリティーを使用する名前リストからの名前のロード

新しい名前リストがあるとき、または名前リストへの更新があるときには、Name Loader ユーティリティーを使用してそれらの名前を処理し、データベース・スキーマにロードすることができます。既存の名前リストを更新する場合は、「addName」および「deleteName」Web サービスを使用することもできます。

このタスクについて

コマンド行で指定する引数により、デフォルト値や構成ファイルの設定値をオーバーライドすることができます。コマンド行で必要なのは構成ファイル名のみです。他のものはすべて構成ファイル内で指定できます。インストールが完了したら、Name Loader 構成ファイルのテンプレートが <install path>/data/loader.config にできています。それには、ENS データベース接続情報があらかじめ入っています。ただし、データベースのユーザー名とパスワードは例外で入っていません。

手順

コマンド行に以下のコマンドを入力して、Name Loader ユーティリティーを開始します。

UNIX/Linux

```
<install path>/bin/nameLoader <options> <configurationFileName>
```

Windows

```
<install path>%bin%nameLoader.bat <options> <configurationFileName>
```

例

すべてのオプションが含まれた Name Loader 使用状況のメッセージを取得するには、次のように入力します。

```
install_path/bin/nameLoader -usage
```

以下の出力を生成します。

```
Usage: nameLoader <options> <configurationFileName>
Options:
-in customers.csv          or -inFileName customers.csv
-nlc CUSTOMERS             or -nameListCode CUSTOMERS
-nld "Our customers"      or -nameListDescription "Our customers"

-sr searcher101           or -searcherRole searcher101
-mr manager101           or -managerRole manager101
-dbu myDatabaseUserid    or -dbUserid myDatabaseUserid
-dbp myDatabasePassword  or -dbPassword myDatabasePassword
-revtrack or -norevtrack or -revisionTracking or -noRevisionTracking or
  -autorevtrack or -autoRevisionTracking

-extendedInputFormat (if applicable)

-verbose or -trace or -quiet (at most one of these)
-load, -reload, -clear, or -clearAll (at most one of these; the default is -load)
```

コマンド行に指定する必要があるのは、**configurationFileName** のみです。他のものはすべて構成ファイル内で指定できます。また、デフォルトの構成ファイルを使用する場合は、コマンド行に構成ファイルを指定せずに、次のように **-nlc**、**-nld**、および **-in** 引数を指定することができます。

```
nameLoader -nlc "CUSTOMERS" -nld "Our customers" -in customers.csv
  -dbu mydbuserid -dbp mydbpassword
```

If you omit the database userid and password from both places, the program will prompt for them in the console. Actions:

- **-load** は、何も削除せずに、データベースに対して追加/更新を行います。
- **-reload** は、この名前リストのソース名を削除してから、ソース名と検索名を再追加します。
- **-clear** は、この名前リストのソース名を削除し、何も追加しません。
- **-clearAll** は、データベースからすべてのソース名、検索名、および名前リストを削除します。

構成ファイルに ENS データベース接続およびパフォーマンス・パラメーターの設定のみが含まれていると仮定した場合、ENS セルがダウン/オフの間に UNIX/Linux システム上で CUSTOMERS 名前リストをロードするコマンドの例は以下のようになります。

```
install_path/bin/nameLoader -load -in /CUSTOMERS.txt -nlc CUSTOMERS
  -nld "List with the names of all customers" -sr searcher120
  -mr manager120 <install path>/data/loader.config
```

ここで、各項目の意味は次のとおりです。

- **-load** は、実行するアクションを指定します。デフォルトはロードです。
- **-in <file path>** は、CSV 形式の名前リストのテキスト・ファイルへのパスを指定します。
- **-nlc CUSTOMERS** は、「name list code」(名前リストの短い ID) が「CUSTOMERS」であることを指定します。
- **-nld <description>** は、名前リストの、人間が読んで分かる形式の説明を指定します。
- **-sr <security role>** は、searcher120 というセキュリティー・ロールのユーザーすべてに対して、この名前リストの読み取り権限を付与します。ロールの形式は、searcher1 から searcher500 にする必要があります。
- **-mr <security role>** は、manager120 というセキュリティー・ロールのユーザーすべてに対して、この名前リストへの書き込み (管理) 権限を付与します。ロールの形式は、manager1 から manager500 にする必要があります。

以下の例は、ENS セルが稼働していない間に UNIX/Linux 上ですべての名前リストをクリアするものです。

```
<install path>/bin/nameLoader -clearAll <install path>/data/loader.config
```

以下の例は、ENS セルが稼働している間に UNIX/Linux 上で VENDORS 名前リストをロードするものです。

```
<install path>/bin/nameLoader -load -revtrack -in /VENDORS.txt -nlc VENDORS  
-nld "List with the names of all vendors" -sr searcher150  
-mr manager150 <install path>/data/loader.config
```

-revtrack は、名前の挿入について改訂のトラッキングをオンにします。この機能を使用すると、稼働中の ENS セル内のサーチャーが、新しく挿入された名前を検出し、インメモリーの名前リスト・パーティションにそれらを順次追加できます。これにより、システムの稼働中に名前リストからの名前の追加が可能になります。これが指定されていない場合、ENS セルの稼働中に Name Loader が実行されても、ENS はセルが再始動されるまで新しい名前を検出しません。

データベース統計の更新

NameLoader を使用して名前をデータベースの ENS テーブルにロードしたら、パフォーマンスを最適化するために、それらの名前が含まれているテーブルの統計をデータベース・コマンドを使用して更新することが重要です。これにより、データベースによってそれらのテーブルへのアクセス方法が最適化され、ENS のパフォーマンスが劇的に変わる可能性があります。

このタスクについて

これは、名前をロードした後に実行します。一般に、200,000 個の名前を追加した後には統計を一度実行したら、それよりも大きなサイズになった後にもう一度統計を実行するようにすると効果的です (例えば、100 万個の名前を追加した後に一度実行した場合、次は 500 万個を追加した後に実行するようにします)。以下にリストする DB2 コマンドは、NameLoader が実行中でも実行可能です。

これは、通常は DBA が行う作業です。自分のシステム (デモ・システムなど) を使用している場合は、(スキーマ名を「ENS」と想定すると) 次のような DB2 コマンドになります。

```
RUNSTATS on table ENS.ENS_SEARCH_NAME ON KEY COLUMNS;  
RUNSTATS on table ENS.ENS_SOURCE_NAME ON KEY COLUMNS;  
RUNSTATS on table ENS.ENS_SEARCH_SOURCE_NAME ON KEY COLUMNS;  
RUNSTATS on table ENS.ENS_SEARCH_NAME_ADDS ON KEY COLUMNS;  
RUNSTATS on table ENS.ENS_SEARCH_NAME_REVS ON KEY COLUMNS;  
COMMIT;
```

Oracle の場合は、DBMS_STATS です。詳しくは、http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_stats.htm#i103646 を参照してください。

システムによっては、操作を最適化するために夜間に統計を自動実行するものがあります。偶然 ENS テーブルが空で名前が入っていないときにこの処理がシステムによって実行された場合は、その前に行われていた手動実行の統計の結果が効率的に元に戻されます。その場合は、上記のように、ロードした名前に対して統計を再実行する必要が生じる可能性があります。

Enterprise Name Search スキーマからの名前の削除

NameLoader ユーティリティでは、**-clearAll** を指定して Enterprise Name Search スキーマからすべての名前および名前リストを効率的に削除することも、**-clear** を指定して特定の名前リストのみを削除することもできます。

このタスクについて

-clearAll オプションは、すべての名前および名前リストを削除します。この操作を実行するには、セルが非アクティブでなければなりません。

```
nameLoader -clearAll
```

-clear オプションは、単一の名前を削除します。複数の名前リストのソース名が同じ検索名にマップされている場合があるので、**-clearAll** と比べて、この操作は複雑であり、本質的に非効率的です。

```
nameLoader -clear -nlc CUSTOMERS
```

第 6 章 IBM NameWorks の構成

IBM NameWorks がどのように機能するかは、デフォルト設定、オーバーライド、検索方針情報、およびその他の要素の指定によって影響を受けます。IBM NameWorks API を効率的に使用するためには、デフォルト設定と構成情報を IBM NameWorks 構成ファイルに指定するか、または Configuration() クラスを使用することによって、IBM NameWorks を構成します。

nw.config などの外部ファイルを使用するか、それとも Configuration() クラスを使用するかは、現在の構成と、処理のために使用可能な CPU の数に基づいて決めます。Configuration() クラスを使用すると、外部構成ファイルをメモリーに追加し、読み取り、処理する必要がないために、初期化の際のパフォーマンスが向上することがあります。ただし、この向上の程度は通常はわずかであり、構成の違いによってさまざまな結果になります。

IBM NameWorks 構成ファイル

システムの初期化中に読み取られる単一のテキスト・ファイル内にある構成設定を使用して、IBM NameWorks の構成情報を指定できます。このテキスト・ファイル内の情報は次の形式に従い、ブラケットで囲まれたセクション名はキー/値のペアのリストを分離します。

```
[Section name] Key=Value
```

セクション名およびキー名は大/小文字の区別がなく、値に含まれる文字に制限はありません。

IBM NameWorks プロセスでの検索の実行時に必要とされるスレッドの数の制限を制御できます。スレッド・プーリング・メカニズムはスレッド生成アクティビティをプールして、Embedded Search によって作成されるスレッドの数が、指定された制限を超えないようにします。この制限は、IBM NameWorks 構成ファイルの [General] セクションの MaxThreads= 項目で設定できます。

比較パラメーター・オーバーライドを指定するための推奨される方法は、各データ・リストに合わせてチューニングされたデフォルトの比較パラメーター・オーバーライド・ファイルを作成し、そのファイルをデータ・リストに適用してから、IBM NameWorks 構成ファイルの [Search Strategy] セクションで相対オーバーライドを使用することです。デフォルトのオーバーライドがデータ・リストに関連付けられた後には、照会ごとにオーバーライドを指定する必要はありません。

注: ターゲット・マシンで利用可能な CPU コアそれぞれに対して 1 つのスレッドを使用すると、IBM NameWorks のパフォーマンスを最適化できます。

Configuration() クラス

IBM NameWorks の Configuration() クラスを使用することにより、外部構成ファイルを使用する代わりに、デフォルト設定を指定してそれらをプログラムの一部と

してメモリー内に保持できます。このクラスを使用して、デフォルトの構成設定を指定してから、以下のいずれかのオプションを使用してオーバーライドを指定します。

- **Strategy()** クラスを使用して、データ・リストごとにチューニングされたデフォルトの比較パラメーターのセットを作成します。その後、`addGivenNameOverride()`、`addSurnameOverride()`、`addOrganizationNameOverride()` などのオーバーライド・メソッドを使用して、オーバーライドを実装します。
- **CompParmsOverrides()** クラスを使用して、個々の名、姓、および組織名の比較パラメーター・オーバーライドを `Scoring.search()` および `Scoring.compare()` メソッドに提供します。

Configuration() クラスを使用すると、IBM NameWorks 構成をデータベースに格納すること、および新しい構成や更新された構成を動的に指定することが可能になります。IBM NameWorks 構成をデータベースに保存して、別のアプリケーションを使用して IBM NameWorks のインスタンスの作成する場合には、このオプションを使用することが役立ちます。

IBM NameWorks 構成ファイルを使用した構成設定の指定

システムの初期化中に読み取られる構成ファイル内の設定を変更して、IBM NameWorks の構成情報を設定したり更新したりすることができます。このファイルは、UTF-8 エンコードを使用しているものと想定されています。

このタスクについて

このタスクは、通常、構成ファイルを変更してそれらの変更が有効になるようにアプリケーション・サーバーを再始動する、システム管理者が行います。IBM NameWorks 構成ファイルの各セクションには、異なるパラメーターが含まれています。構成ファイルの各セクションに含まれる変更可能なパラメーターについて詳しくは、このセクションのトピックを参照してください。

手順

1. IBM NameWorks 構成ファイルが存在するディレクトリーに移動します。

オプション	説明
IBM NameWorks をクライアント・アプリケーションで使用する場合	構成ファイルは、カスタム・クライアント・アプリケーション用に使用するディレクトリー内に存在します。
IBM NameWorks を Web サービスとして使用する場合	構成ファイルのデフォルトの場所は、 <code>install_dir/data</code> です。 <code>install_dir</code> は IBM InfoSphere Global Name Management をインストールしたディレクトリーです。

2. テキスト・エディターで構成ファイルを開き、必要な変更を行います。そこに含めるセクションとパラメーターは、構成に応じて異なります。
3. 構成ファイルを保存します。
4. 変更を有効にするには、クライアント・アプリケーションまたは Web サービスを停止して再始動します。

オプション	説明
IBM NameWorks をクライアント・アプリケーションで使用する場合	<ol style="list-style-type: none"> 1. クライアント・アプリケーションを停止します。 2. クライアント・アプリケーションを再始動します。
IBM NameWorks を Web サービスとして使用する場合	<ol style="list-style-type: none"> 1. IBM InfoSphere Global Name Management をインストールしたディレクトリーに移動します。 2. stopGNR コマンドを実行して、Web サービスを停止します。 3. startGNR コマンドを実行して、Web サービスを再始動します。

構成ファイルのサンプル

```
[General]
MaxThreads=4
CompParmsDefaults=/gnr/data/compparms.config

[Custom Tokens]
OR=GivenName
BARONESS=Title

[Datalist:Distributed]
Type=1
Server=localhost|4250|1|

[Datalist:Embedded]
Type=0
List=part1.csv
List=part2.csv
CompressedBitSig=true
TAQ=/gnr/data/taq.ibm
GNV=/gnr/data/gnv.ibm
SNV=/gnr/data/snv.ibm
PNREG=/gnr/data/angloRegRule.ibm,Anglo
PNREG=/gnr/data/swasianRegRule.ibm,Arabic
PNREG=/gnr/data/swasianRegRule.ibm,Pakistani
PNREG=/gnr/data/russianRegRule.ibm,Russian

[DateCompare]
EII=93

[Strategy:Broad]
[GNParms:Broad]
ANCHOR_FACTOR=0.95
COMPRESSED_SCORE_MAX=1.00
DO_COMPRESSED_SCORE=Y
FIELD_THRESH=0.50
FIELD_WEIGHT=0.40
INITIAL_INITIAL_SCORE=0.70
INITIAL_TOKEN_SCORE=0.75
MATCH_INITIALS=Y
OOPS_FACTOR=0.95
[SNParms:Broad]
ANCHOR_FACTOR=0.95
COMPRESSED_SCORE_MAX=1.00
DO_COMPRESSED_SCORE=Y
FIELD_WEIGHT=0.60
NAME_UNKNOWN_SCORE=0.50
```

```

OOPS_FACTOR=0.95

[Strategy:Narrow]
[GNParms:Narrow]
ANCHOR_FACTOR=0.85
COMPRESSED_SCORE_MAX=0.00
DO_COMPRESSED_SCORE=Y
FIELD_THRESH=0.60
FIELD_WEIGHT=0.40
INITIAL_INITIAL_SCORE=0.65
INITIAL_TOKEN_SCORE=0.70
NAME_UNKNOWN_SCORE=0.50
OOPS_FACTOR=0.85
[SNParms:Narrow]
ANCHOR_FACTOR=0.85
COMPRESSED_SCORE_MAX=1.00
DO_COMPRESSED_SCORE=Y
FIELD_WEIGHT=0.60
NAME_UNKNOWN_SCORE=0.40
OOPS_FACTOR=0.85

[Strategy:Standard]
[GNParms:Standard]
[SNParms:Standard]
[ONParms:Standard]

[Transliteration Modules]
Module=/gnr/data/arabicTransRule.ibm
Module=/gnr/data/cyrillicTransRule.ibm

[Reference Files]
NameSifter=/gnr/data/SifterRules.ibm

[Comparison Files]
TAQ=/gnr/data/taq.ibm
GNV=/gnr/data/gnv.ibm
SNV=/gnr/data/snv.ibm
PNREG=/gnr/data/angloRegRule.ibm,Anglo
PNREG=/gnr/data/swasianRegRule.ibm,Arabic
PNREG=/gnr/data/swasianRegRule.ibm,Pakistani
PNREG=/gnr/data/russianRegRule.ibm,Russian

```

構成ファイルの **General** セクション

[General] セクションには、すべてのフィーチャーで使用可能な情報が含まれません。構成ファイルのこのセクションに、デフォルトの比較パラメーター (CompParms) のスレッド・プーリングおよびオーバーライドのサポートがリストされます。

構成ファイルの [General] セクションで、以下のフィールドを構成可能です。

```

[General]
MaxThreads=
CompParmsDefaults=
DefaultAltScoreFactor=

```

MaxThreads=*n*

同時検索操作をサポートするために使用されるスレッドの最大数を示します。この値は、外部検索エンジンと組み込み検索の両方に適用され、ゼロより大か等しい (\geq) 正整数を指定できます。ただし、プロセッサ・コアの数より大きい値は、パフォーマンスが低下する場合があります。デフォルト値は **MaxThreads=0**

で、個別の検索スレッドを使用しないことを示します。指定された値がゼロより小さい場合は、無効パラメーター例外 (GODW031E) が生成され、無効な値はエラー条件になります。

CompParmsDefaults=オーバーライドを含むファイル名

デフォルトの比較パラメーターのオーバーライドを含むファイルの名前を指定します。ファイルの形式は、Distributed Search で使用するものと同じです。デフォルト値は空白で、オーバーライド・ファイルを使用しないことを示します。無効なファイル名または無効なデータが指定されている場合、不正なデフォルトの比較パラメーター・エラー (GODW035E) が生成されます。

DefaultAltScoreFactor=factor

代替解析で見つかった一致の最終類似性スコアに適用されるデフォルトの係数を指定します。この値は、特定の Datalist 用にオーバーライドすることができます。値は正の実数 (浮動小数点) 値で、1.0 以下であることが必要です。そうでない場合、無効な値が検出されるとき、無効パラメーター例外 (GOD031E) が生成されます。

構成ファイルの Custom Tokens セクション

構成ファイルの [Custom Tokens] セクションには、カスタム解析トークンのタイプがリストされています。

カスタム解析トークンは、[Custom Tokens] セクションに次の形式でリストされます。

```
[Custom Tokens]
token=type [,comment ]
```

token

カスタム・トークンのテキスト。

type

次のタイプのいずれかのトークン・タイプ。

名 通常は名として使用される John や Michael などの通常の名前トークン。このタイプのトークンは、姓としても出現する場合がありますが、名の可能性が 4 倍として処理されます。

姓 通常は姓として使用される McGillicuddy や Wiltshire などの通常の名前トークン。このタイプのトークンは、名としても出現する場合がありますが、姓の可能性が 4 倍として処理されます。

Initial

ローマ数字またはその他のトークン・タイプの代わりに、イニシャルとして処理される単一文字。

Title 通常は社会的地位を反映し、一般に他の名前トークンの前に使用されるストリング。

Prefix 後続の名前の語幹トークンと同じ名前の句に含まれている小辞。de や la などのトークンは接頭部です。

Suffix

先行する名前の語幹トークンと同じ名前の句に含まれている小辞。aldeen などのトークンは接尾部です。

Qualifier

通常は世代関係や社会的な地位を示す修飾子。 *Jr.* や *Esq.* などのトークンは修飾子です。修飾子は名フィールドまたは姓フィールドのいずれにも含まれません。

Conjunction

複数の名前を結び付ける *and* などの単語。

comment

トークンのオプションの記述。

構成ファイルの **Datalist** セクション

データ・リストの記述は、各 `[Datalist:name]` セクションに格納されます。ここで、*name* 値は、`search()` メソッド呼び出しの *datalists* 引数に渡されるデータ・リストの名前を表します。

データ・リストごとのオーバーライドは、組み込みデータ・リストだけに適用されます。1 つ以上の `CompParmsDefaults=` 項目を構成ファイルの `[Datalist:]` セクションに追加することにより、個別のデータ・リストの比較パラメーター・オーバーライドを指定できます。このオプションを使用すると特定のデータ・リストにデフォルト値を設定できますが、`CompParmOverrides` クラスで示される検索方針とオーバーライドでは個別の照会のデフォルト値だけがオーバーライドされます。

構成ファイル内の **Distributed Search** 情報

Distributed Search 項目は、構成ファイルの `[Datalist:name]` セクションによってサポートされ、Distributed Search が有効の場合に使用されます。

以下のパラメーターが、Distributed Search の検索に対して IBM NameWorks 構成ファイルの `[Datalist:name]` セクションに適用されます。

複数の `Server=` 項目を指定でき、指定されたデータ・リストに複数のサーバーが関連付けられたことを示します。このため、照会がフェデレート (連合) され、複数の結果が累積される必要があります。追加フラグが `Server=` 項目に関連付けられている場合、そのサーバーが `addName()` 操作で使用されます。1 つのデータ・リストにつき、1 つのサーバーのみを追加フラグで構成できます。

```
[Datalist:Distributed]
Type=1
Server=host|port|listname[|add]
IncludeTAQs=
```

Type=*n*

このデータ・リストが、全検索 (`Type=1`) または固有の名前 (`Type=2`) のタイプのいずれであるかを示します。`Type=n` 項目が見つからない場合、データ・リストは全検索タイプであると見なされます。

host

ホスト・マシンの IP アドレス (シンボル名または数値のいずれか)。IP アドレスが (`Server=[host]|port` のように) 大括弧で囲まれている場合、Distributed Search エンジンとの通信には IPv6 プロトコルが使用されます (エンジンも IPv6 プロトコルを使用するように構成する必要があります)。

port

10 進数の IP ポート・アドレス (範囲は 0:65535)

listname

Distributed Search インスタンス内で使用される内部データ・リスト名。このパラメーターは現在使用されていません。

add

新しい名前のレコードをデータ・リストに追加するときには、このサーバーを使用する必要があることを示します。 *add* オプションの指定されていないサーバーはすべて、読み取り専用として扱われます。

IncludeTAQs=

データ・リストの項目に称号および修飾子の値を含める必要があるかどうかを示す単一項目。 *IncludeTAQs=true* を指定することによって、追加、更新、および検索操作中にデータ・リストの項目に適用するように、IBM NameWorks 構成ファイル内にこの値を指定できます。

注: TAQ 情報が含まれている場合、称号が名フィールドに追加され、修飾子が姓フィールドに追加されてから、名前データが検索操作またはペアワイズ比較操作で使用されます。

構成ファイル内の Embedded Search 情報

Embedded Search 項目は、構成ファイルの `[Datalist:name]` セクションによってサポートされ、Embedded Search が有効の場合に使用されます。無効な値は、すべてのパラメーターでエラー条件になります。

`search()` メソッドで組み込みデータ・リストを検索するときは、以下のパラメーターが適用可能です。項目名はすべて、大文字、小文字、または大/小文字混合の形式で指定できます。

`[Datalist:Embedded]`

Type=0

List=

CompressedBitSig=

TAQ=

GNV=

SNV=

ONV=

PNREG=

ONREG=

ONTERM=

ONTOPN=

NativeTaq=

NativePnVar=

NativeOnVar=

NativeOnReg=

NativeOnReg=

Type=n

Embedded Search データ・リストは、構成ファイルの `Datalist` セクション内の `Type=0` 項目によって識別されます。この項目のデフォルト値は、`Type=0` (Embedded Search データ・リスト) です。 `Type=0` のデータ・リストに関連付けられている `Server=` 項目は無視されます。

List=名前レコードを含むファイルの名前

データ・リストに関連付けられている個別の名前リストは、構成ファイルの

Datalist セクション内の List= 項目によって識別されます。複数の List= 項目が受け入れられ、各項目はデータ・リストに関連付けられた名前の個別のリストとして処理されます。名前リスト・ファイルは、.csv 形式で指定する必要があります。固有の追加リストは、ファイル名にストリング |add を追加することによって指定できます。追加リストは、名前リストがロードされた後で追加された名前を受け入れます。Embedded Search のデータ・リストの記述に List= 項目が含まれていない場合、単一の空の名前のリストが作成され、名前を追加できます。空の名前リストは、追加リストとしてマークが付けられます。

このパラメーターで無効なファイル名が指定されている場合、不正データ・ファイル・エラー (GODW032E) が生成されます。

CompressedBitSig=n

ビット・シグニチャーを名前の圧縮形式に含める必要があるかどうかを決定します。この値は 0 または 1 のいずれかで、デフォルト値は 1 です。

無効な値が指定された場合、無効パラメーター値エラー (GODW031E) が生成されます。

TAQ=TAQ リストのパス名

1 つ以上の TAQ 指定変更ファイルを、データ・リストに関連付けることができます。TAQ 指定変更ファイルは、データ・リストに関連付けられている各名前リストに適用されます。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

GNV= | SNV= | ONV=変形リストのパス名

1 つ以上の変形ファイルを、データ・リストに関連付けることができます。変形ファイルは、データ・リストに関連付けられている各名前リストに適用されます。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

PNREG= | ONREG=正規化規則ファイルのパス名、国/地域別情報名

1 つ以上の正規化ファイルを、データ・リストに関連付けることができます。正規化ファイルは、データ・リストに関連付けられている各名前リストに適用されます。この項目の形式は、IBM NameWorks 構成ファイルの [Comparison Files] セクションの類似する項目と同じです。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

ONTERM=用語ファイルのパス名

1 つ以上の用語ファイルを、データ・リストに関連付けることができます。

用語ファイルにブランクの用語がある場合、ブランクの用語テキスト・データ・リスト・エラー (GODH065E) が生成されます。ファイル内の用語に許可されない文字が使用されている可能性もあります。

ONTOPN=*n*

個人名に対して組織名を検索するための追加のサポートを含める必要があるかどうかを決定します。この値は 0 または 1 のいずれかで、デフォルト値は 0 です。

無効な値が指定された場合、無効パラメーター値エラー (GODW031E) が生成されます。

NativeTaq=ネイティブ文字 TAQ リストのパス名

1 つ以上のネイティブ文字 TAQ 指定変更ファイルを、データ・リストに関連付けることができます。ネイティブ文字 TAQ 指定変更ファイルは、データ・リストに関連付けられている各名前リストに適用されます。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

NativePnVar=ネイティブ文字個人名の変形リストのパス名

1 つ以上のネイティブ文字個人名の変形ファイルを、データ・リストに関連付けることができます。変形ファイルは、データ・リストに関連付けられている各名前リストに適用されます。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

NativeOnVar=ネイティブ文字組織名の変形リストのパス名

1 つ以上のネイティブ文字組織名の変形ファイルを、データ・リストに関連付けることができます。変形ファイルは、データ・リストに関連付けられている各名前リストに適用されます。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

NativePnReg=ネイティブ文字個人名の正規化ファイルのパス名、文字名

1 つ以上のネイティブ文字個人名の正規化ファイルを、データ・リストに関連付けることができます。正規化ファイルは、データ・リストに関連付けられている各名前リストに適用されます。この項目の形式は、IBM NameWorks 構成ファイルの [Comparison Files] セクションの類似する項目と同じです。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

NativeOnReg=ネイティブ文字組織名の正規化ファイルのパス名、文字名

1 つ以上のネイティブ文字組織名の正規化ファイルを、データ・リストに関連付けることができます。正規化ファイルは、データ・リストに関連付けられている各名前リストに適用されます。この項目の形式は、IBM NameWorks 構成ファイルの [Comparison Files] セクションの類似する項目と同じです。

無効なファイル名が指定されている場合、またはファイルの内容をロードできない場合は、NameHunter データ・リスト・エラー (GODW037E) が生成されます。

構成ファイルの検索方針セクション

検索方針に関する情報は、構成ファイルの複数のセクションに格納されます。個人名の個別のパラメーターは **GNParms** セクションおよび **SNParms** セクションに、組織名のパラメーターは **ONParms** セクションに設定できます。

```
[Strategy:name]
  GNCulture=
  SNCulture=
  ONCulture=
  MinScore=
  MaxReplies=
  SearchOpt=
  IncludeTAQs=
```

```
[GNParms:name]
```

```
...
```

```
[SNParms:name]
```

```
...
```

```
[ONParms:name]
```

```
...
```

name

値 *name* は、検索方針の名前を参照します。これらのセクションには、名、姓、および組織名の比較パラメーターが、NameHunter Distributed Search が想定する *name=value* 形式で含まれます。関連付けられた **[Strategy:name]** セクションが存在しない場合、これらのセクションは無視されます。

GNCulture | SNCulture

名および姓に使用する必要がある国/地域別情報コードを示します。有効値は、-1:20 の範囲です。これらの項目のいずれかが存在しない場合、それぞれの値はデフォルトで -1 に設定されます。

ONCulture

組織名に使用する必要がある国/地域別情報コードを示します。この値はサポートされていますが、現在は使用されていません。

MinScore

返される一致に関する最小の名前スコアの値 (0:100 の範囲内)。この数値は最上位の一致のフィルターであり、氏名のスコアによってソートされます。この値が -1 の場合、IBM NameWorks は指定された検索方針の中で **minScore=** オーバーライド項目を確認し、指定されている場合はその値を使用します。オーバーライド値が指定されていない場合、指定された国/地域別情報の NameHunter のデフォルト値が使用されます。

MaxReplies

返される一致の最大数。この数値は最上位の一致のフィルターであり、氏名のスコアによってソートされます。この値が -1 の場合、IBM NameWorks は指定された検索方針の中で「**MaxReplies=**」オーバーライド項目を確認し、指定されている場合はその値を使用します。オーバーライドが指定されていない場合、一致の数は制限されません。

SearchOpt

検索する名前リストのタイプを指定します。この値が 0 の場合、IBM NameWorks はすべての名前リストを検索します (SearchOpt=3)。

- 1 = 個人名リストのみの検索

- 2 = 組織名リストだけを検索
- 3 = 個人名と組織名の両方のリストの検索

IncludeTAQs=

データ・リストの項目に称号および修飾子の値を含める必要があるかどうかを示す単一項目。 IncludeTAQs=true を指定することによって、追加、更新、および検索操作中にデータ・リストの項目に適用するように、IBM NameWorks 構成ファイル内にこの値を指定できます。

注: TAQ 情報が含まれている場合、称号が名フィールドに追加され、修飾子が姓フィールドに追加されてから、名前データが検索操作またはペアワイズ比較操作で使用されます。

Index

NameHunter の索引を検索時に使用する必要があるかどうかを示します。有効値は Y または N のいずれかです。この項目が存在しない場合、値はデフォルトで Y になります。

[GNParms:name] | [SNParms:name] | [ONParms:name]

構成ファイルの日付比較セクション

[DateCompare] セクションの値を使用して、さまざまな日付比較テストのスコアの値をオーバーライドできます。

オーバーライドは、次の形式で表すことができます。

[DateCompare]
key=value

key

以下の表から取得される名前値。無効なキー値は無視されます。

value

有効値は、1:100 の範囲の数値にする必要があります。この範囲外の値は無視されます。

表 18. 日付比較のオーバーライドおよびその説明

キー	説明	デフォルトのスコア
ESS	Y=Y、M および D を入れ替え	99
EET	Y=Y、M=M、D 桁を入れ替え	98
ETE	Y=Y、M 桁を入れ替え、D=D	97
ETT	Y=Y、M 桁を入れ替え、D 桁を入れ替え	96
E EI	Y=Y、M=M、D を無視	95
EIE	Y=Y、M を無視、D=D	94
ETI	Y=Y、M 桁を入れ替え、D を無視	93
EIT	Y=Y、M を無視、D 桁を入れ替え	92
VEE	Y +/-5、M=M、D=D	91
VET	Y +/-5、M=M、D 桁を入れ替え	90
VTT	Y +/-5、M 桁を入れ替え、D 桁を入れ替え	89
EII	Y=Y、M を無視、D を無視	88
TEE	Y 桁を入れ替え、M=M、D=D	87

表 18. 日付比較のオーバーライドおよびその説明 (続き)

キー	説明	デフォルトのスコア
TET	Y 桁を入れ替え、M=M、D 桁を入れ替え	86
TTE	Y 桁を入れ替え、M 桁を入れ替え、D=D	85
TTT	Y 桁を入れ替え、M 桁を入れ替え、D 桁を入れ替え	84
TII	Y 桁を入れ替え、M を無視、D を無視 (x)	83
VII	Y +/-5、M を無視、D を無視	82
XEE	Y +/-10、M=M、D=D	81
XET	Y +/-10、M=M、D 桁を入れ替え (x)	80
XTE	Y +/-10、M 桁を入れ替え、D=D	79
XTT	Y +/-10、M 桁を入れ替え、D 桁を入れ替え	78
XII	Y +/-10、M を無視、D を無視	77
OB1	日付 +/- 1 日	76
OB2	日付 +/- 2 日	75
OB3	日付 +/- 3 日	74
OB4	日付 +/- 4 日	73
OB5	日付 +/- 5 日	72

構成ファイルの **Transliteration Modules** セクション

Transliteration Modules セクションには、どの音訳モジュールがインストールされているかがリストされます。

インストールされた音訳モジュールは、次の形式で [Transliteration Modules] セクションにリストされます。

[Transliteration Modules]
Module=音訳モジュールのパス名

以下の音訳モジュールは、IBM InfoSphere Global Name Management で有効に使用できます。

- arabicTransRule.ibm
- chineseTransRule.ibm
- cyrillicTransRule.ibm
- greekTransRule.ibm
- japaneseTransRule.ibm
- koreanTransRule.ibm
- latinTransRule.ibm
- chinese0nTransRule.ibm
- cyrillic0nTransRule.ibm
- hindi0nTransRule.ibm
- japanese0nTransRule.ibm
- korean0nTransRule.ibm
- anyTransRule.ibm*

注: anyTransRule.ibm は、例外的なケースのためのフォールバックとして追加された音訳モジュールです。anyTransRule.ibm を使用すると、サポート対象外の文字が使用されているときの例外を回避できます。これは、通常のインストールでは推奨されません。

あるモジュールの規則が別のモジュールの規則に干渉しないように、音訳モジュールは構成ファイルに以下の順序で指定されている必要があります。

1. 個人名の音訳ファイル
2. 組織名の音訳ファイル
3. anyTransRule.ibm

構成ファイルの **Reference Files** セクション

参照データ・ファイルの場所は、[Reference Files] セクションにリストされています。

構成ファイルの Reference Files セクションでは、以下の項目がサポートされています。

```
[Reference Files]
NameSifter=path name
CustomCultures=path name
```

path name

セミコロン (;) で区切られた、NameSifter 規則ファイルのリストの絶対パス名。Unix マシンでは、コロン (:) もサポートされています。

いずれかのカスタム国/地域別情報 (.cc) 規則ファイルの絶対パス名。他のタイプのファイルも含まれるディレクトリーを使用している場合、パスにファイル名を含めます。例: CustomCultures=<path name>/italian.cc

構成ファイルの **Comparison Files** セクション

NameHunter のサポート・ファイルの場所は、[Comparison Files] セクションにリストされています。

compare() メソッドを使用してペアワイズ比較を実行する場合、以下のパラメーターが適用可能です。このセクションはオプションですが、特定のファイル名を指定する場合は、ONREG および PNREG を属性として使用できます。各フォームが、NameWorks 構成ファイルの [Comparison Files] セクションでサポートされます。

```
[Comparison Files]
TAQ=taq.ibm
GNV=gnv.ibm
SNV=snv.ibm
ONV=onv.ibm
ONTERM=terms.ibm
PNREG=angloRegRule.ibm,Anglo
PNREG=swasianRegRule.ibm,Arabic
PNREG=chineseRegRule.ibm,Chinese
PNREG=frenchRegRule.ibm,French
PNREG=germanRegRule.ibm,German
PNREG=hispanicRegRule.ibm,Hispanic
PNREG=indianRegRule.ibm,Indian
PNREG=koreanRegRule.ibm,Korean
PNREG=swasianRegRule.ibm,Pakistani
PNREG=polishRegRule.ibm,Polish
PNREG=portugueseRegRule.ibm,Portuguese
PNREG=russianRegRule.ibm,Russian
```

PNREG=thaiRegRule.ibm,Thai
PNREG=turkishRegRule.ibm,Turkish
PNREG=swasianRegRule.ibm,SouthwestAsian
ONREG=genericOnRegRule.ibm,Ambiguous
ONREG=angloOnRegRule.ibm,Anglo
ONREG=chineseOnRegRule.ibm,Chinese
ONREG=hispanicOnRegRule.ibm,Hispanic
ONREG=koreanOnRegRule.ibm,Korean
ONREG=polishOnRegRule.ibm,Polish
ONREG=portugueseOnRegRule.ibm,Portuguese
ONREG=russianOnRegRule.ibm,Russian
NativeTaq=ctaq.ibm
NativePnVar=cnv.ibm
NativeOnVar=conv.ibm
NativeOnReg=chineseOnRegRule.ibm,Hanzi
NativeOnReg=japaneseOnRegRule.ibm,Kanji

TAQ=file_path_name

TAQ リストのパス名。

GNV=file_path_name

名の変形リストのパス名。

SNV=file_path_name

姓の変形リストのパス名。

ONV=file_path_name

組織名の変形リストのパス名。

ONTERM=file_path_name

用語リストのパス名。

ONREG=file_path_name,culture_name

正規化規則ファイルおよび国/地域名のパス名。有効な国/地域名は下にリストされています。

PNREG=file_path_name,culture_name

正規化規則ファイルおよび国/地域名のパス名。有効な国/地域名は下にリストされています。

NativeTaq=file_path_name

ネイティブ文字 TAQ リストのパス名。

NativePnVar=file_path_name

ネイティブ文字個人名の変形リストのパス名。

NativeOnVar=file_path_name

ネイティブ文字組織名の変形リストのパス名。

NativePnReg=file_path_name,script_name

ネイティブ文字個人名の正規化規則ファイルのパス名と関連した文字名。有効な文字名は下にリストされています。

NativeOnReg=file_path_name,script_name

ネイティブ文字個人名の正規化規則ファイルのパス名と関連した文字名。有効な文字名は下にリストされています。

国/地域名には、以下のいずれかの値を指定できます。

- アフガニスタン語
- 英語

- アラビア語
- 中国語
- ペルシア語
- フランス語
- 汎用 (または未確定)
- ドイツ語
- ヒスパニック
- インド語
- インドネシア語
- 日本語
- 韓国語
- パキスタン語
- ポーランド語
- ポルトガル語
- ロシア語
- タイ語
- トルコ語
- ベトナム語
- ヨルバ語
- -----
- ヨーロッパ
- ハン
- 南西アジア
- -----
- カスタムの国/地域別情報 ID (Custom1..Custom20)

文字名には、以下のいずれかの値を指定できます。

- 漢字 (中国語)
- 漢字 (日本語)
- デーバナーガリー文字
- キリル文字
- ローマ字
- ハングル
- アラビア文字
- ギリシャ文字

Configuration クラスを使用した構成設定の指定

Configuration クラスを使用するプログラムを記述することにより、IBM NameWorks の構成情報を設定および更新できます。このクラスを使用すると、デフォルトの構成設定を指定して、CompParmsOverrides クラスを使用することによりオーバーライドを動的に指定できます。

このタスクについて

このタスクは、通常、IBM NameWorks API に関する作業を直接行うプログラム開発者が行います。 Configuration クラス、そのコンストラクター、および関連する他のメソッドの使用方法について詳しくは、API Reference の資料を参照してください。

注: 作成するプログラムは、クライアント・アプリケーションの必要と、 Configuration クラスのメソッドの使用方法とに基づいて、さまざまに異なります。この情報の目的は、IBM NameWorks 構成ファイルの代わりに Configuration クラスを使用するために必要な手順の概要を示すことです。以下の手順には Java のコード例が含まれていますが、C++ でプログラムを作成するためのステップもこれと同じです。

手順

1. 開発アプリケーションを使用して、 Configuration オブジェクトを作成します。以下のコード行は、空の Configuration オブジェクトを作成します。続くステップは、 Configuration オブジェクトの一部である追加情報を例示しています。

```
Configuration configuration = new Configuration();
```
2. Datalist オブジェクトを作成して、検索に使用するデータ・リストを指定します。次のサンプルは、 Customers という名前のデータ・リストを含む Configuration オブジェクトを作成し、検索の際に TAQ を含めないように指定します。検索のタイプに応じて、リスト項目をデータ・リストに追加するなど、追加のパラメーターを指定できます。

```
Configuration configuration = new Configuration();
Datalist customers = configuration.addDatalist("Customers");
customers.setIncludeTaqs(false);
```
3. Strategy クラスを使用して Strategy オブジェクトを作成します。この検索方針は、ペアワイズ比較と検索で使用するためにプログラムから呼び出します。次のサンプルは、 Broad という検索方針を作成し、複数の変数を指定します。

```
Strategy broad = configuration.addStrategy("Broad");
broad.setMinScore(75)
    .setMaxReply(1000)
    .setSearchOptions(EnumSet.of(NameCategory.PERSONAL));
```
4. 検索で使用するスレッドの最大数やデフォルトの比較パラメーター・オーバーライドを含むファイル名など、構成データを指定するためのさまざまなメソッドを呼び出します。

```
configuration.setMaxThreads(8)
    .setDefaultCompParmsOverridesFile("compparms.config");
```
5. Configuration オブジェクトを Analytics() コンストラクター、 Scoring() コンストラクター、またはその両方に渡して、 Analytics オブジェクトと Scoring オブジェクトを作成します。これらのオブジェクトが作成されると、同じ Configuration オブジェクトを使用して Analytics オブジェクトや Scoring オブジェクトをさらに作成するのではなく、 Configuration オブジェクトは廃棄でき、それ以降の変更は効果がなくなります。

```
Analytics analytics = new Analytics(configuration);
Scoring scoring = new Scoring(configuration);
```

追加の音訳規則ファイルを使用するように IBM NameWorks 構成を更新する

IBM NameWorks は、規則ファイルを使用して、名前が音訳される方法を判別します。IBM NameWorks が追加の規則ファイルを使用するためには、その前に、システム管理者が IBM NameWorks 構成ファイルを変更する必要があります。バージョン 6.0 では、個人名のほかに組織名に固有の音訳規則のサポートが追加されています。組織名の音訳規則は、ファイル名に「On」が含まれています。

手順

1. IBM NameWorks 構成ファイルを変更して [Transliteration Modules] 見出しの下に規則ファイルを含めます。次の構文を使用します。

```
Module= full_path/rule_file_name
```

full_path は、絶対パスとディレクトリー名です。 */rule_file_name* は、使用する規則ファイルの特定の名前です。

以下の音訳規則ファイルは、IBM NameWorks で有効に使用できます。

arabicTransRule.ibm

アラビア文字で記述された個人名の音訳規則。

chineseTransRule.ibm

中国語の漢字で記述された個人名の音訳規則。

cyrillicTransRule.ibm

アラビア文字で記述された個人名の音訳規則。

greekTransRule.ibm

ギリシャ文字で記述された個人名の音訳規則。

hindiTransRule.ibm

デーバナーガリー文字で記述された個人名の音訳規則。

japaneseTransRule.ibm

日本語の仮名で記述された個人名の音訳規則。

koreanTransRule.ibm

ハングル文字で記述された個人名の音訳規則。

chineseOnTransRule.ibm

中国語の漢字で記述された組織名の音訳規則。

cyrillicOnTransRule.ibm

キリル文字で記述された組織名の音訳規則。

hindiOnTransRule.ibm

デーバナーガリー文字で記述された組織名の音訳規則。

japaneseOnTransRule.ibm

日本語の漢字と仮名で記述された組織名の音訳規則。

koreanOnTransRule.ibm

ハングル文字で記述された組織名の音訳規則。

anyTransRule.ibm

任意の文字を処理できる音訳規則の特殊なセット。ただし基本的な形式

の処理だけなので、名前の分析やスコアリングには適していません。この音訳規則のセットは、サポート対象外の文字またはその組み合わせによって音訳エラーが発生しないようにするための、フォールバックとしてのみ使用してください。このファイルは、その他の音訳モジュールが名前を処理できない場合の最終手段として使用されるように、常にリストの最後に指定してください。

2. 該当するサーバーを停止後再始動して **IBM NameWorks** を再初期化します。これにより、サーバーは更新済み構成ファイルの情報を使用するようになります。**IBM NameWorks Web** サービスのインストール済み環境には、このような操作のための停止コマンドと開始コマンドが含まれています。

例

例えば、C:¥ ドライブの ¥NW ディレクトリーにあるアラビア語の個人名の規則ファイルを含めるには、構成ファイルを次のように更新します。

```
[Transliteration Modules]  
Module=C:¥NW¥arabicTransRule.ibm
```

第 7 章 トラブルシューティングとサポート

IBM InfoSphere Global Name Management のインストール済み環境で問題を切り分けて解決するためには、トラブルシューティングとサポート情報を使用して、問題の原因を特定する方法、診断情報を収集する方法、フィックスの入手場所、および検索する知識ベースを決めます。

IBM サポートに連絡を取る必要がある場合、この情報を使用して、サービス技術者が問題解決の手助けをするために必要な診断情報を収集してください。

IBM InfoSphere Global Name Management のトラブルシューティング・チェックリスト

チェックリストに構造化された一連の質問に答えることにより、自分で問題の原因を特定したり問題の解決策を見つけたりできる場合があります。

以下の質問に答えることにより、IBM InfoSphere Global Name Management で発生している問題の原因を特定するのに役立つ場合があります。

1. この構成はサポートされていますか? ご使用のシステムが、ハードウェア、オペレーティング・システムおよびソフトウェアのすべての要件を満たすことを確認するため、システム要件を参照してください。
2. IBM InfoSphere Global Name Management の最新フィックスを適用しましたか? IBM InfoSphere Global Name Management のサポート・ポータルを参照してください。
3. 問題が発生したとき何を行っていましたか?

製品のインストール・プログラムから、**CD** または **DVD** が間違っているというプロンプトが出ましたか?

そうであれば、完全なディレクトリー構造がローカル・ハード・ディスクに存在していることを確認してください。インストール・プログラムの作動には、Disk1 ディレクトリーからの完全なディレクトリー構造がローカル・ハード・ディスク・ドライブに存在していなければなりません。ローカル・ハード・ディスクにインストール・プログラムしか存在しない場合には、**CD** または **DVD** からこのディレクトリー構造全体をコピーしてください。

1 つ以上のコンポーネントが正常にインストールされなかったとインストール・プログラムから通知が出されましたか?

そうであれば、インストール・ログ・ファイルを確認して問題を修正してください。その後インストール・プログラムを使用して、これらのコンポーネントを再インストールします。

IBM NameWorks を開始するときに **GODW033E** エラー・メッセージを受け取りましたか?

そうであれば、このエラー・メッセージは、名前データ・ファイル内の 1 つ以上の名前レコードが不適切にコード化されているために無効になっていることを示しています。エラー・メッセージには、問題に

なっているデータ・ファイルの名前が含まれています。そのデータ・ファイルを UTF8 フォーマットで保存してから IBM NameWorks を再開してみてください。

製品またはコンポーネントをシャットダウンする際に、新しい名前データが欠落していますか? **Distributed Search** を使用していますか?

そうであれば、製品またはコンポーネントの実行中に名前を検索リストに追加しましたか? 複数のサーバーを使用して **Distributed Search** で名前処理する場合、**ds.config** ファイルを確認して以下の設定を構成してあることを確認してください。

- **[commgr]** セクションの下で、**Searcher** の数 (**numSearchers=n**) を確認します。 *n* が、検索するリストの数に 1 つの専用 **Add Searcher** を加えたものに等しくなることを確認してください。
- 少なくとも 1 つの構成済み **Searcher** が **Add** 専用になっていることを確認します (少なくとも 1 つの **Searcher** が **doAdds=true** に設定されている)。

これらの設定を変更した場合、**Distributed Search** サーバーを再開します。詳しくは、インフォメーション・センターの『*Distributed Search* の構成ファイルおよび設定』を参照してください。

エラー・メッセージ **GODS054E** を受け取りましたか?

そうであれば、このエラー・メッセージを受け取る最も一般的な原因は、IBM NameWorks 構成ファイルの中で **NameSifter** データ・ファイル名が欠落しているか、指定パスが無効であるかのいずれかです。構成ファイルの **[Reference Files]** セクションの下を確認し、**NameSifter=** 設定に構成された値を確認してください。ファイル名およびパスが正確である場合には、Java ランタイムで多くのメモリーを消費しているために **NameSifter** データをロードするためのメモリーが足りないことが問題であるのかもしれませんが。メモリーが足りないことがこのエラーの原因であると思う場合には、**-Xmx128m** またはこれに類似したパラメーターを使用して十分なメモリーを確保してください。

NameSifter をオフにする場合には、**NameSifter=** 設定でファイル名と **PATH** ステートメントを削除します。この設定は IBM NameWorks 構成ファイルの **[Reference Files]** セクションの下にあります。

NameSifter をオフにする理由の例として、名前データに組織名が含まれないため、IBM NameWorks で名前データを個人と組織にカテゴリー化する必要がないという場合があります。さらに、名前のカテゴリーが個人名になるか組織名になるかが既に分かっている場合は、この項目を指定する必要があります。

4. コンポーネントのログ・ファイルに問題に関するメッセージが含まれているかどうかを確認しましたか?
5. 含まれている場合、どのエラー・メッセージまたはエラー・コードが出されましたか? 詳しくは、インフォメーション・センターの『メッセージおよびエラー・コードの情報』を参照してください。
6. 問題の解決に役立つ可能性がある製品知識ベースを確認しましたか?
7. 該当するこれらの各オプションを試してもまだ問題が解決しない場合は、IBM ソフトウェア・サポートにお問い合わせください。

コンポーネント API の C++ エラー・コード

IBM InfoSphere Global Name Management のコンポーネントにエラーが発生すると、数値エラー・コードが返されます。エラーが発生した場合、IBM InfoSphere Global Name Management 文書でエラー・コード番号を確認し、エラーのタイプ、その発生場所、およびその修正方法についての情報を取得します。

エラー・コードの構文

例外クラス `ibmgmr::Exception` を使用してエラー情報が報告されます。エラーは、データ、入力、または内部エラーの 3 つのカテゴリに分割され、これを使用してエラーの原因と重大度の間で区別できます。このクラスには、クライアント・アプリケーションで使用できるいくつかのメソッドが組み込まれています。テキスト情報は、`ibmgmr::Exception::text()` メソッドを呼び出すことで取得できます。整数値は、`ibmgmr::Exception::value()` メソッドを呼び出すことで取得できます。以下の例は、このタイプの例外における基本的な `catch` 節の外観を例示しています。

```
catch (ibmgmr::Exception & e)
{
    ibmgmr::Exception Type type = e.type(),
    char component = e.component(),
    int code = e.code(),
    std::string text = e.text(),
    reportComplexError(type, component, code, text),
};
```

ExceptionType type()

どのタイプのエラーが発生したかを記述する列挙子。 `type()` では、次の 3 つの異なる値が返される可能性があります。

Internal

内部エラー。続行できません。

Reference data

参照データ破損。続行できません。

Input

入力データが無効です。

component()

エラーが発生した IBM InfoSphere Global Name Management のコンポーネントを特定する 1 文字のコードが返されます。 `component()` 関数では、以下の値が有効です。

識別子	コンポーネント
A	Country of Association
C	NameClassifier
D	Distributed Search
H	NameHunter
I	グローバル・エラー ¹
N	NameClassifier Country of Association
P	NameParser
S	NameSifter
T	NameTransliterator
V	NameVariantGenerator
W	NameWorks

識別子
1

コンポーネント

グローバル・エラー (001 から 006) は GODInnnE として出現します。nnn は返される数字コードです。文字 I はグローバル・エラーを示し、これはどのコンポーネントからでも報告される可能性があります。例えば、エラー GODH002E は、NameHunter (H) で発生したファイル・オープン・エラー (002) を意味します。この同じエラーは、NameParser などの別のコンポーネントで発生する可能性があり、この場合、エラーは GODP002E と表示されます。エラー 001 から 006 の文書を参照するとき、数値エラーの前に置かれる 1 文字のコードを確認すると、エラーが発生したコンポーネントを特定できます。

code()

特定のエラー条件に関連付けられたエラー・コードが返されます。

const throw()

エラーを伴う可能性がある関連テキスト情報が返されます。

value() const

エラーに関連付けられている可能性がある整数値が返されます。

id() const throw() | wid() const throw()

エラー条件を特定するストリングが GODcnnnE のフォーマットで返されます。

GOD

IBM InfoSphere Global Name Management 製品に割り当てられる 3 文字のエラー識別接頭部。

c char component() 関数によって返される 1 文字のコンポーネント ID。

nnn

code() 関数によって返される数値エラー・コード。

E エラー・メッセージの標準 IBM インディケータ。

const throw()

エラー条件 ID と、それに関連付けられた整数値およびテキスト情報の両方が含まれるストリングが、シングル・スペース文字区切りで返されます。

参照データのエラー・コード

参照データのエラーは、参照データ・ファイルに破損があることを示しています。このエラーが再発する場合、クライアント・アプリケーションによって IBM InfoSphere Global Name Recognition の内部データ構造が上書きされていることを示しています。

GODD901E 無効な構成または無効なデータ

説明: NamePreprocessor 構成ファイル npp.config の中に無効なデータが存在します。多くの場合、このエラーは、固有でないデータ・リストを正規化しようとすることによって引き起こされます。固有データ・リストを作成しない限り正規化は不可能です。

システムの処置: GNR モジュールは ibmgnr::Exception 例外をスローします。

ユーザーの処置: エラー・メッセージを読んで、問題を

判別してください。その後、npp.config ファイルを開き、無効なデータ項目を修正してください。

GODS051E NameSifter データ・エラー

説明: 指定されたいずれかの規則ファイルが欠落しているか、規則ファイルを開こうとして入出力エラーが発生しました。

システムの処置: GNR モジュールは ibmgnr::Exception 例外をスローします。

ユーザーの処置: 規則ファイルが存在すること、およびファイル名のスペルが正しいことを確認してください。

GODS052E NameSifter データ・エラー

説明: 指定されたいずれかの規則ファイルに構文エラーが含まれています。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: 規則ファイルの構文エラーを訂正して、NameSifter に再び名前を渡してください。

GODS053E NameSifter データ・エラー

説明: NameSifter のコンストラクターの呼び出し元は、指定された規則ファイルの中に実際に存在しないメイン規則リスト名を指定しました。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: メイン規則リスト名が規則ファイルの中に存在することを確認してください。

GODS054E NameSifter データ・エラー

説明: NameAnalyzer.dat ファイルが欠落しているか、ファイルから読み取ろうとして入出力エラーが発生しました。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: NameAnalyzer.dat ファイルが存在すること、およびパス名が正しいことを確認してください。問題が解決しない場合は、該当するファイルの新しいコピーを取得します。

GODS055E NameSifter データ・エラー

説明: NameAnalyzer.dat ファイルが何らかの形で破損しています。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: NameAnalyzer.dat ファイルの新規コピーを取得してください。

GODT031E 規則ファイルを読み取ることができません

説明: 音訳ファイル (`xxxTransRule.ibm`) または正規化規則ファイル (`xxxRegRules.ibm`) が破損しています。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、`ibmgnr::Exception::value()` メソッドで取得できません。

ユーザーの処置: 適切な場所にファイルがあること、およびパス名が正しいことを確認してください。適切な場所にファイルを配置するか、間違ったパス名を修正します。

GODT032E ヘッダーにモジュール・プロパティ名が含まれていません

説明: モジュール名プロパティがヘッダー・ファイルに含まれていませんでした。このエラーは通常、音訳ファイル (`xxxTransRule.ibm`) または正規化規則ファイル (`xxxRegRules.ibm`) が破損したことを示します。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、`ibmgnr::Exception::value()` メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT033E 音訳規則の後にファイル読み取りエラーが発生しました

説明: 音訳ファイル (`xxxTransRule.ibm`) または正規化規則ファイル (`xxxRegRules.ibm`) が破損しています。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、`ibmgnr::Exception::value()` メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT034E Transliterator 規則の欠落

説明: Transliterator 規則が欠落しています。このエラーは通常、音訳ファイル (`xxxTransRule.ibm`) または正規化規則ファイル (`xxxRegRules.ibm`) が破損したことを示します。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、`ibmgnr::Exception::value()` メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT035E 最後の規則セットの名前が、
Transliterator-ID 属性で指定された名前と異なります

説明: 音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損しています。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgrr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT036E ヘッダー行にコロンがありません

説明: このエラーは通常、音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損したことを示します。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgrr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT037E 空のプロパティ名

説明: プロパティ名が空です。このエラーは通常、音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損したことを示します。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgrr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT038E 空のプロパティ値

説明: プロパティ値が空です。このエラーは通常、音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損したことを示します。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgrr::Exception::value() メソッドで取得できません。

ユーザーの処置: ファイルの新規コピーを取得します。

GODT039E 無効なプロパティ値

説明: 音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損しています。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgrr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT040E 不明なプロパティ名

説明: 音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損しています。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgrr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT041E 不明な規則セット・タイプ

説明: 規則セット・タイプが不明です。このエラーは通常、音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損したことを示します。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgrr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT042E Transliterator ID が指定されていません

説明: 音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損しています。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgmr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT043E 参照リスト規則のフィールドの欠落

説明: このエラーは通常、音訳ファイル (xxxTransRule.ibm) または正規化規則ファイル (xxxRegRules.ibm) が破損したことを示します。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgmr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT045E 無効な UTF-8 文字が検出されました

説明: NameTransliterator の入力ストリングに、無効な UTF-8 文字が含まれています。通常、このエラーは、入力ストリングのエンコード方式が UTF-8 以外のもになっていることを示しています。この例外は、一部のコンポーネントがファイルを読み取るために NT を呼び出し、ロードするためにファイル名にパスするときのみ報告されます。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgmr::Exception::value() メソッドで取得できません。

ユーザーの処置: 無効文字を除去して、NameTransliterator への入力ストリングの引き渡しを再試行してください。入力ストリングと文字位置は ibmgmr::Exception のテキスト・フィールドと値フィールド、および what() ストリングにも表示されます。

GODT046E 不明なエンコード ID

説明: 不明なエンコード ID が呼び出し元から NameTransliterator に渡されたことを示しています。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgmr::Exception::value() メソッドで取得できません。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT047E 不明な Transliterator ID

説明: 無効な Transliterator ID が NameTransliterator に渡されました。

システムの処置: GNR モジュールは ibmgmr::Exception 例外をスローします。

ユーザーの処置: 影響したファイルの新規コピーを取得します。

GODT048E 入力をトランスコードできません

説明: 入力ストリングが、指定された文字エンコードでなかったことを示しています。

システムの処置: 音訳ファイル内の行番号は例外で報告され、ibmgmr::Exception::value() メソッドで取得できません。

ユーザーの処置: NameTransliterator で想定されるエンコード形式のストリングが渡されることを確認してください。エンコード方式が未指定の場合は、プラットフォーム・デフォルトに一致するエンコード方式でなければなりません。

GODT049E 入力を音訳できません

説明: 入力名に含まれる 1 つ以上の文字が、選択された音訳モジュールでサポートされる文字の範囲外であるため、音訳できません。

システムの処置: サポートされない最初の文字を示す文字位置が例外で報告され、ibmgmr::Exception::value() メソッドを使ってこれを取得できます。

ユーザーの処置: 音訳モジュールがまだロードされていない場合、それをロードします。さらに、サポートされない文字が入力名に含まれる場合、すべて除去してください。

グローバル・エラー・コード

グローバル・エラーはさまざまなコンポーネントで発生し、名前分析の特定の局面に固有なものとはかぎりません。

グローバル・エラー (001 から 006) は GODI nnn E として出現します。 nnn は返される数字コードです。文字 I はグローバル・エラーを示し、これはどのコンポーネントからでも報告される可能性があります。例えば、エラー GODH002E は、NameHunter (H) で発生したファイル・オープン・エラー (002) を意味します。この同じエラーは、NameParser などの別のコンポーネントで発生する可能性があり、この場合、エラーは GODP002E と表示されます。エラー 001 から 006 の文書を参照するとき、数値エラーの前に置かれる 1 文字のコードを確認すると、エラーが発生したコンポーネントを特定できます。

GODI001E アサーション・エラー

説明: 名前を分類しようとして内部エラーが発生しました。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、またエラー発生時に何をしようとしていたかを必ず把握しておいてください。

キーワードを使って問題の原因を診断することができません。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラーに関連しているキーワードは何か、またエラー発生時に何をしようとしていたかを必ず把握しておいてください。

GODI002E ファイルを開くことができません

説明: 以下のいずれかの関数で、指定されたファイルを開くことができませんでした。ファイルが存在しない可能性があります。

- `ConfigHandler::load()`
- `NameHunter::loadFieldVariants()`
- `NameHunter::loadTaqS()`
- `NameHunter::loadRegRules()`
- `NameHunter::loadTransRules()`
- `NameHunter::loadVariants()`

システムの処置: NameHunter のいずれかの関数を介してエラー・メッセージが戻されます。例えば `NameHunter::fieldVariantError()` 関数は、`addFieldVariant` 関数が `false` を戻す場合に説明を戻します。

ユーザーの処置: ファイルが存在すること、およびファイルへの適切なアクセス許可があることを確認してください。

GODI004E 内部メソッド・エラー

説明: 名前を分類しようとして内部エラーが発生しました。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラー発生時に何をしようとしていたかを必ず把握しておいてください。

GODI005E NameAnalyzer.dat ファイルの欠落

説明: 指定された場所に `NameAnalyzer.dat` ファイルが見つかりませんでした。

システムの処置: GNR モジュールは `ibmgnr::Exception` 例外をスローします。

ユーザーの処置: 適切な場所にファイルがあること、およびパス名が正しいことを確認してください。適切な場所にファイルを配置するか、間違ったパス名を修正します。

GODI003E 内部分析エラー

説明: 内部エラーが発生したことを示します。このエラーを報告する例外については、`text()` 値に含まれるキ

GODI006E 異なる `NameAnalyzer.dat` ファイル・パスが指定されました

説明: オブジェクトを作成するために 1 つのファイル名がコンポーネントによって使用され、最初のオブジェクトがまだアクティブである間に別の名前を使って 2 番目のオブジェクトが作成されました。以下のコンポーネントによってこれらのオブジェクトが生成される可能性があります。

- NameParser
- Country of Association (COA)

入力エラー・コード

入力エラーは、入力ストリングが読めないフォーマットであることを示します。通常このエラーは、誤った UTF-8 文字シーケンスによって発生します。処理を続行するには、その前に入力ストリングを修正する必要があります。

GODH008E 構文解析後、GN および SN が空白です

説明: SearchList::add() 関数に空の名前が渡されました。

システムの処置: GNR モジュールは ibmgnr::Exception 例外をスローします。

ユーザーの処置: データ・リストに追加される項目に有効な名前を提供して、追加関数を再試行してください。

GODH009E 1 番目の名前が空白です

説明: NameHunter::addVariant() 関数は、変形ペアの 1 番目のトークンが空白であることを検出しました。

システムの処置: GNR モジュールは ibmgnr::Exception 例外をスローします。

ユーザーの処置: 以下の変形ファイルを検査して、空白のトークンを見つけてください。

- ibmGnvFile
- ibmSnvFile
- ibmBnvFile
- ibmFieldVarFile

その後、変形ペアの 1 番目のトークンの名前を入力する必要があります。

GODH010E 2 番目の名前が空白です

説明: NameHunter::addVariant() 関数は、変形ペアの 2 番目のトークンが空白であることを検出しました。

システムの処置: GNR モジュールは ibmgnr::Exception 例外をスローします。

ユーザーの処置: 以下の変形ファイルを検査して、空白

- NameClassifier COA
- NameVariantGenerator

システムの処置: GNR モジュールは ibmgnr::Exception 例外をスローします。

ユーザーの処置: NameAnalyzer.dat ファイル名が正しいこと、および別のコンポーネントに新しいファイル名が渡されていないことを確認してください。

空白のトークンを見つけてください。

- ibmGnvFile
- ibmSnvFile
- ibmBnvFile
- ibmFieldVarFile

その後、変形ペアの 2 番目のトークンの名前を入力する必要があります。

GODH011E 無効なグループ名

説明: フィールド変形項目に空のテキスト・フィールドが含まれています。

システムの処置: GNR モジュールは ibmgnr::Exception 例外をスローします。

ユーザーの処置: エラーが発生するファイルの新規コピーを取得してください。

GODH012E 無効なスコア (0 と 1 の間でなければなりません)

説明: NameHunter::addVariant() 関数は、無効な変形スコアを検出しました。

システムの処置: このエラー・コードは行番号とエラー情報を記録します。

NameHunter::ConfigHandler::errorList() メソッドを呼び出してこれらを取り出すことができます。

ユーザーの処置: 以下の変形ファイルを検査して、空白のトークンを見つけてください。

- ibmGnvFile
- ibmSnvFile
- ibmBnvFile
- ibmFieldVarFile

エラーが発生するファイルの新規コピーを取得してください。

GODH013E 重複する項目

説明: 以下のいずれかの関数が、重複する項目を検出しました。

- ConfigHandler::load()
- NameHunter::addTaq()
- NameHunter::loadTaqS()
- NameHunter::addVariants()
- NameHunter::loadVariants()

システムの処置: このエラー・コードは行番号とエラー情報を記録します。

NameHunter::ConfigHandler::errorList() メソッドを呼び出してこれらを取り出すことができます。

ユーザーの処置: エラーが発生するファイルの新規コピーを取得してください。

GODH014E 不明なフィールド・タイプ

説明: 無効なフィールド・タイプ (姓名のどちらでもない) が CompParms データ構造 (setDefault() または setParmsDefault()) に渡されました。

システムの処置: GNR モジュールは ibmgmr::Exception 例外をスローします。

ユーザーの処置: NameFieldType (NameConstants.h) 列挙で有効なフィールド・タイプを 1 つ入力します。有効なフィールド・タイプは次のとおりです。

- GivenName
- SurName
- OrgName

GODH015E 不明な国/地域別情報コード

説明: 無効な国/地域別情報コードが、以下のいずれかの関数に渡されました。

- CompParms::setDefault()
- NameHunter::loadFieldVariants()
- NameHunter::addTaq()
- NameHunter::loadTaqS()
- NameHunter::addVariant()
- NameHunter::loadVariants()

システムの処置: GNR モジュールは ibmgmr::Exception 例外をスローします。

ユーザーの処置: Culture (NameConstants.h) 列挙で有

効な国/地域別情報コードを 1 つ入力します。を参照してください。

GODH016E 不明な TAQ タイプ

説明: 不明な TAQ タイプが NameHunter::addTaq() に渡されました。有効な TAQ タイプは、NameConstants.h. 列挙の Tokentypes フィールドから派生します。

システムの処置: GNR モジュールは ibmgmr::Exception 例外をスローします。

ユーザーの処置: いずれかの有効なフィールド・タイプを入力してください。

GODH017E 空の、または欠落している構成ヘッダー

説明: ConfigHandler::load() 関数は有効なヘッダー (例えば [ParmsGnAnglo]) を検出できませんでした。構成ファイルを訂正してください。

システムの処置: このエラー・コードは行番号とエラー情報を記録します。

NameHunter::ConfigHandler::errorList() メソッドを呼び出してこれらを取り出すことができます。

ユーザーの処置: 有効なヘッダーを提供することにより、構成ファイルを訂正してください。

GODH018E タグと値の区切り文字 (=) を検出できませんでした

説明: ConfigHandler::load() 関数は、区切り文字 (通常は等号 =) のない値の組を検出しました。

システムの処置: このエラー・コードは行番号とエラー情報を記録します。

NameHunter::ConfigHandler::errorList() メソッドを呼び出してこれらを取り出すことができます。

ユーザーの処置: 有効な区切り文字 (=) を提供することにより、構成ファイルを訂正してください。

GODH019E 参照ファイルの中でバージョン・ヘッダーが検出されませんでした

説明: 無効な、または古い TAQ ファイルが、ロードする NameHunter に関して指定されています。通常、このエラーは、参照ファイルの中でバージョン・ヘッダーを検出できなかったことを示しています (例えば taq.ibm または var.ibm)。

システムの処置: GNR モジュールは ibmgmr::Exception 例外をスローします。

ユーザーの処置: 最新バージョンの TAQ ファイル (GNR バージョン 4.1 以降) を入手してください。

GODH020E 無効な TAQ 係数タイプ。

説明: 無効な TAQ 係数タイプが NameHunter に渡されました。TAQ 係数タイプは 1 (相違) または 2 (欠落) でなければなりません。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: 無効な TAQ 係数タイプを訂正して、有効な値を入力してください。

GODH021E TAQ テキストをブランクにすることはできません

説明: TAQ テキスト項目が空です。TAQ テキストの値を提供する必要があります。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: 有効な TAQ テキストを入力した後、名前を NameHunter に渡します。

GODH022E 無効な名前カテゴリーが提供されました

説明: 無効な名前カテゴリーが NameHunter に渡されました。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: 有効な名前カテゴリーを NameHunter に入力してください。有効な名前カテゴリーは、個人および組織です。

GODH065E terms ファイル中のブランク・テキスト

説明: デフォルトの terms ファイル `terms.ibm` の中にブランクの項目が存在します。このエラーは、フィールド・タイプを姓や名として指定することにより引き起こされます。その場合、名前のクレンジング・プロセスの一部として特定の組織用語が削除されることとなります。terms.ibm ファイル内の項目は、組織名専用です。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: デフォルトの terms ファイル `terms.ibm` を使用するには、`loadTerms()` 関数の中で組織名のフィールド・タイプを指定する必要があります。例えば、`loadTerms("terms.ibm",OrgName)` のようにします。

GODP026E Unicode からの変換でエラーが発生しました

説明: NameParser で Unicode からストリングを変換しようとして、エラーが発生しました。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: NameParser で想定されるエンコード形式のストリングが渡されることを確認してください。エンコード方式が未指定の場合は、プラットフォーム・デフォルトに一致するエンコード方式でなければなりません。

GODP028E Unicode ストリングの作成でエラーが発生しました

説明: NameParser は、想定されないエンコード形式のストリングを受け取りました。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: NameParser で想定されるエンコード形式のストリングが渡されることを確認してください。エンコード方式が未指定の場合は、プラットフォーム・デフォルトに一致するエンコード方式でなければなりません。

内部エラー・コード

内部エラーは、何らかの方法でアプリケーションが破壊されたことを示しています。内部エラーの後に続けるには、アプリケーションを再始動する必要があります。通常このタイプのエラーは、クライアント・コードによって IBM InfoSphere Global Name Recognition の内部データ構造が上書きされることによって発生します。

GODH007E メモリーが不足しています

説明: 名前の追加中に十分なメモリーが得られない場合、SearchList クラスによってこのエラーが報告されます。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: 使用可能なメモリーの量を増やすか、メモリーにロードされる名前の数を減らしてください。

GODP024E NameAnalyzer インスタンスの作成に失敗しました

説明: NameAnalyzer ライブラリーを初期化しようとして問題が発生しました。通常、このエラーは、指定された場所に NameAnalyzer.dat ファイルが存在しないことを示しています。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: NameAnalyzer.dat ファイルのパス名が正しいことを確認します。パス名が間違っている場合は、NameAnalyzer.dat ファイルが存在する正しいパス名を入力します。

GODP025E コンバーターを開こうとしてエラーが発生しました

説明: 呼び出し元が、無効なエンコード名を NameParser に渡しました。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: アプリケーション・コードを検査して、NameParser に渡される名前が有効な Internet Assigned Numbers Authority (IANA) 文字セット名であることを確認してください。

GODP027E UTF-8 への変換でエラーが発生しました

説明: NameParser でストリングを UTF-8 形式に変換しようとして、内部エラーが発生しました。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: NameParser で想定されるエンコード形式のストリングが渡されることを確認してください。エンコード方式が未指定の場合は、プラットフォーム・デフォルトに一致するエンコード方式でなければなりません。

GODP029E Transliterator の作成でエラーが発生しました

説明: NameParser ノイズ・フィルター・リストで構文エラーまたはオーバーフロー・エラーが発生しました。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: NameWorks 構成ファイルに含まれるノイズ・フィルターの数を減らしてください。

GODP030E 規則から Transliterator を作成しているときにエラーが発生しました

説明: 通常は、NameParser ノイズ・フィルター・リストのいずれかのフィルターで構文エラーが発生しているか、フィルター・リスト内のフィルター数が多すぎます。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: 該当するノイズ・フィルターの構文エラーを修復します。さらに、NameWorks を介して NameParser を呼び出す場合、NameWorks 構成ファイルに含まれるノイズ・フィルターの数を減らすことができます。

GODT044E ICU エラー

説明: ICU ライブラリーがエラーを戻しました。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラー発生時に何をしようとしていたかを必ず把握しておいてください。

GODV050E 分析が失敗しました

説明: NameAnalyzer.dat ファイルを検索しているときに NameVariantGenerator で内部エラーが発生したことを示しています。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラー発生時に何をしようとしていたかを必ず把握しておいてください。

GODW101E 無効な国/地域別情報コード

説明: 無効な国/地域別情報コードが IBM NameWorks で検出されました。エラー・メッセージの後に国/地域別情報コード番号が示されます。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラー発生時に何をしようとしていたかを必ず把握しておいてください。

GODW102E 無効な国/地域別情報セット・ビットマップ

説明: 無効な国/地域別情報セット・ビットマップが IBM NameWorks で検出されました。エラー・メッセージの後にビットマップ整数が示されます。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

IBM NameWorks のエラー・コード

IBM NameWorks にエラーが発生すると、数値エラー・コードが返されます。エラーが発生した場合、IBM InfoSphere Global Name Management 文書でエラー・コード番号を確認し、エラーのタイプ、その発生場所、およびその修正方法についての情報を取得します。

エラーの原因と重大度の間で区別をするために、エラー・コードが使用されます。C++ および Java では、IBM NameWorks の Scoring クラスおよび Analytics クラスからのエラー情報を報告するために、以下の例外クラスが使用されます。これらのクラスには、ご使用のアプリケーションで使用できるいくつかのメソッドが組み込まれています。

表 19. C++ および Java の例外クラス

C++ 例外	Java 例外
<code>ibmgmr::NwException</code>	<code>java.lang.RuntimeException</code>

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラー発生時に何をしようとしていたかを必ず把握しておいてください。

GODW103E 無効な名前カテゴリー・コード

説明: 無効な名前カテゴリー・コードが IBM NameWorks で検出されました。エラー・メッセージの後に名前カテゴリー・コードが示されます。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラー発生時に何をしようとしていたかを必ず把握しておいてください。

GODW104E 無効な名前カテゴリー・セット・ビットマップ

説明: 無効な名前カテゴリー・セット・ビットマップが IBM NameWorks で検出されました。エラー・メッセージの後にビットマップ整数が示されます。

システムの処置: GNR モジュールは `ibmgmr::Exception` 例外をスローします。

ユーザーの処置: アプリケーションを再始動します。アプリケーションを何度も再始動してもエラーが継続する場合は、IBM 製品サポートに連絡してください。いつどこでエラーが発生したか、エラー発生時に何をしようとしていたかを必ず把握しておいてください。

C++ catch 節の例

```
catch (ibmgnr::NwException & e)
{
    int code = e.code(),
    std::string text = e.text(),
    reportComplexError(type, code, text),
};
```

int code() const

特定のエラー条件に関連付けられたエラー・コードが返されます。

const char *text() const throw()

エラーを伴う可能性がある関連テキスト情報が返されます。

const char *what() const throw()

エラー条件 ID と、それに関連付けられた整数値およびテキスト情報の両方が含まれる文字列が、シングル・スペース文字区切りで返されます。

Java catch 節の例

```
catch ( Throwable exception )
{
    StringWriter stackTrace = new StringWriter();
    PrintWriter printer = new PrintWriter(stackTrace);
    exception.printStackTrace(printer);
    printer.close();
    reportError(exception.getMessage(), stackTrace);
}
```

reportError()

いくつかの異なる例外を報告することを許可します。

IBM NameWorks の C++ エラー・コード

これらのエラー・コードは、IBM NameWorks 内の個々のエラーにかなり固有のもので、1 つのメソッド (例えば、Analytics::analyze()) を呼び出すと、エラーが発生した IBM InfoSphere Global Name Management コンポーネントによっては、いくつかの異なるエラーがスローされる場合があります。

GODW001E 構成ファイルのロード中にエラーが発生しました

説明: 通常、このエラーは、構成ファイルのファイル名が間違っていることを示しています。

システムの処置: IBM NameWorks モジュールは `ibmgnr::NwException` 例外をスローします。

ユーザーの処置: ファイル名が正しいこと、および構成ファイルが正しい場所にあることを確認してください。

GODW002E 音訳オブジェクトを作成できませんでした

説明: 音訳オブジェクトが作成されませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgnr::NwException` 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW003E NameTransliterator ロック

説明: Transliterator オブジェクトを使用するためのロックができませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgnr::NwException` 例外をスローします。

ユーザーの処置: 実行しようとしていた操作を再試行します。問題が解決しない場合は、アプリケーションを再始動してください。

GODW004E NameTransliterator エラー

説明: 音訳エラーが発生しました。

`ibmgnr::NwException` オブジェクトの `what()` スtring を介して NameTransliterator エラー・メッセージが戻されます (例えば GODTnnnE、ただし `nnn` はエラー番号)。

システムの処置: IBM NameWorks モジュールは

ibmgmr::NwException 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW005E NameParser 構築エラー

説明: パーサー・オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW006E NameParser ロック・エラー

説明: パーサー・オブジェクトを使用するためのロックができませんでした。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: 実行しようとしていた操作を再試行します。問題が解決しない場合は、アプリケーションを再始動してください。

GODW007E NameParser エラー

説明: 構文解析エラーが発生しました。

ibmgmr::NwException オブジェクトの **what()** スtring を介して NameParser エラー・メッセージが戻されます (例えば GODPnnnE、ただし *nnn* はエラー番号)。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW008E 無効なカスタム・トークン

説明: カスタム・トークンが無効です。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: 構成ファイル内のカスタム・トークン情報を訂正して、アプリケーションを再始動します。

GODW009E NameVariantGenerator 構築エラー

説明: NameVariantGenerator オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW010E NameVariantGenerator ロック・エラー

説明: NameVariantGenerator オブジェクトを使用するためのロックができませんでした。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: 実行しようとしていた操作を再試行します。問題が解決しない場合は、アプリケーションを再始動してください。

GODW011E NameVariantGenerator エラー

説明: NameVariantGenerator オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW012E Country of Association (COA) 構築エラー

説明: COA オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW013E Country of Association (COA) ロック・エラー

説明: COA オブジェクトを使用するためのロックができませんでした。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

ユーザーの処置: 実行しようとしていた操作を再試行します。問題が解決しない場合は、アプリケーションを再始動してください。

GODW014E Country of Association (COA) エラー

説明: 構文解析エラーが発生しました。

ibmgmr::NwException オブジェクトの **what()** String を介して COA エラー・メッセージが戻されます (例えば GODAnnnE、ただし *nnn* はエラー番号)。

システムの処置: IBM NameWorks モジュールは ibmgmr::NwException 例外をスローします。

GODW015E • GODW024E

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW015E Analytics 実装の構築エラー

説明: メモリー不足のため Analytics オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: より多くのメモリーをプロセスに割り振り、実行しようとしていた操作を再試行します。

GODW016E Scoring 実装の構築エラー

説明: メモリー不足のため Scoring オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: より多くのメモリーをプロセスに割り振り、実行しようとしていた操作を再試行します。

GODW017E NameSifter 構築エラー

説明: NameSifter オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW018E DateCompare 構築エラー

説明: メモリー不足のため DateCompare オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: より多くのメモリーをプロセスに割り振り、実行しようとしていた操作を再試行します。

GODW019E NameHunter 構築エラー

説明: NameHunter オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、遂行しようとした操作を再試行します。

GODW020E NameHunter での名前比較エラー

説明: 構文解析エラーが発生しました。

`ibmgmr::NwException` オブジェクトの `what()` スtring を介して NameHunter エラー・メッセージが戻されます (例えば `GODHnnnE`、ただし `nnn` はエラー番号)。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: エラー・メッセージを調べ、エラーを修復して、比較操作を再試行します。

GODW021E データ・リストの欠落

説明: 提供されたデータ・リスト名は構成ファイルの中ではありません。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: データ・リスト名が正しいことを確認してください。構成ファイルを修復するか、データ・リスト名を訂正して、実行しようとしていた操作を再試行します。

GODW022E 不適切なサーバー・データを含むデータ・リスト

説明: データ・リストの "Server=" 項目が不完全です。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: 構成ファイル内のサーバー・データを訂正して、アプリケーションを再始動します。

GODW023E サーバー・データのないデータ・リスト

説明: データ・リストに "Server=" 項目が含まれていません。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: 構成ファイルを訂正して、アプリケーションを再始動します。

GODW024E 検索方針の欠落

説明: 提供された検索方針名は構成ファイルの中ありません。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: 検索方針の名前が正しいことを確認してください。構成ファイルを修復するか、検索方針名を

訂正して、実行しようとしていた操作を再試行します。

GODW025E Searcher 構築エラー

説明: メモリー不足のため、検索の実行が失敗しました。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: より多くのメモリーをプロセスに割り振り、実行しようとしていた操作を再試行します。

GODW026E 検索エラー

説明: 検索エンジンがエラーを戻しました。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: メッセージ内のエラーは、Distributed Search プロセスがエラー情報を戻したことを示しています。あるいは、通信エラーが発生して、メッセージにエラー・コードが含まれています。エラーを訂正して検索操作を再試行してください。

GODW027E データ・リスト追加エラー

説明: 追加操作が失敗しました。追加サーバーが指定されなかったこと (ブランクのメッセージ)、または検索エンジンからのエラー情報がメッセージに含まれることが原因です。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: メッセージ内のエラーは、Distributed Search プロセスがエラー情報を戻したことを示しています。あるいは、通信エラーが発生して、メッセージにエラー・コードが含まれています。エラーを訂正して追加操作を再試行してください。

GODW028E データ・リスト更新エラー

説明: 通信エラーのために更新操作が失敗しました。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: 通信エラーを訂正して更新操作を再試行してください。

GODW029E データ・リスト削除エラー

説明: 通信エラーのために削除操作が失敗しました。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: 通信エラーを訂正して削除操作を再試行してください。

GODW030E データ・リスト・フェッチ・エラー

説明: 通信エラーのためにフェッチ操作が失敗しました。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: 通信エラーを訂正してフェッチ操作を再試行してください。

GODW031E 無効なパラメーター値

説明: NameVariantGenerator オブジェクトを作成できませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。

ユーザーの処置: 無効なパラメーターを訂正して、実行しようとしていた操作を再試行します。

GODW032E 不適切なデータ・ファイル

説明: 名前データ・ファイルを開けませんでした。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。この例外は、Java API および Web サービスの Java `RunTimeException` として報告されます。

ユーザーの処置: 名前データ・ファイルが存在すること、およびパス名が正しいことを確認してください。問題が解決しない場合は、該当するファイルの新しいコピーを取得します。

GODW033E 不適切なレコード

説明: 無効な名前レコードが名前データ・ファイルの中に存在します。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。この例外は、Java API および Web サービスの Java `RunTimeException` として報告されます。

ユーザーの処置: 無効なレコードを訂正して、実行しようとしていた操作を再試行します。

GODW034E トランザクション ID ロック

説明: このトランザクションで内部エラーが発生しました。

システムの処置: IBM NameWorks モジュールは `ibmgmr::NwException` 例外をスローします。この例外

は、Java API および Web サービスの Java RuntimeException として報告されます。

ユーザーの処置: アプリケーションを再始動して、エラーを解決してください。

GODW035E 不適切なデフォルト比較パラメーター

説明: デフォルトの比較パラメーター (CompParms) 指定変更ファイルでエラーが発生しました。

システムの処置: IBM NameWorks モジュールは ibmgnr::NwException 例外をスローします。この例外は、Java API および Web サービスの Java RuntimeException として報告されます。

ユーザーの処置:

GODW036E 正規化ロックの失敗

説明: 内部正規化エラーが発生しました。

システムの処置: IBM NameWorks モジュールは ibmgnr::NwException 例外をスローします。この例外は、Java API および Web サービスの Java RuntimeException として報告されます。

ユーザーの処置: アプリケーションを再始動して、エラーを解決してください。

GODW037E 無効な CompParm 名

説明: 無効な比較パラメーター (CompParm) オーバライド名が存在します。

システムの処置: IBM NameWorks モジュールは ibmgnr::NwException 例外をスローします。この例外は、Java API および Web サービスの Java RuntimeException として報告されます。

Distributed Search のエラー・コード

Distributed Search のエラーは、XML 応答の <ERROR> タグの中でクライアントに戻されます。各エラーは重大度の分類、エラー・コード、およびメッセージを報告します。

gnrds-001 メッセージの始まりを検出できませんでした。破棄します。

説明: Distributed Search で、メッセージの始まりを識別するタグが検出されませんでした。このタグは <NHServerMessage> または <NH_SERVER_MESSAGE> のいずれかです。このエラーは、プログラミング上のエラーまたは通信エラーを示しています。

ユーザーの処置: 失敗したメッセージの再送信を試行で

ユーザーの処置: 無効な CompParm 名を修復してください。エラー・メッセージと共に示されるテキスト・ストリングは、エラーが発生したファイル名をリストしています。

GODW038E 無効な CompParm 値

説明: 無効な比較パラメーター (CompParm) オーバライド値が存在します。

システムの処置: IBM NameWorks モジュールは ibmgnr::NwException 例外をスローします。この例外は、Java API および Web サービスの Java RuntimeException として報告されます。

ユーザーの処置: 無効な CompParm 値を修復してください。エラー・メッセージと共に示されるテキスト・ストリングは、エラーが発生したファイル名をリストしています。

GODW039E 無効なデータ・リスト・タイプ

説明: 無効なデータ・リスト・タイプが IBM NameWorks 構成ファイルの中に存在します。

システムの処置: IBM NameWorks モジュールは ibmgnr::NwException 例外をスローします。この例外は、Java API および Web サービスの Java RuntimeException として報告されます。

ユーザーの処置: 無効なデータ・リスト・タイプ・パラメーターを訂正してください。Type= パラメーターとして有効な項目の値は、次のとおりです。

- 0 = 組み込み検索
- 1 = 全検索
- 2 = 固有の名前検索

きます。ただし、システム再始動が唯一の解決策である可能性もあります。

gnrds-002 メッセージの終わりを検出できませんでした。破棄します。

説明: Distributed Search で、メッセージの終わりを識別するタグが検出されませんでした。このタグは "/NHServerMessage>" または "/NH_SERVER_MESSAGE>" のいずれかです。このエラーは、プログラミング上のエラーまたは通信エラーを示しています。

ユーザーの処置: 失敗したメッセージの再送信を試行できます。ただし、システム再始動が唯一の解決策である可能性もあります。

gnrds-003 メッセージ・タイプを判別できませんでした

説明: request_type フィールドに無効値が入っているか、Distributed Search で応答メッセージが認識されません。このエラーは、プログラミング上のエラーまたは通信エラーを示しています。

ユーザーの処置: 失敗したメッセージの再送信を試行できます。ただし、システム再始動が唯一の解決策である可能性もあります。

gnrds-004 レコード・ヘッダー RECORD_HEADER を検出できませんでした

説明: Distributed Search は、必要なレコード・ヘッダー (例えば SEARCH_NAME) を検出できませんでした。この欠落しているヘッダーが、上記の "RECORD_HEADER" の代わりに表示されます。このエラーは、プログラミング上のエラーを示しています。

gnrds-005 タグ TAG を検出できませんでした

説明: Distributed Search は、必要なタグ (例えば request_type) を検出できませんでした。上記の "TAG" の代わりに、欠落しているタグが表示されます。このエラーは、プログラミング上のエラーを示しています。

gnrds-006 無効なメッセージ・タイプ X

説明: "request_type" フィールドの値が無効です。上記の "X" の代わりに、間違っている値が表示されます。

gnrds-007 数値でなければなりません:
TAG=VALUE

説明: 数値フィールドに非数値が指定されました。問題となっている TAG と VALUE がメッセージに表示されます。

gnrds-008 0.0 と 1.0 の間でなければなりません:
TAG=VALUE

説明: スケール値 (例えばしきい値) が、必要な範囲の外にあります。問題となっている TAG と VALUE がメッセージに表示されます。

gnrds-009 無効な国/地域別情報コード:
TAG=VALUE

説明: 国/地域別情報コードが、サポートされる範囲外にあります。有効な値については、国/地域別情報コードの表を参照してください。問題となっている TAG と VALUE がメッセージに表示されます。

gnrds-010 無効なブール値: TAG=VALUE

説明: ブール値に有効な値が含まれていません。Distributed Search では true として T、TRUE、Y、YES、ON、1、および false として F、FALSE、N、NO、OFF、0 が受け入れられます。それらの値で大/小文字は区別されません。問題となっている TAG と VALUE がメッセージに表示されます。

gnrds-011 0 より大きい数値でなければなりません:
TAG=VALUE

説明: 正数の必要なフィールドに、1 より小さい数値が指定されました。問題となっている TAG と VALUE がメッセージに表示されます。

gnrds-012 アンカー・タイプは 0、1、または 2 でなければなりません: TAG=VALUE

説明: ANCHOR_TYPE の値が無効です。

gnrds-013 スコア・モードは 0、1、または 2 でなければなりません: TAG=VALUE

説明: SCORE_MODE の値が無効です。

gnrds-014 追加または更新をサポートする Searcher が構成されていません

説明: 追加または更新の要求が Distributed Search で受信されましたが、追加をサポートする Searcher がありません。

ユーザーの処置: 追加をサポートするには、1 つの Searcher の構成設定を "doAdds=true" にする必要があります。

gnrds-015 欠落 GN、および欠落 SN の照会は許可されません

説明: ブランクの GN およびブランクの SN を含む検索要求を受け取りましたが、Distributed Search はこのような種類の照会を拒否するよう構成されています。構成ファイルには "allowFnuLnu=false" という項目があります。

gnrds-016 欠落 GN、および SN イニシャルの照会は許可されません

説明: ブランクの GN、および SN の単一イニシャルを含む検索要求を受け取りましたが、Distributed Search はこのような種類の照会を拒否するよう構成されています。構成ファイルには"allowFnuInit=false"という項目があります。

gnrds-017 GN イニシャル、欠落 SN の照会は許可されません

説明: GN の単一イニシャル、およびブランクの SN を含む検索要求を受け取りましたが、Distributed Search はこのような種類の照会を拒否するよう構成されています。構成ファイルには"allowInitLnu=false"という項目があります。

gnrds-018 GN イニシャル、SN イニシャルの照会は許可されません

説明: GN の単一イニシャル、および SN の単一イニシャルを含む検索要求を受け取りましたが、Distributed Search はこのような種類の照会を拒否するよう構成されています。構成ファイルには"allowInitInit=false"という項目があります。

gnrds-020 1 つ以上の Searcher が応答していません

説明: 1 つ以上の Searcher が応答していません。完全な応答を作成できません。

ユーザーの処置: ほとんどの場合、Distributed Search を再始動する必要があります。

gnrds-021 名前 ID をブランクにすることはできません

説明: 追加および更新の要求では ID を指定する必要があります。このメッセージが戻される場合、ID がブランクです。

gnrds-022 更新する名前 ID をブランクにすることはできません

説明: 追加および更新の要求では、更新される ID を指定する必要があります。このメッセージが戻される場合、ID_TO_UPDATE がブランクです。

gnrds-023 Searcher 応答がメッセージ・バッファーに対して大きすぎます

説明: この原因は、ほとんどの場合、Searcher から

commgr に送られる検索結果メッセージの応答が多すぎるためです。

ユーザーの処置: 多すぎる結果が生成されるような照会を避けてください。ただし、ds.config で msgBufferSize を設定することによってメッセージ・バッファー・サイズを増やすことができます。デフォルトは 1Mb です。

gnrds-024 メッセージを構文解析できませんでした

説明: NameParser は、受け取ったメッセージを構文解析できませんでした。

gnrds-025 音訳できませんでした。無効な UTF8

説明: NameTransliterater は、照会または追加メッセージの SN または GN フィールドの中に無効な UTF8 を検出しました。また、入力ファイルの中に無効な UTF8 が検出される場合、開始時および前処理中にこのメッセージがエラー・ログに書き込まれることもあります。

gnrds-026 スコア・タイプは 0 または 1 でなければなりません

説明: compparms.config ファイル内でスコア・タイプに 0 または 1 以外の値が指定されました。このエラーは非常に珍しく、通常の操作条件の下では発生しません。

注意:

score_type 値を変更すると、Distributed Search で使われるスコアリング・アルゴリズムが変更されます。この値を変更しないでください。

gnrds-027 名前のカテゴリは P、O、または A でなければなりません

説明: 個人 (P)、組織 (O)、またはすべて (A) 以外の名前カテゴリが照会名に指定されました。

ユーザーの処置: 無効値を除去して、有効な名前カテゴリを挿入してください。

gnrds-028 名前を空にすることはできません

説明: 空の照会名が Distributed Search に渡されました。

ユーザーの処置: 有効な名前を照会名に指定して、検索操作を再試行してください。

gnrds-029 検索オプションは 1、2、または 3 でなければなりません

説明: 検索方針の SearchOpt= パラメーターで 1、2、または 3 以外の値が指定されました。

ユーザーの処置: 検索方針の SearchOpt= パラメーターに有効な値を指定してください。

- 1 = 個人名リストのみの検索
- 2 = 組織名リストだけを検索
- 3 = 個人名と組織名の両方のリストの検索

gnrds-030 負の数値にすることはできません

説明: Distributed Search では負の値がサポートされないため、すべての数値に関してこのエラーが発生する可能性があります。

ユーザーの処置: ゼロ以上の値を指定することにより、無効値を修正してください。

gnrnh-001 構文解析後、GN および SN がブランクです

説明: このメッセージは、姓名がブランクであるために、レコードの追加試行が失敗したことを示しています。ユーザーがこれらをブランクとして入力したか、音訳によってブランクに変換された可能性があります。

Enterprise Name Search のエラー・コード

Enterprise Name Search のエラーは、XML 応答の <ERROR> タグ、JSON 出力のエラー・リスト、または GUI に表示されるダイアログでクライアントに返され、通常は ENS サーバー・ログにも記録されます。Web サービスの出力に表示されるエラー通知には、エラー・コードが含まれ、場合によってはメッセージも含まれます。

ENS コンソールのエラー・コード

ENS コンソール GUI のエラー・ダイアログ、場合によってはサーバー・ログ・ファイルにも現れる可能性があるメッセージおよび関連コードを次に示します。

CDHNC1001E

アクションが指定されませんでした。

CDHNC1002E

セルの開始中または停止中に例外が発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1301E

内部エラーが発生し、それによりデータの状態が不整合になりました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1302E

内部エラーが発生しました。複数のセルが定義されています。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1303E

内部エラーが発生しました: セルが定義されていません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1304E

内部サーバー・エラーが発生しました: 製品はセルを見つけることができません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1305E

redundancyType の値が正しくありません。この値は、NONE、MIRRORED、または OVERLAPPING のいずれか 1 つでなければなりません。

CDHNC1306E

ミラー・カウントを指定できるのは冗長タイプが MIRRORED のときだけで、そのカウントは 1 より大きくなければなりません。

CDHNC1307E

内部データベース・エラーが発生しました: 更新に失敗しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1308E

内部エラーが発生しました: パーティションが作成されませんでした。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1309E

内部エラーが発生し、それによりデータの状態が不整合になりました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1310E

アクティブ状態のセルは構成できません。セルの状態を非アクティブに変更してから、セルを構成してください。

CDHNC1311E

内部エラーが発生し、それによりデータの状態が不整合になりました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1312E

内部エラーが発生し、それによりデータの状態が不整合になりました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1313E

内部エラーが発生し、それによりデータの状態が不整合になりました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1314E

次のフォーマットをサーバー ID に使用します: `ipAddress_ProfileName`。

CDHNC1315E

フィールド (*fieldname*) は空でない配列でなければなりません。

CDHNC1316E

フィールド (*fieldname*) はタイプ (*type*) の非ヌル値でなければなりません。

CDHNC1317E

内部エラーが発生しました: セル状態の収集中に不明な例外が発生しました。データベースが稼働していない可能性があります。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1318E

ここにリストされていないサーバー (*serverID*) がセルに含まれるようになりました。この表示を更新して、セル構成操作を再試行してください。

CDHNC1319E

内部エラーが発生しました: ここにリストされたサーバー (*serverID*) がセルの一部ではなくなりました。この表示を更新して、セル構成操作を試行してください。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1320E

内部エラーが発生しました: サーバー (*serverID*) の (*autoStartFlag*) が欠落しています。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1321E

内部エラーが発生しました: サーバー (*serverID*) が複数回リストされました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1322E

内部エラーが発生しました: 少なくとも 1 つのサーチャーを選択する必要があります。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1323E

内部エラーが発生しました: *mirrorCount* の入力データが無効です。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1324E

内部エラーが発生しました: *redundancyType* の入力データが無効です。シ

システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1401E

アクションが指定されませんでした。

CDHNC1402E

サーバーの管理中に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1501E

アクションが指定されませんでした。

CDHNC1502E

ディスパッチャーの開始中または停止中に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1601E

アクションが指定されませんでした。

CDHNC1602E

サーチャーの管理中に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1604E

アクティブなサーチャーは削除できません。

CDHNC1801E

ダッシュボードの管理中に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1802E

serverID を整数として解析できませんでした。

CDHNC1803E

serverID を取得できませんでした。

CDHNC1804E

パラメーターが定義されていません。

CDHNC1805E

内部エラーが発生しました: 複数のサーバーが同じ ID に割り当てられています。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1806E

内部エラーが発生しました: 製品がデータベース内でこの ID のサーバーを見つけられません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1807E

内部エラーが発生しました: データベースを正常に更新できませんでした。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1809E

アクションが指定されませんでした。

CDHNC1901E

内部エラーが発生し、それによりデータの状態が不整合になりました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1902E

実行するアクションはありません。

CDHNC1903E

内部エラーが発生しました: 製品は複数のアクティブ・セルを見つけました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1904E

内部エラーが発生しました: 製品は構成済みのセルを見つけることができません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1906E

内部エラーが発生しました: 製品は変更するセルを見つけることができません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1907E

内部エラーが発生しました: 製品はデータベースを正常に更新できませんでした。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC1908E

コンソールの転送中に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2101E

URL から serverID を取得できませんでした。

CDHNC2102E

serverID を解析できませんでした。

CDHNC2103E

内部エラーが発生しました: 製品はデータベース内に指定の ID を持つ複数のサーバーを見つけました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2104E

製品はデータベース内にサーバー ID を見つけることができません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2105E

パラメーター (*paramName*) の値は true または false でなければなりません。

CDHNC2106E

コンポーネントは既に開始または停止の処理中です。

CDHNC2107E

サーバーがアクティブではないので、コンポーネントを開始または停止することができません。

CDHNC2108E

内部エラーが発生しました: データベースの更新に失敗しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2109E

内部エラーが発生し、それによりデータの状態が不整合になりました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2110E

サーバー状態表をロックしようとして例外が生じました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2111E

データベースへのアクセス中に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2112E

データベースへの書き込み中に内部エラーが発生しました。システム管理

者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2113E

サーチャードが自動開始するように設定する前に、パーティションを定義する必要があります。

CDHNC2114E

サーチャードを開始する前に、パーティションを定義する必要があります。

CDHNC2115E

現在アクティブな製品サーバーはありません。

CDHNC2116E

内部エラーが発生しました: 製品はセルの状況を取得できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2117E

URL を判別できません。

CDHNC2118E

予期しないパラメーター名がサービスに渡されました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2119E

mirroredFrom 値が serverID と同じになっています。サーバーをミラーリングできません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2120E

セルが非アクティブなので、開始や停止を実行できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2121E

newServer 値が serverID と同じになっています。コンソールを転送できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2122E

予期しない空白のパラメーター名がサービスに渡されました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2123E

内部エラーが発生しました。システム管理者にお問い合わせください。製

品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNC2124E

セル内にもう 1 つだけあるサーバーには既に対応するコンポーネントが構成されているので、このサーバーをミラーリングできません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

ENS 検索のエラー・コード

ENS 検索 GUI のエラー・ダイアログ、Web サービスのエラー応答、場合によってはサーバー・ログ・ファイルにも現れる可能性があるメッセージおよび関連コードを次に示します。

CDHNS3000E

予期されるタイムアウト・インターバル内に、サーバーからの応答がありませんでした。

CDHNS3001E

検索権限がある名前リストが見つかりませんでした。

CDHNS3002E

戦略が見つかりませんでした。

CDHNS3003E

国/地域別情報は見つかりませんでした。

CDHNS3010E

内部エラーが発生しました: ディスパッチャーを再初期化できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3011E

nameType パラメーターはヌルまたは空にはできません。

CDHNS3012E

nameText パラメーターはヌルまたは空にはできません。

CDHNS3013E

nameType の値は PERSON または ORGANIZATION でなければなりません。

CDHNS3014E

サーチャーとの通信中に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3015E

内部エラーが発生しました: いくつかの必要なサーチャー・コンポーネントが使用できません。システム管理者にサーバー状況の検査を依頼してください。

CDHNS3016E

内部エラーが発生しました: 製品はサーチャーの応答を読み取ることができません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3017E

内部エラーが発生しました: 製品はサーチャー要求を構成できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3018E

内部エラーが発生しました: 製品はアクティブなサーチャーのリストを取得できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3019E

内部エラーが発生しました: 製品はこのサーバーでディスパッチャー・インスタンスを見つけることができません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3020E

内部エラーが発生しました: ディスパッチャーがアクティブではありません。システム管理者にサーバー状況の検査を依頼してください。

CDHNS3021E

nameListCode パラメーターは NULL や空にはできません。

CDHNS3022E

externalID パラメーターは NULL や空にはできません。

CDHNS3023E

使用するユーザー ID には、この名前リストのために設定されたサーチャー権限がありません。システム管理者に連絡してください。

CDHNS3024E

nameListCodes が固有ではないか、欠落しているために、名前の詳細を取得できません。

CDHNS3025E

無効なソース名が指定されました。その名前は存在しません。

CDHNS3026E

内部エラーが発生しました: ソース名は固有でなければなりません。製品は複数のソース名を見つけました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3027E

名前の基準が欠落しているために、名前を追加できません。

CDHNS3028E

要求パラメーターに指定された `externalID` がメッセージ本体に指定された `externalID` と一致しません。

CDHNS3029E

要求パラメーターに指定された `nameListCode` がメッセージ本体に指定された `nameListCode` と一致しません。

CDHNS3030E

使用するユーザー ID には、この名前リストのために設定された管理者権限がありません。システム管理者に連絡してください。

CDHNS3031E

`nameCategory` パラメーターは NULL や空であってはなりません。

CDHNS3032E

`nameCategory` パラメーターが有効な名前カテゴリーではありません。

CDHNS3033E

`nameListCodes` が固有ではないか、欠落しているために、名前を削除できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3034E

指定したパラメーターは、個人名を追加するときには許可されていません:
`paramNames`。

CDHNS3035E

`includeAlternateParses` パラメーターの値は `true` または `false` でなければなりません。

CDHNS3036E

FULL 名前タイプを追加するとき、パラメーター `paramNames` は無効です。

CDHNS3037E

PARSED 名前タイプを追加するとき、`givenName` パラメーターと `surname` パラメーターの両方を NULL や空にはできません。

CDHNS3038E

`nameListCodes` が複数あるか、または欠落しているために、名前を追加できません。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (`errorTimestamp`) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3039E

指定したパラメーターは、組織名を追加するときには許可されていません:
`paramNames`。

CDHNS3040E

組織名を追加するとき、`nameText` パラメーターは NULL や空にはできません。

CDHNS3041E

基準が欠落しているために、外部参照を取得できませんでした。

CDHNS3042E

サーチャーのポート番号を取得しようとして、内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3043E

サーチャーへの SSL 接続を確立しようとして、内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3044E

内部エラーが発生しました: サーチャーの SSL セッションが無効です。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3045E

サーチャーへの SSL 接続を構成しようとして、内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3046E

内部エラーが発生しました: 使用可能なすべてのサーチャー・コンポーネントが指定されたタイムアウト間隔を超過しています。管理者にサーバー状況の検査を依頼してください。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3047E

内部エラーが発生しました: 必要なサーチャー・コンポーネントが使用できません。管理者にサーバー状況の検査を依頼してください。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3048E

名前音訳の際に内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3049E

存在しないか、現在の権限では検索できない、1 つ以上の名前リストを指定しました。

CDHNS3050E

検索権限がある名前リストがありません。

CDHNS3051E

内部エラーが発生しました: サーチャーがエラー・メッセージを返しました。システム管理者にお問い合わせください。製品は、概算日時のタイム

スタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3052E

maxResults は 0 (ゼロ) より大きい整数でなければなりません。あるいは、-1 を指定してデフォルトの maxResults を構成に使用するか、0 (ゼロ) を指定して最大限度を構成に使用します。

CDHNS3053E

minScore は 0 (ゼロ) 以上の整数でなければなりません。あるいは、-1 を指定してデフォルトの minScore を構成に使用します。

CDHNS3061E

名前を解析できません。おそらく無効な文字が原因です。

CDHNS3062E

検索名を解析できません。おそらく無効な文字が原因です。

CDHNS3063E

内部エラーが発生しました: 製品が既存の検索名を確認できません。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3072E

searchName に渡す名前リストのリストの項目数が、構成されている最大項目数を超えてはいけません。

CDHNS3092E

戦略コードを取得しようとして、不明エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3093E

国/地域別情報を取得しようとして、不明エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3094E

名前検索を実行しようとして、不明エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3095E

名前リストを取得しようとして、不明エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3096E

名前の詳細を取得しようとして、不明エラーが発生しました。システム管

理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3097E

名前を追加しようとして、不明エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3098E

名前を削除しようとして、不明エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS3099E

外部参照を取得しようとして、不明エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS4001E

サーバーでエラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS4002E

paramName の値を指定する必要があります。

CDHNS4003E

パラメーター *paramName* に無効な値が指定されました。

CDHNS4004E

minScore は 0 から 100 の間の整数である必要があります。あるいは、-1 を指定して検索戦略のデフォルト値を使用します。

CDHNS4005E

maxResults はゼロより大きい整数でなければなりません。あるいは、-1 を指定して検索戦略のデフォルト値を使用します。

CDHNS4006E

サーバーでデータベース・エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS4007E

サーバーで製品エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS4008E

予期しないパラメーターが検索サービスに渡されました。システム管理者

にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

CDHNS4009E

指定された名前には、音訳できない文字が含まれています。この原因として、サポートされない文字セットが使用されていることや、同じ名前に複数の異なる文字セットが混在していることなどが考えられます。

CDHNS4010E

内部エラーが発生しました。システム管理者にお問い合わせください。製品は、概算日時のタイムスタンプ (errorTimestamp) とともに、アプリケーション・サーバーのログ・ファイルにメッセージを記録しました。

知識ベースの検索

多くの場合、IBM 知識ベースを検索することによって、問題の解決策を見つけることができます。利用可能なリソース、サポート・ツール、検索方式を使用することにより、結果を最適化できます。

このタスクについて

IBM InfoSphere Global Name Management のインフォメーション・センターを検索することによって役立つ情報を見つけることができますが、疑問点を解消したり問題を解決したりするために、場合によってはインフォメーション・センター以外でも調べる必要が生じます。

手順

必要な情報を知識ベースで検索するには、以下のアプローチの 1 つ以上を使用します。

- IBM InfoSphere Global Name Management 用の IBM Support Portal を使用することにより、必要なコンテンツを見つける。

IBM Support Portal は、あらゆる IBM システム、ソフトウェア、およびサービスを対象としたすべての技術サポート・ツールと技術情報の集中統合ビューです。この IBM Support Portal という 1 つの場所から、IBM エレクトロニック支援ポータルフォリオにアクセスできます。問題の予防とより迅速な問題解決のために必要な情報とリソースに重点を置くようにページを調整することができます。IBMSupport Portal をよく理解するには、このツールに関するデモ・ビデオ (https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos) をご覧ください。これらのビデオでは、IBM Support Portal を紹介し、トラブルシューティング用などのリソースを探索するほか、ポートレットを移動、追加、削除してページを調整する方法を示します。

- 次のいずれかの追加技術リソースを利用して IBM InfoSphere Global Name Management に関するコンテンツを検索する。
 - IBM InfoSphere 技術情報
 - IBM InfoSphere APAR (問題報告書)

- IBM マストヘッド検索を使用することにより、コンテンツを検索する。 どの `ibm.com`[®] ページでも、上部の検索フィールドに検索ストリングを入力することによって、IBM マストヘッド検索を使用できます。
- Google や Yahoo、Bing などの外部検索エンジンを使用することにより、コンテンツを検索する。 外部検索エンジンを使用すると、`ibm.com` ドメイン以外の情報が結果に含まれる可能性が高くなります。ただし、`ibm.com` がないニュースグループやフォーラム、ブログで、IBM 製品に関する有益な問題解決情報が見つかることもあります。

ヒント: IBM 製品に関する情報を検索する場合は、「IBM」と製品の名称を検索に含めてください。

ログ・ファイル

該当する条件が特定のシステム・コンポーネントに発生したときに (例えばコンポーネントのインストール時や開始時、または処理中にエラーが発生した場合)、ログ・ファイルに情報が書き込まれます。

IBM NameWorks 構成ファイルなどの構成ファイルを使用して、ログ・ファイルの名前と場所を定義できます。さらに、トレースの有効化を含む、ロギング・レベルの設定も可能です。すべてのトレース情報がログに含められます。

IBM NameWorks は `java.util.logging` パッケージに含まれる標準メカニズムを使用します。このため、この Java パッケージに含まれているクラスを使用すると、IBM NameWorks 機能によって行われるロギングとトレースのあらゆる面を有効化し、制御することができます。デフォルトでは、ロギングが無効になっています。

IBM NameWorks が使用するロガーの名前は `com.ibm.gnr.NameWorks` です。ロギング制御にアクセスするには、例えば以下のようなコードを使用します。

```
java.util.logging.Logger gnrLogger =
java.util.logging.Logger.getLogger( "com.ibm.gnr.NameWorks" );
```

次に、標準のロガー・メソッドを使ってロギング・レベルとハンドラーを設定します。

トレース

トレースは、コンポーネント処理またはトランザクション処理のレコードです。トレースで収集された情報は、問題およびパフォーマンスを評価するために使用できます。

IBM NameWorks では、トレース・ログ・ファイル (`trace.log`) は、デフォルトの製品インストール・ディレクトリー内の `/ewas/profiles/NameWorksProfile/logs/NameWorksServer` ディレクトリーに置かれます。

IBM NameWorks は、WebSphere Application Server で提供されるほとんどのログ・レベル設定 (例えば「重大 (Severe)」、「詳細 - 中 (Fine)」、「詳細 - 高 (Finer)」、「詳細 - 最高 (Finest)」など) をサポートします。IBM NameWorks のトレース設定を変更するには、デフォルト・インストール・ディレクトリーの `/ewas/profiles/NameWorksProfile/config/cells/DefaultNode/nodes/DefaultNode/`

servers/NameWorksServer ディレクトリーにある server.xml ファイルを編集し、**startupTraceSpecification** パラメーターを変更して適切なログ・レベル設定を含めます。例えば、トレース・ログ設定を「詳細 - 中 (fine)」に設定するには、`startupTraceSpecification="*=info:com.ibm.gnr.NameWorks=fine"` を指定します。

IBM サポートへのお問い合わせ

IBM サポートは、製品の障害に関する支援を行うと同時に、FAQ に回答し、再ディスカバリーを実施します。

始める前に

技術情報などのセルフ・ヘルプの他の選択肢で答や解決策を探してみたいうえで、IBM サポートにご連絡ください。IBM サポートに連絡を取るには、お客様の会社の IBM 保守契約が現在有効であり、問題を IBM に処理依頼する権限がお客様になければなりません。利用可能なサポートのタイプについては、「*Software Support Handbook*」の Support portfolio のトピックを参照してください。

このハンドブックの PDF コピーをご希望の場合は、英語によるハンドブックをダウンロードできます。別の言語 (日本語またはフランス語) によるハンドブックのコピーをご希望の場合は、「*Software Support Handbook*」のウェルカム・ページの「**Other languages**」セクションをご覧ください。

手順

IBM サポートに連絡を取って問題を伝えるには、以下のステップを実行します。

1. 問題を定義します。問題と症状をできるだけ具体的に記述します。詳しくは、「*Software Support Handbook*」の Getting IBM support のトピックを参照してください。
2. 問題の重大度レベルを判断します。必要に応じて、「*Software Support Handbook*」の Getting IBM support のトピックで、重大度レベルの参考定義を参照してください。
3. 以下のデータを含めて、背景情報を収集します。
 - 製品のどのバージョンおよびリリース・レベルをご使用ですか。
 - 問題が発生したときにどのレベルのソフトウェアを実行していましたか。データベースおよびオペレーティング・システムのバージョン番号と Service Pack 番号のほかに、関連製品もすべて含めてください。
 - この問題は以前にも起こったことがありますか、あるいは切り分けられた問題ですか。
 - 問題を再現できますか。そうであれば、問題を再現するためにはどのようなステップが必要ですか。
 - お客様はこの問題の主な連絡先ですか、また、どのような連絡方法をご希望ですか (E メールまたは電話)。お客様に連絡の取れる電話番号、ページ、または E メール・アドレスが複数ありますか。お客様の対応可能状況。この問題に IBM ソフトウェア・サポートと一緒にいつ取り組みますか。
 - IBM ソフトウェア・サポートが話をできる見識のある代替連絡先はありますか。

4. 生成された入手可能メッセージまたは診断データ (ログ・ファイルなど) をすべて収集します。サポートに連絡するときに、受け取ったメッセージのメッセージ番号が分かっていると助けになります。
5. 以下のいずれかの方法で IBM サポートに問題を処理依頼します。
 - IBM サポート・ポータルによるオンライン: 「Service Request」ページの「Service Request」ポータルから、お客様のすべてのサービス要求をオープン、更新、および表示できます。
 - 電話: お客様の国での連絡先電話番号については、Directory of worldwide contacts Web ページを参照してください。

タスクの結果

お客様から処理依頼された問題がソフトウェア障害あるいは文書の欠落または不正確に関するものであった場合、IBM サポートはプログラム診断依頼書 (APAR) を作成します。APAR には、問題が詳細に記述されます。IBM サポートは、APAR が解決してフィックスが配信されるまで、お客様が実装できる回避策を可能な限り提供します。IBM は、同じ問題が起こっている他のユーザーが同じように解決できるように、解決した APAR を IBM サポート Web サイトで日々発行しています。

サポート・アップデートへのサブスクライブ

ご使用の IBM 製品に関する重要な情報が通知されるようにしておくには、アップデートにサブスクライブします。

このタスクについて

アップデートを受信するようサブスクライブすることにより、特定のサポート・ツールおよびリソースについての重要な技術情報とアップデートを受信することができます。以下の 2 つのアプローチのいずれかを使用することにより、アップデートにサブスクライブできます。

RSS フィードおよびソーシャル・メディア・サブスクリプション

始めるにあたってのステップや RSS 対応 IBM Web ページのリストを含め、RSS に関する一般情報については、IBM Software Support RSS feeds サイトにアクセスしてください。

My Notifications

どの IBM 製品についても、My Notifications を使用して、サポート・アップデートにサブスクライブできます。E メール告知を毎日受信するか毎週受信するかを指定できます。受信する情報のタイプ (発表、ヒント、製品フラッシュ (アラートとも言う)、ダウンロード、ドライバーなど) を指定できます。My Notifications では、通知を希望する製品と最もニーズに適した配信方式をカスタマイズおよびカテゴリー化できます。

手順

サポート・アップデートにサブスクライブするには、以下のようになります。

1. IBM InfoSphere Global Name Management の通知にサブスクライブするには、まず IBM InfoSphere Global Name Management Support Portal にアクセスして、「通知」ポートレットの「この製品のサブスクリプションの作成または更新」をクリックします。
 - a. MySupport に登録済みの場合は、サインインして次のステップにスキップしてください。登録済みでない場合は、「こちらからご登録ください」をクリックします。お客様の E メール・アドレスを IBM ID として使用して登録フォームを完成させ、「送信する」をクリックします。
 - b. 「通知方式 (**Notify me by**)」で、アップデートを受信するための方式 (E メールや RSS フィードなど) を 1 つ以上選択します。
 - c. 特定のフォルダーへの通知の配信を希望する場合は、「オプション (**Options**)」でフォルダーを選択するかフォルダーの名前を入力します。
 - d. 「文書タイプ (**Document types**)」リストで、受信する通知の対象とする文書のタイプを、希望するだけリストから選択します。
 - e. 「送信する」をクリックします。
2. 他の IBM 製品の My Notifications にサブスクライブするには、まず IBM Support Portal にアクセスし、「通知」ポートレット内の「**My Notifications**」をクリックします。
 - a. MySupport に登録済みの場合は、サインインして次のステップにスキップしてください。登録済みでない場合は、「こちらからご登録ください」をクリックします。お客様の E メール・アドレスを IBM ID として使用して登録フォームを完成させ、「送信する」をクリックします。
 - b. 「プロフィールの編集 (**Edit profile**)」をクリックします。
 - c. 「製品の追加 (**Add products**)」をクリックし、製品カテゴリーを選択します (例えば「ソフトウェア」)。 2 番目のリストが表示されます。
 - d. 2 番目のリストで、製品セグメントを選択します (例えば「データおよび情報管理 (**Data & Information Management**)」)。 3 番目のリストが表示されます。
 - e. 3 番目のリストで、製品サブセグメントを選択します (例えば「データベース」)。 該当する製品のリストが表示されます。
 - f. 受信するアップデートの対象とする製品を選択します。
 - g. 「製品の追加 (**Add products**)」をクリックします。
 - h. 対象とする製品をすべて選択した後、「プロフィールの編集 (**Edit profile**)」タブにある「E メールにサブスクライブ (**Subscribe to email**)」をクリックします。
 - i. 「これらの文書を毎週 E メールで送信してください (**Please send these documents by weekly email**)」を選択します。
 - j. 必要に応じて E メール・アドレスを更新します。
 - k. 「文書リスト (**Documents list**)」で、製品カテゴリーを選択します (例えば「ソフトウェア」)。
 - l. 「文書リスト (**Documents list**)」で、製品カテゴリーを選択します (例えば「ソフトウェア」)。
 - m. 「更新 (**Update**)」をクリックします。

タスクの結果

RSS フィードおよび My Notifications の設定を変更するまでは、要求済みのアップデートの通知を受信します。必要なときに設定を変更できます (例えば、ある製品を使用しなくなり、別の製品を使い始めた場合)。

関連情報

-  [IBM Software Support RSS feeds](#)
-  [Subscribe to My Notifications support content updates](#)
-  [IBM 技術サポートの My notifications](#)
-  [My notifications for IBM technical support overview](#)

付録. 用語集

この用語集には、IBM InfoSphere Global Name Management の用語と定義が含まれています。

この用語集では、以下の相互参照を使用します。

- 「を参照」は、ある用語から推奨同義語へ、または頭字語や省略語から完全形式の定義への参照です。
- 「も参照」は、関連する用語や対比する用語への参照です。

他の IBM 製品の用語集を表示するには、www.ibm.com/software/globalization/terminology を参照してください (新しいウィンドウで開きます)。

A

アクセント記号 (accent mark)

音節の抑揚を示すために使用される発音区別符号。「発音区別符号 (diacritic)」も参照。

接辞 (affix)

名前の従属要素の 1 つで、意味を変える作用を持つ。名前の最初、真ん中、または末尾に追加され、それぞれ、接頭部、接中辞、接尾部と呼ばれる。接辞は、名前に直接付加したり (「Macintosh」の「Mac」など)、句読点で名前語幹から分離したり (「O'Connell」の「O」など)、空白文字で名前語幹から分離したり (「Abd Allah」の「Abd」など) できる。接辞は、ファミリー・ネームで使用されるのが最も一般的で、民族的出身を識別できる場合がある。「名前語幹 (name stem)」も参照。

代替解析名 (alternate parse name)

名前の変形の候補。名前の分析とスコアリングを向上させるために使用される。「名前の変形 (name variant)」も参照。

C

セル (cell)

同じデプロイメント・マネージャーに統合されて、高可用性コア・グループを含めることができる、管理対象プロセスのグループ。

等位結合名 (conjoined name)

複数の名前セグメントをグループ化したもので、その構成要素は、2 つ以上のギブンネーム (名)、2 つ以上の称号、称号と名のペア、2 つ以上のフルネーム、またはこれらの名前要素の任意の組み合わせのいずれかである。例えば「Mr. and Mrs. John Smith」には等位結合された称号が含まれ、「Mr. and Mrs. John and Mary Smith」には等位結合された称号と等位結合された名が同じ構造に含まれている。

D

データ・リスト (**data list**)

外部データ・ソースによって作成される、メモリーに常駐する名前リスト。
1 つ以上のデータ・リストに対して検索要求が実行される。

発音区別符号 (**diacritic**)

1 つの文字、または文字の組み合わせの音価の変化を示す符号。

分散 (**distributed**)

コンピューティング環境のプログラムおよびコンピューター化された情報ソースに関する用語。物理的に別々のコンピューター・システムに配置されていても、単一の論理装置として作動する。

F

ファミリー・ネーム (家名、苗字、**family name**)

「姓 (surname)」を参照。

ファーストネーム (**first name**)

英語圏の名前において最初に出現するギブンネーム。「名 (ギブンネーム、given name)」および「ミドルネーム (middle name)」を参照。

G

名 (**given name**)

家族などの 1 つのグループの中で個人を識別するために使用される名前。
1 人の人が複数のギブンネームを有する場合もある。名つまりギブンネームは、個人名の主要な要素である。ギブンネームは、世界中のすべての名前に共通する唯一の名前要素であると考えられる。「姓 (surname)」、「個人名 (personal name)」も参照。

H

敬称 (**honorific**)

人が自ら獲得するかまたは人に授与された、社会的な地位を表す接頭部または接尾部。「修飾子 (qualifier)」、「称号 (title)」も参照。

I

イニシャル (**initial**)

名前の先頭文字を表す単一文字から成る名前トークン。単一文字の名前トークン (イニシャル) は、複数文字の名前トークン (名前) とは異なる仕方で扱われる。

J

JavaScript Object Notation (JSON)

JavaScript のオブジェクト・リテラル表記に基づいた、軽量のデータ交換フォーマット。JSON は特定のプログラミング言語に属しておらず、中立的であるが、C、C++、C#、Java、JavaScript、Perl、Python などの言語の規則を使用する。

JSON 「JavaScript Object Notation」を参照。

M

母方の名前 (matronymic)

その人の母親、または他の女性祖先の名前から派生した名前要素。

ミドルネーム (middle name)

英語圏の名前において 2 番目に出現するギブンネーム。「名 (ギブンネーム、given name)」および「ファーストネーム (first name)」を参照。

N

名前フィールド (name field)

1 つ以上の名前句または名前トークンから成るデータ構造。「名前句 (name phrase)」 「名前トークン (name token)」も参照。

名前句 (name phrase)

名前語幹と、その名前語幹に関連した接辞 (存在する場合) から成る分離不能な単位。名前句の中には複数の語幹から成るものもある (例えば中国語の名前 Mei-Hui、英語の名前 Mary-Anne)。1 つ以上の名前句を結合して名前フィールドを作成できる。「名前フィールド (name field)」、「名前語幹 (name stem)」、「名前トークン (name token)」も参照。

名前語幹 (name stem)

独立して、あるいは接辞や他の語幹と結合して名前全体または名前句を形成することのできる名前要素。「名前句 (name phrase)」 「名前トークン (name token)」も参照。

名前トークン (name token)

分割できない最小の名前要素。空白文字や句読法によって描出される。複数の名前トークンが組み合わされて名前句および名前フィールドが形成される。1 つの名前トークンの中に複数の名前パーツが含まれることがある。名前トークンは接辞、語幹のいずれかである。名前トークンの厳密な機能は、個人名の中で占める位置によって異なる。「名前フィールド (name field)」、「名前句 (name phrase)」、「名前語幹 (name stem)」も参照。

名前の変形 (name variant)

指定された名前に代わる名前。その名前と同じと見なされるが、特定の外形が異なる。名前の変形は、スペルの変形 (例えば "Geoff" と "Jeff")、ニックネーム (例えば "William" を表す "Bill")、省略形 (例えば "Guadalupe" を表す "GPE")、翻訳 (例えば "Pierre" を表す "Peter")、その他の過程によって発生する。

ニックネーム (nickname)

個人名の代替名。その他の名前要素から派生することが多い。

O

組織名 (organization name)

何らかの共通した機能を果たすために存在する、組織化された 1 人以上の集団を指す非人格名。通常、組織名には、それらを非個人名として特定するためのある種の標識またはパターン、つまり単語が含まれる。

P

解析 (parsing)

テキストを分析してその構造を判別し、テキストを個々のトークンに分割するプロセス。

解析された名前 (parsed name)

構文上の構造 (つまり名前句、名前フィールド、称号、修飾子) が定義されて解析処理の出力として表される名前。

父方の名前 (patronymic)

その人の父親、または他の男性祖先の名前から派生した名前要素。ファミリー・ネームとギブンネームはどちらも、世界のさまざまな場所で父方の名前として機能する。

個人名 (personal name)

特定の個人を指し示す名前。1 つ以上のギブンネーム (名)、姓、称号、または修飾子で構成される。完全な個人名は個人を表すものであり、場合によって、社会的階級、宗教的背景、政治的背景、教育水準、民族または国/地域の背景、および地域的由来を示す情報が織り込まれていることがあります。1 つの個人名は 1 つまたは 2 つの名前フィールドから成る。「名 (ギブンネーム、given name)」、「姓 (surname)」、「称号 (title)」、「修飾子 (qualifier)」、「名前フィールド (name field)」も参照。

精度 (precision)

取得されるすべてのデータに対する、関連性のあるデータの比率を指定する情報取得の尺度。精度は正の予測値である。情報取得は、精度およびリコールの両方を使用すると最も適切に計測される。「リコール (recall)」も参照。

接頭部 (prefix)

名前の最初に出現する接辞。例えば「de Rosa」というファミリー・ネームの中で、接辞「de」は接頭部である。

Q

修飾子 (qualifier)

個人名を区別するためにその名前の末尾に追加される用語または句。世代順序 (例えば Junior、Senior、フランス語で Junior は「fils」)、その人が獲得した学術的功績や宗教的階位 (例えば Ph.D.)、または何らかの職業上の資

格 (例えば D.D.S.) を指定する。名前の突き合わせでは、修飾子は個人名の中で末梢的な (重要度の低い) 部分と見なされる。「敬称 (honorific)」も参照。

R

リコール (recall)

関連性のある取得可能な全データの中で、実際に取得される関連データのパーセンテージを指定する情報取得の尺度。リコールは感度を示す尺度である。情報取得は、精度およびリコールの両方を使用すると最も適切に計測される。「精度 (precision)」も参照。

冗長度 (redundancy)

1 つのコンピューター・システム内で、同一の機能ユニット (ディスク・ドライブや電源機構システムなど) を複数使用すること。これにより、データ・セキュリティが確保され、ハードウェア障害に備えて一定レベルのフォールト・トレランスが実現される。

正規化 (regularization)

名前トークンを正規化してそれらの正規化された名前をデータ・リストに追加するプロセス。「名前トークン (name token)」および「データ・リスト (data list)」も参照。

リレーショナル・マーカ (relational marker)

個人間の家族関係を示す、個人名に含まれる用語。例えば「Karim bin Hassan」という名前の中で、リレーショナル・マーカ「bin」は「の息子」を意味します。

Representational State Transfer (REST)

ワールド・ワイド・ウェブなどの分散ハイパーメディア・システム用のソフトウェア・アーキテクチャー・スタイル。この用語はしばしば、SOAP などの追加メッセージング層を使用せずに、HTTP を介して XML (または YAML、JSON、プレーン・テキスト) を使用する、単純なインターフェースを指すときにも使用される。

REST 「Representational State Transfer」を参照。

ローマ字表記 (ラテン・アルファベット化、Romanization)

ローマ字 (ラテン・アルファベット) 以外のテキストをローマ字に音訳するプロセス。「音訳 (transliterate)」も参照。

S

検索方針 (search strategy)

既存の値またはデフォルト値をオーバーライドし、検索を実行するために使用される比較パラメータ値から成る名前付きコレクション。例えば既存の検索方針には、Standard (デフォルト値)、Broad (検索を広げる値)、Narrow (検索を制限する値) があります。管理者は独自の比較パラメータ値のセットを定義して、それを検索方針として保存することができます。

score 計算分析の結果。「スコアリング (scoring)」も参照。

スコアリング (scoring)

入力 ID の属性が既存のエントリ属性とどの程度一致しているかを計算するプロセス。

SOAP

非集中型の分散環境で情報を交換するための XML ベースの軽量プロトコル。SOAP を使用すると、情報を照会して返すことや、インターネットを介してサービスを呼び出すことができる。

ストップワード (stop word)

組織名に何ら意味を付与しない単語で、名前比較や名前スコアリングには使用されない。

サフィックス (suffix)

名前の末尾に出現する接辞。例えば「Nur-eddin」の中で、接辞「eddin」は接尾部である。

姓 (surname)

個人を人々のグループ (家族、部族、カーストなど) の一部として識別するため、名 (ギブンネーム) に追加される名前。姓は名と共に個人名の主要な要素となる。世界の中には、姓を使用しない場所もある。「名 (ギブンネーム、given name)」、「個人名 (personal name)」も参照。

構文 (syntax)

名前 (または他の式や句) に含まれる各要素の配置と相互関係。例えば英語の名前構文において、Todd Lane と Lane Todd ではギブンネームとファミリー・ネームが異なる。

T

TAQ 「称号、接辞、および修飾子 (title, affix, and qualifier)」を参照。

称号 (title)

個人名的一部分で、社会的、宗教的、または学術的な地位を表す「Dr.」、「Ms.」、「Colonel」など。個人名の中で称号はオプション部分であり、通常は名 (ギブンネーム) の前に置かれる。名前の突き合わせでは、称号は個人名の中で末梢的な (重要度の低い) 部分と見なされる。「敬称 (honorific)」も参照。

称号、接辞、および修飾子 (TAQ) (title, affix, and qualifier (TAQ))

あるテキストのストリングが名前を表すということを識別する助けになることが多い名前トークン。通常、これらのトークンは、名前に関する何らかの情報を示すが、実際の名前の一部になるのは接辞だけである。「称号 (title)」、「接辞 (affix)」、および「修飾子 (qualifier)」も参照。

音訳 (transliterate)

ローマ字以外の文字から、それに相当するローマ字にテキストを変換すること。247 ページの『ローマ字表記 (ラテン・アルファベット化、Romanization)』も参照。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することが

できます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

© Copyright IBM Corp. (2003, 2016). All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

本書では、IBM の商標および IBM 以外の商標の一部につき、それぞれが最初に出現する個所でマークを付けています。

IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

以下の会社名、製品名およびサービス名等は、それぞれ各社の商標または登録商標です。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

Intel、Intel (ロゴ)、Intel Inside、Intel Inside (ロゴ)、Intel Centrino、Intel Centrino (ロゴ)、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Cell Broadband Engine は、Sony Computer Entertainment, Inc.の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

米国郵政公社は、次の商標を所有しています。CASS、CASS Certified、DPV、LACSLink、ZIP、ZIP + 4、ZIP Code、Post Office、Postal Service、USPS and United States Postal Service。IBM Corporation は、米国郵政公社の非独占的な DPV および LACSLink のライセンス所有者です。

使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アーキテクチャー、製品
Global Name Management 10
新しい機能
バージョン 6.0 8
エラー・コード 205, 215, 220, 223
オーバーライド
比較パラメーター 66
お問い合わせ
IBM サポート 238
音訳 28
規則ファイル 29
音訳規則ファイル
Distributed Search プロセスで使用するための構成 127
IBM NameWorks で使用するための構成 201
NameHunter と連動して使用するための構成 127

[カ行]

解析
解析した名前 20
analyze() メソッドの使用 39
解析ツリー
作成 33
説明 18
外部トークン・リスト
説明 28
概要
Global Name Management 1
カスタム・トークン・リスト
説明 28
関連国
氏名についての識別 46
名と姓についての識別 46
analyze() メソッドを使用した識別 39
NameWorks を使用した識別 45
関連情報 v
技術リソース
検索 236
サポート・アップデートへのサブスクライブ 239

規則ファイル
音訳 29
機能
バージョン 6.0 の新機能 2
機能拡張
バージョン 6.0 8
国/地域別情報
国/地域別情報コード 41
氏名についての識別 42
組織名についての識別 43
名前フィールドについての識別 43
analyzeForSearch() メソッド 58, 70
NameWorks を使用した識別 40
国/地域別情報の識別
説明 40
NameWorks の使用 40
クライアント・アプリケーション 11
検索方針
管理
IBM NameWorks 63
削除 66
作成
構成ファイルを使用して 65
Strategy クラスを使用して 65
比較パラメーターのオーバーライド 68
変更
構成ファイルを使用して 65
IBM NameWorks 63
検索方針 (search strategy)
オーバーライド 66
サンプル
Broad 64
Narrow 64
Broad 64
Narrow 64
検索方針セクション 194
コード
国/地域別情報コード 41
理由コード 75
更新
データ・リスト上の名前 53
構成設定
変更および更新 186
語幹トークン
説明 27
個人名
構成要素および構造 14, 71
修飾子 23
接辞 23
タイトル 23
名前モデル 14, 71

個人名 (続き)
IBM NameWorks を使用した比較 76
TAQ 23
コメント
送信 v
コメントを送る v
固有の名前の検索 78
説明 62
補足データの取得 79
コンポーネント
API 10
コンポーネント・エラー・コード
一般エラー
GODC001E 210
GODC002E 210
GODC003E 210
GODC004E 210
GODC005E 210
GODC006E 210
GODD001E 210
GODD002E 210
GODD003E 210
GODD004E 210
GODD005E 210
GODD006E 210
GODH001E 210
GODH002E 210
GODH003E 210
GODH004E 210
GODH005E 210
GODH006E 210
GODL001E 210
GODL002E 210
GODL003E 210
GODL004E 210
GODL005E 210
GODL006E 210
GODP001E 210
GODP002E 210
GODP003E 210
GODP004E 210
GODP005E 210
GODP006E 210
GODS001E 210
GODS002E 210
GODS003E 210
GODS004E 210
GODS005E 210
GODS006E 210
GODT001E 210
GODT002E 210

コンポーネント・エラー・コード (続き)
一般エラー (続き)
GODT003E 210
GODT004E 210
GODT005E 210
GODT006E 210
GODV001E 210
GODV002E 210
GODV003E 210
GODV004E 210
GODV005E 210
GODV006E 210
GODW001E 210
GODW002E 210
GODW003E 210
GODW004E 210
GODW005E 210
GODW006E 210

[サ行]

削除

データ・リスト上の名前 54

サブスクリプト

サポート・アップデートへのサブスクリプト 239

サポート

お問い合わせ 238

サポート・アップデートへのサブスクリプト 239

トラブルシューティングにおける解決策の検索 236

暫定データ・ファイル 133

識別

氏名の関連国 46

氏名の国/地域別情報 42

氏名の性別 44

性別、NameWorks を使用した 44

組織名の国/地域別情報 43

名前カテゴリー 74

名前フィールドの国/地域別情報 43

名と姓の関連国 46

名の性別 45

システム管理者タスク

データ・リストの管理 51

氏名検索 78

修飾子

説明 21, 23

商標 253

使用法シナリオ

名前の検索 59

スキーマ

名前リストから Enterprise Name Search スキーマへの名前のロード 180

スキーマ (続き)

Enterprise Name Search スキーマからのすべての名前の削除 183

ストップワード

SW 26

姓

関連国の識別 46

国/地域別情報の識別 43

説明 21

名前変形の生成 49

生成

氏名の名前変形 48

名前フィールドの名前変形 49

製品の機能強化

バージョン 6.0 2

性別

氏名についての識別 44

名前についての識別 45

analyze() メソッドを使用した識別 39

NameWorks を使用した識別 44

セキュリティ

アクセス制御、Enterprise Name Search 171

パスワードの作成、Enterprise Name Search 168

パスワード・ファイル (ens.passwd)、Enterprise Name Search 166

ユーザーの削除、Enterprise Name Search 170

ユーザーへのセキュリティ・グループの割り当て、Enterprise Name Search 168

ユーザーへのロール・ベースのセキュリティ・グループの割り当て、Enterprise Name Search 168

ユーザー名の作成、Enterprise Name Search 168

ユーザー・パスワードの変更、Enterprise Name Search 170

ユーザー・パスワードのリセット、Enterprise Name Search 170

wspswd ユーティリティの構文、Enterprise Name Search 167

接辞

説明 23

説明 11

全検索

説明 62

先行結合子

説明 21

前提条件情報 v

組織指定子

OD 24

組織接辞

OA 27

組織名

ストップワード 24

説明 18, 73

組織指定子 24

組織接辞 24

TAQ 24

組織名 TAQ

ストップワード 26

組織指定子 24

組織接辞 27

[タ行]

タイトル

説明 21, 23

チェックリスト

一般的な製品トラブルシューティング・チェックリスト 203

知識ベース

トラブルシューティングにおける解決策の検索 236

注意事項

リリース 2

ツール

サポート・ツール 236

追加

データ・リスト上の名前 52

データ・リスト

管理 51

説明 52

名前と関連情報の更新 53

名前の削除 54

名前の追加 52

出生国

重要度 45

説明 45

頻度 45

confidence 45

トークン

語幹トークン 27

等位結合名 17

トラブルシューティング

一般的な製品トラブルシューティング・チェックリスト 203

サポート・アップデートへのサブスクリプト 239

知識ベースの検索 236

トレース 237

ログ・ファイルの確認 237

IBM サポートへのお問い合わせ 238

トレース

IBM NameWorks のログ・レベルの設定 237

[ナ行]

名前

- 音訳 28
- 解析した 20
- 概要 13
- カテゴリ 14, 71
- カテゴリ化 74
- 国/地域別情報の識別 40
- 検索 78
- 構成要素および構造 14, 71
- 語幹トークン 27
- 個人名 14, 71
- 氏名の国/地域別情報の識別 42
- 氏名の性別の識別 44
- 修飾子 23
- 性別の識別 45
- 接辞 23
- 組織名 18, 73
- 組織名の国/地域別情報の識別 43
- タイトル 23
- 等位結合 17
- 名前モデル 14, 71
- 補足データの取得 79
- NameWorks を使用した国/地域別情報の識別 40
- NameWorks を使用した名前変形の生成 47
- TAQ 23
- 名前カテゴリ 14, 71
 - 理由コード 75
- 名前句
 - 関連国の識別 46
 - 説明 22
- 名前突き合わせ
 - 概要 13
 - 方法 13
- 名前データ・アーカイブ
 - 説明 27
- 名前トークン
 - 説明 22, 88
- 名前の解析 33
 - 解析ツリーの説明 18
 - 検索のための 58, 70
 - 個々の部分に 33
 - analyzeForSearch() メソッド 58, 70
 - NameWorks の使用 33
- 名前の検索 51, 78
 - 補足データの取得 79
 - analyzeForSearch() メソッド 58, 70
 - Enterprise Name Search の使用 166
 - IBM NameWorks 使用 62
 - NameHunter の使用 80
- 名前の正規化 91
- 名前のパーツ 18

名前の比較

- IBM NameWorks 使用 76
- 名前の分析 39
 - 名前の完全分析の実行 39
- 名前の分類
 - 国/地域別情報 58, 70
- 名前の変形
 - 氏名についての生成 48
 - 説明 47
 - 名と姓についての生成 49
 - NameWorks を使用した生成 47
- 名前の前処理 133
- 名前フィールド
 - 氏名の名前変形の生成 48
 - 修飾子 21
 - 姓 21
 - 説明 21
 - 先行結合子 21
 - タイトル 21
 - 名前フィールドの国/地域別情報の識別 43
 - 名 21
 - 名と姓の名前変形の生成 49
- 名前リスト
 - 外部トークン・リスト 28
 - カスタム・トークン・リスト 28
 - 説明 27
 - Enterprise Name Search の名前リストの管理 137, 172

[ハ行]

- バージョン 6.0 の新機能 2
- 比較パラメーター 98
 - オーバーライド 66
 - 検索方針 68
 - CompParmOverrides クラス 69
 - Configuration クラス 69
 - Distributed Search 118
 - NameHunter 118
- 日付
 - IBM NameWorks を使用した比較 77
 - IBM NameWorks を使用した日付の差の判別 77
- 日付の比較
 - IBM NameWorks 使用 77
 - IBM NameWorks を使用した日付の差の判別 77
- ファイル
 - パスワード・ファイル (ens.passwd)、Enterprise Name Search 166
- 変形
 - 氏名についての生成 48
 - 説明 47
 - ファイル・フォーマット 87

変形 (続き)

- 名と姓についての生成 49
- NameWorks を使用した生成 47
- 変形名前形式
 - analyze() メソッドを使用したリストの生成 39
- 変形名
 - 変換ユーティリティー 88
- 変更 98
- 法的なガイドライン
 - 使用条件 255
 - 資料
 - 再使用 255

[マ行]

名

- 関連国の識別 46
- 国/地域別情報の識別 43
- 説明 21
- 名前変形の生成 49
- メソッド
 - categorize() 74
 - compare() 76
 - dateCompare() 77
 - dateDifference() 77
 - genderizeField() 45
 - genderize() 44
 - getVariants() 48, 49
 - parse() 33
 - search() 62
- 問題と回避策
 - 知識ベースの検索 236

[ヤ行]

ユーティリティー

- wspswd ユーティリティーの構文、Enterprise Name Search 167
- 用語
 - ファイル・フォーマット 89
- 用語集 243

[ラ行]

理由コード

- 名前カテゴリ 75
- ログ・ファイル
 - トラブルシューティング 237
 - IBM NameWorks のトレース・ログ・レベルの設定 237

A

- addNameToDatalist() メソッド 52

addName()
メソッド 52
analyzeForSearch()
メソッド 58, 70
analyzeForSearch() メソッド 58, 70
QueryName オブジェクトの作成 60
analyze()
メソッド 39
analyze() メソッド 39
API
エラー・コード 205
arabicTransRule.ibm
Distributed Search プロセスで使用する
ための構成 127
NameHunter と連動して使用するた
めの構成 127

C

categorize() メソッド 74
compare() メソッド 76
Comparison Files セクション 197
CompParms 98
Distributed Search 118
NameHunter 98, 118
CompParmsOverrides クラス
オーバーライド 66
比較パラメーターのオーバーライド 69
Configuration クラス
オーバーライド 66
構成設定の指定 200
比較パラメーターのオーバーライド 69
createName() メソッド
Name オブジェクトの作成 60
Custom Tokens セクション 189
cyrillicTransRule.ibm
Distributed Search プロセスで使用する
ための構成 127
NameHunter と連動して使用するた
めの構成 127
C++ エラー・コード 216

D

dataFetch()
メソッド 79
dataFetch() メソッド 79
Datalist セクション 190
DateCompare セクション 195
dateCompare() メソッド 77
dateDifference() メソッド 77
deleteNameFromDatalist()
メソッド 54
deleteNameFromDatalist() メソッド 54
Distributed Search 154, 220

Distributed Search (続き)
エラー・コード 220
サーバー・アプリケーション 12
IBM NameWorks 構成ファイル 190
Distributed Search XML メッセージ・イ
ンターフェース 153
Distributed Search XML 要求 154
Distributed Search プロセス
音訳規則ファイルを使用するための構
成 127
DS XML 要求 154

E

Embedded Search
名前リストの前処理 131
IBM NameWorks 構成ファイル 191
Enterprise Name Search 223
アクセス制御 171
エラー・コード 223
セキュリティ、wspswd ユーティリ
ティの構文 167
セキュリティ・グループへのユーザ
ーの割り当て 168
データベース統計の更新 182
名前リストからスキーマへの名前のロ
ード 180
名前リストからの名前の削除 183
名前リストの管理 137, 172
パスワード・ファイル
(ens.passwd) 166
ユーザーの削除 170
ユーザー名およびパスワードの作成
168
ユーザー・セキュリティの管理 166,
170
ユーザー・セキュリティの管理、ユ
ーザー・パスワードのリセット 170
ユーザー・パスワードの変更 170
ユーザー・パスワードのリセット 170
Enterprise Name Search スキーマか
らのすべての名前の削除 183
wspswd ユーティリティの構文 167

G

genderizeField() メソッド 45
genderize() メソッド 44
General セクション 188
getVariants() メソッド 48, 49
Global Name Management
説明 1
greekTransRule.ibm
Distributed Search プロセスで使用する
ための構成 127

greekTransRule.ibm (続き)
NameHunter と連動して使用するた
めの構成 127

I

IBM NameWorks 11, 216
音訳規則ファイルを使用するための構
成 201
管理
検索方針 63
組み込み検索 131
検索方針 63
検索方針の削除 66
更新済みデータのロード 186
構成 185
構成ファイル 186
データ・リストの管理 51
比較パラメーターのオーバーライド 66
マイグレーション情報 55
API 11
Configuration クラス 200
IBM NameWorks 構成ファイル 194
Distributed Search 190
Embedded Search 191
IBM サポート
お問い合わせ 238
ibmgnr::Exception 205
ibmgnr::NwException 215
IPv6 接続
IBM NameWorks 構成ファイル 190

L

LDAP 166

N

Name Loader ユーティリティ
名前リストからの名前のロード 180
Name Preprocessor 133
Name オブジェクト
作成 60
NameHunter
音訳規則ファイルを使用するための構
成 127
比較パラメーター 98
CompParms 98
NameHunter CompParms
韓国語 - ロシア語の国/地域別情報
123
グループの国/地域別情報 126
スペイン語 - 日本語の国/地域別情報
122

NameHunter CompParms (続き)
タイ語 - ヨルバ語の国/地域別情報
124
中国語 - ドイツ語の国/地域別情報
121
ビジネス名 124
未確定 - アラビア語の国/地域別情報
119
NameHunter デベロッパーズ・ツールキ
ット 93
NameHunter 名前トークンの変形 86
NameHunter の TAQ データ 81
NameHunter の why サンプル・アプリ
ケーション 97
NameHunter の概要 80
NameHunter のサンプル検索アプリケー
ション 97
NameHunter のサンプル・アプリケーシ
ョン 97
NameLoader 構成ファイル 177
NameLoader コマンド 176
NameLoader ユーティリティ
名前リストからの名前の削除 183
Enterprise Name Search スキーマか
らのすべての名前の削除 183
NameWorks 215
エラー・コード 215
国/地域別情報の識別 40
NameWorks 構成ファイル 188, 189, 190,
195, 196, 197
NDA
説明 27

P

parse() メソッド 33

Q

QueryName オブジェクト
作成 60

R

Reference Files セクション 197
RSS フィード
トラブルシューティング 239

S

search()
メソッド 78
search() メソッド 62, 78
Strategy クラス
オーバーライド 66

T

TAQ

ストップワード 24
説明 23
組織指定子 24
組織接辞 24
ファイル・フォーマット 81, 86
Transliteration Modules セクション 196

U

updateNameInDatalist() メソッド 53
updateName()
メソッド 53

[特殊文字]

[Datalist:] セクション
CompParmsDefaults= 66



Printed in Japan