

Natural for Windows

オペレーション

バージョン 6.3.3

October 2008

This document applies to Natural バージョン 6.3.3 and to all subsequent releases.

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

Copyright © Software AG 1992-2008. All rights reserved.

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

目次

| | |
|--|----|
| 1 オペレーション | 1 |
| 2 Natural を実行するための Microsoft Windows XP パーソナルファイアウォール の設定 | 3 |
| メソッド 1 - 特定の実行ファイルを許可してポートを開く | 4 |
| メソッド 2 - 特定のポートを PC 上で使用することを許可する | 5 |
| 実行ファイルとポート番号の概要 | 5 |
| 3 プロファイルパラメータの使用方法 | 7 |
| パラメータの階層 | 8 |
| パラメータ値のスタティックな割り当て | 9 |
| パラメータ値のダイナミックな割り当て | 9 |
| パラメータ値のランタイム割り当て | 10 |
| 4 システムファイル | 13 |
| システムファイル構造 | 14 |
| システムファイル FNAT と FUSER | 15 |
| システムファイル FDDM | 17 |
| 重要な情報および警告 | 21 |
| ファイル FILEDIR.SAG | 21 |
| 移植可能な Natural システムファイル | 22 |
| 5 ワークファイル | 25 |
| ワークファイルの定義 | 26 |
| ワークファイルフォーマット | 29 |
| 拡張子 NCD を持つワークファイルについての特別な考慮 | 33 |
| 6 Natural バッファプール | 35 |
| 全般的な情報 | 36 |
| バッファプールの設定 | 40 |
| Natural バッファプールサービスの使用 | 40 |
| ユーティリティ NATBPSRV を使用したバッファプールの作成 | 44 |
| バッファプールのモニタリング | 44 |
| トラブルシューティング | 45 |
| 7 GUI バージョンのバッファプールモニタの使用 | 47 |
| バッファプールモニタの開始と終了 | 48 |
| Natural バッファプールモニタウィンドウの要素 | 49 |
| バッファプールの切断と接続 | 52 |
| バッファプールサーバーのシャットダウン | 53 |
| バッファプールサーバーの開始 | 54 |
| バッファプールモニタのプロパティの変更 | 55 |
| 一般情報 | 56 |
| バッファプールコンテンツ | 58 |
| グラフィックアナライザ | 61 |
| レポート | 66 |
| 8 コマンド行バージョンのバッファプールモニタの使用 (NATBPMON) | 71 |
| NATBPMON ユーティリティの呼び出し | 72 |
| NATBPMON コマンド | 73 |

| | |
|---|-----|
| バッファプールのオブジェクトの表示 | 74 |
| パターンの指定 | 75 |
| バッファプール設定の表示 | 76 |
| バッファプールの統計情報 | 77 |
| 9 バッチモードでの Natural | 81 |
| バッチモードについて | 82 |
| バッチモードの Natural セッションの開始 | 82 |
| バッチモードの Natural セッションの終了 | 83 |
| バッチモードでの Natural の使用 | 83 |
| バッチモードのサンプルセッション | 85 |
| バッチモード検出 | 88 |
| バッチモードの制限事項 | 89 |
| バッチモードで Natural マップおよびダイアログを使用するためのヒント | 89 |
| 10 出力ウィンドウ | 91 |
| 出力ウィンドウについて | 92 |
| 出力ウィンドウでの作業 | 92 |
| 出力ウィンドウプロファイルの変更 | 93 |
| 出力ウィンドウ用の独自のアイコンの使用 | 94 |
| 11 Natural ランタイム | 97 |
| Natural ランタイムの対象外について | 98 |
| 移植手順の概要 | 99 |
| 手順 1：開発ワークステーションでのアプリケーションのパッケージ | 100 |
| 手順 2：Natural ランタイムのインストール | 105 |
| 手順 3：ランタイムワークステーションへのアプリケーションのインストール | 105 |
| 手順 4：ランタイムワークステーションでのアプリケーションの開始 | 107 |
| Natural ランタイム起動サービスの使用 | 108 |
| 12 NATCONVINI での異なる文字セットのサポート | 113 |
| 異なる文字セットのサポートの重要性 | 114 |
| 異なる文字セットの使用方法 | 114 |
| 13 Natural 出口コード | 119 |
| Natural スタートアップエラー | 120 |
| 14 Entire System Server インターフェイスの設定 | 123 |
| 前提条件 | 124 |
| 有効化 | 124 |
| Entire System Server DDM のデータベース ID の変更 | 125 |
| 15 Web I/O インターフェイスサービスの設定 | 127 |
| Web I/O インターフェイスサービスのインストール | 128 |
| Web I/O インターフェイスサービスの設定 | 129 |
| イベントログ | 131 |
| Natural を開始するためのバッチファイル | 132 |
| 16 Web I/O インターフェイスクライアントの設定 | 135 |
| 17 前提条件 | 137 |
| アプリケーションサーバーまたは Web サーバー | 138 |
| Apache Ant | 139 |

| | |
|---|-----|
| Natural for UNIX | 139 |
| Natural for Mainframes | 139 |
| Natural for Windows | 140 |
| クライアントとサーバーの互換性 | 140 |
| ブラウザの前提条件 | 141 |
| 18 Sun Java System Application Server への Web I/O インターフェイスクライアントのインストール | 143 |
| インストール手順 | 144 |
| インストールの確認 | 151 |
| 19 JBoss Application Server への Web I/O インターフェイスクライアントのインストール | 153 |
| インストール手順 | 154 |
| インストールの確認 | 157 |
| 20 Microsoft Internet Information Services (IIS) への Web I/O インターフェイスクライアントのインストール | 159 |
| インストール手順 | 160 |
| インストールの確認 | 162 |
| 21 コンフィグレーションファイルでの Natural セッションの定義 | 163 |
| ログオンページについて | 164 |
| コンフィグレーションファイルの名前と場所 | 165 |
| コンフィグレーションファイルの内容 | 166 |
| コンフィグレーションファイル要素の概要 | 167 |
| ログオンページでのパスワードの変更 | 171 |
| タイムアウトの設定 (J2EE のみ) | 172 |
| ブラウザの制限 | 172 |
| 22 色属性とフィールド属性の変更 (J2EE のみ) | 173 |
| 前景色の設定 | 174 |
| 背景色の設定 | 175 |
| PF キーボタンの適合 | 177 |
| 反転属性の設定 | 177 |
| 下線属性と点滅属性の設定 | 178 |
| 筆記体/イタリック体属性の設定 | 178 |
| 高輝度属性の設定 | 179 |
| 23 テーマの使用 (IIS のみ) | 181 |
| スキンファイル | 182 |
| スタイルシート | 184 |
| 24 Web I/O インターフェイスクライアントと端末エミュレーションの違い | 187 |
| 25 Web I/O インターフェイスを Natural アプリケーションで使用する場合の制限 | 189 |
| 26 NaturalX アプリケーションの管理 | 191 |
| 27 NaturalX サーバー | 193 |
| COM クラスとサーバー | 194 |
| NaturalX クラスとサーバー | 194 |
| Windows での NaturalX サーバーと Natural セッション | 194 |
| サーバー ID の役割 | 195 |

| | |
|---|-----|
| サーバー ID の編成 | 196 |
| 28 アクティベーションポリシー | 197 |
| Windows 環境のアクティベーションポリシー | 198 |
| アクティベーションポリシーの設定 | 199 |
| アクティベーションポリシーを使用するタイミング | 199 |
| 29 登録 | 205 |
| Natural での登録 | 206 |
| 自動登録 | 206 |
| 手動登録 | 207 |
| 登録ファイルとタイプライブラリ | 210 |
| クライアントの登録 | 210 |
| 登録のヒント | 211 |
| 30 タイプ情報 | 213 |
| 概要 | 214 |
| NaturalX とタイプ情報 | 214 |
| タイプ情報の使用 | 214 |
| 31 コンフィグレーション概要 | 219 |
| サーバーのコンフィグレーション - 一般的な設定 | 220 |
| サーバーのコンフィグレーション - アプリケーション固有の設定 | 221 |
| クライアントのコンフィグレーション - 一般的な設定 | 222 |
| クライアントのコンフィグレーション - アプリケーション固有の設定 | 222 |
| 32 NaturalX でのセキュリティ | 223 |
| 概要 | 224 |
| アクティベーションセキュリティ | 224 |
| コールセキュリティ | 225 |
| 33 Windows 上での DCOM コンフィグレーション | 229 |
| NaturalX サーバーの構成 | 230 |
| NaturalX クライアントの構成 | 240 |
| 34 NaturalX システムレジストリエントリ | 245 |
| サーバーのためのレジストリエントリ | 246 |
| クライアントのためのレジストリエントリ | 247 |
| 35 NaturalX サーバー環境でのステートメントおよびコマンドの使用 | 249 |
| Natural ステートメント | 250 |
| Natural システムコマンド | 251 |
| 索引 | 253 |

1 オペレーション

このドキュメントでは、Windows 環境での Natural の操作について説明します。次の項目で構成されています。

| | |
|---|---|
|  Natural を実行するための Microsoft Windows XP パーソナルファイアウォールの設定 | Windows ファイアウォールで保護された環境での Natural の実行方法。 |
|  プロファイルパラメータの使用法 | パラメータ階層に関する情報。プロファイルパラメータに値をスタティックに、ダイナミックに、およびランタイム時に割り当てる方法。 |
|  システムファイル | システムファイルおよび Natural オブジェクトをファイルシステムに保存する方法。システムファイル FNAT、FUSER、および FDDM に関する情報。 |
|  ワークファイル | ワークファイルの定義方法。異なるワークファイルフォーマットに関する情報。 |
|  Natural バッファプール | Natural によるバッファプールの使用方法および開始方法。 |
|  GUI バージョンのバッファプールモニタの使用 | バッファプールとの接続および切断方法と、バッファプールサーバーのシャットダウンおよび開始方法。バッファプールモニタを使用して表示できる情報に関する説明。 |
|  コマンド行バージョンのバッファプールモニタの使用 (NATBPMON) | NATBPMON ユーティリティの呼び出し方法。このユーティリティで使用できるコマンドに関する情報。 |
|  バッチモードでの Natural | バッチモードで Natural を実行する方法。必要な入力および出力チャンネルに関する情報。 |
|  出力ウィンドウ | 出力ウィンドウを使用する方法、出力ウィンドウプロファイルを変更する方法、および出力ウィンドウに独自のアイコンを使用する方法。 |
|  Natural ランタイム | アプリケーションを開発ワークステーションからランタイムワークステーションに移植する方法。Natural ランタイムプロセスを開始するためのサービスを使用する方法。 |

| | |
|--|---|
|  NATCONV.INI での異なる文字セットのサポート | ファイル NATCONV.INI で異なる文字セットを定義する方法。 |
|  Natural 出口コード | スタートアップエラーを含む、出口コードに関する情報。 |
|  Entire System Server インターフェイスの設定 | Entire System Server 製品用の Entire System Server インターフェイスをアクティブにする方法。 |
|  Web I/O インターフェイスサービスの設定 | ランタイム環境で使用する Web I/O インターフェイスサービスを設定する方法。 |
|  Web I/O インターフェイスクライアントの設定 | Web I/O インターフェイスクライアントをランタイム環境で使用できるように、アプリケーションサーバーまたは Web サーバーをインストールおよび設定する方法。Web I/O インターフェイスクライアントと端末エミュレーションの違い。Web I/O インターフェイスを Natural アプリケーションで使用する場合の制限。 |
|  NaturalX アプリケーションの管理 | NaturalX クラスで構成されるアプリケーションを、DCOM を使用して複数のプロセスおよびマシンに配布する方法。 |

多数の管理機能を実行するために使用できる Natural ユーティリティについては、個別に説明しています。詳細については、『[ツールおよびユーティリティ](#)』ドキュメントを参照してください。

また、Natural Security についても、個別に説明しています。詳細については、『[Natural Security](#)』ドキュメントを参照してください。

 **Note:** Natural ランタイムバージョンを購入した場合は、Natural ランタイムバージョンには Natural 開発ツールが含まれていないことに注意してください。また、Natural ランタイムバージョンでは、使用できる Natural システムコマンドに制限があります。

2 Natural を実行するための Microsoft Windows XP パーソナルファイアウォールの設定

- メソッド 1- 特定の実行ファイルを許可してポートを開く 4
- メソッド 2- 特定のポートを PC 上で使用することを許可する 5
- 実行ファイルとポート番号の概要 5

Natural を実行するための Microsoft Windows XP パーソナルファイアウォールの設定

Microsoft Windows XP Service Pack 2 以降では、パーソナルファイアウォールを使用できます。Service Pack のデフォルトの設定では、ファイアウォールは有効になっています。Natural を許可するようにファイアウォールを設定しないと、Natural を起動できません。

 **Caution:** ファイアウォールを無効にすることはお勧めしません。ファイアウォールの無効化は、ユーザー自身の責任で行ってください。

Windows XP ファイアウォールの設定方法については、Windows XP Service Pack 2 に関する Microsoft ドキュメントを参照してください。

このchapterでは、Windows XP ファイアウォールで保護された環境で Natural を実行する方法の例を示します。ただし、これらの例は単なる技術的なガイドラインであり、ユーザーに必要なセキュリティを提供するとは限りません。

例は、2つのメソッドに基づいています。1つのメソッドでは、特定の実行ファイルを許可してポートを開くことができますようにします。もう1つのメソッドでは、特定のポートをPC上の特定のプログラムが使用できるようにします。これらのメソッドでは、Naturalを例として使用します。他の Natural コンポーネントについては、「[実行ファイルとポート番号の概要](#)」に関連する情報を参照してください。

このchapterでは、次のトピックについて説明します。

メソッド 1 - 特定の実行ファイルを許可してポートを開く

このメソッドでは、Natural 実行ファイルを「許可されたプログラム」として追加します。これにより、TCP および UDP 通信の両方で任意のポートを開くことができるようになります。

これらのパラメータは、インストール処理中にカスタマイズできます。デフォルト設定を使用してインストールしていない場合、カスタムパラメータを使用する必要があります。

以降の例は、Natural 実行ファイルに適用されます。他の Natural または Entire Access コンポーネントを許可されたプログラムとして追加するには、後述の「[実行ファイルとポート番号の概要](#)」を参照してください。

 **Note:** 以下の例では、Windows XP および Natural for Windows バージョン 6.3 のデフォルトのインストール設定を使用します。

▶手順 2.1. Natural を許可されたプログラムとして追加するには

■ 次のコマンドを入力します。

```
netsh firewall add allowedprogram program="C:\Program Files\Software AG\Natural\6.3\Bin\natural.exe" name="Natural" profile=ALL
```

▶手順 2.2. Natural を許可されたプログラムから除外するには

- 次のコマンドを入力します。

```
netsh firewall delete allowedprogram program="C:\Program Files\Software AG\Natural\6.3\Bin\natural.exe" profile=ALL
```

メソッド 2 - 特定のポートを PC 上で使用することを許可する

このメソッドでは、特定のポートを開きます。

以降の例は、Natural 実行ファイルに適用されます。他の Natural または Entire Access コンポーネント用にポートを開くには、後述の「[実行ファイルとポート番号の概要](#)」を参照してください。

▶手順 2.3. 特定のポートを開くには

- 次のコマンドを入力します。

```
netsh firewall add portopening protocol=TCP port=nnnnn name="Natural" profile=ALL
```

nnnnn は開くポートの番号です。

▶手順 2.4. 特定のポートを閉じるには

- 次のコマンドを入力します。

```
netsh firewall delete portopening protocol=TCP port=nnnnn profile=ALL
```

nnnnn は閉じるポートの番号です。

実行ファイルとポート番号の概要

Natural およびそのサブプログラムをすべて実行するには、使用している機能に応じて、さまざまな通信ポートを開く必要があります。以下に、通信ポートを確立する必要があるプログラムのリストを示します。PC で使用するプログラムまたはポートを選択できます。

必要な構文については、「[メソッド 1 - 特定の実行ファイルを許可してポートを開く](#)」および「[メソッド 2 - 特定のポートを PC 上で使用することを許可する](#)」を参照してください。

 **Important:** 次のリストに記載されているファイルの場所およびポート番号は、デフォルト設定です。これらのパラメータは、インストール処理中にカスタマイズできます。デフォ

Natural を実行するための Microsoft Windows XP パーソナルファイアウォールの設定

ルト設定を使用してインストールしていない場合、カスタムパラメータを使用する必要があります。

| コンポーネント | メソッド1 | | メソッド2 |
|----------------------|--------------------------|---|-------------|
| | 実行ファイル | ファイルの場所 | デフォルトのポート番号 |
| Natural | <i>natural.exe</i> | <i>C:\Program Files\Software AG\Natural\6.3\bin\</i> | |
| デバッガ | <i>natdbgsrv.exe</i> | <i>C:\Program Files\Software AG\Natural\6.3\bin\</i> | 2600 |
| 端末エミュレーション | <i>natpccserver2.exe</i> | <i>C:\Program Files\Software AG\Natural\6.3\Terminal\</i> | 22334 |
| Entire Access サーバー | <i>serversingle.exe</i> | <i>C:\Program Files\Software AG\Entire Access\6.1.1\bin</i> | |
| Entire Access クライアント | <i>vtx3.dll</i> | <i>C:\Program Files\Software AG\Entire Access\6.1.1\bin</i> | |

3 プロファイルパラメータの使用法

- パラメータの階層 8
- パラメータ値のスタティックな割り当て 9
- パラメータ値のダイナミックな割り当て 9
- パラメータ値のランタイム割り当て 10

Natural プロファイルパラメータは、操作環境の外観とレスポンスに影響します。

このchapterでは、次のトピックについて説明します。

パラメータの詳細については、『パラメータリファレンス』を参照してください。

パラメータの階層

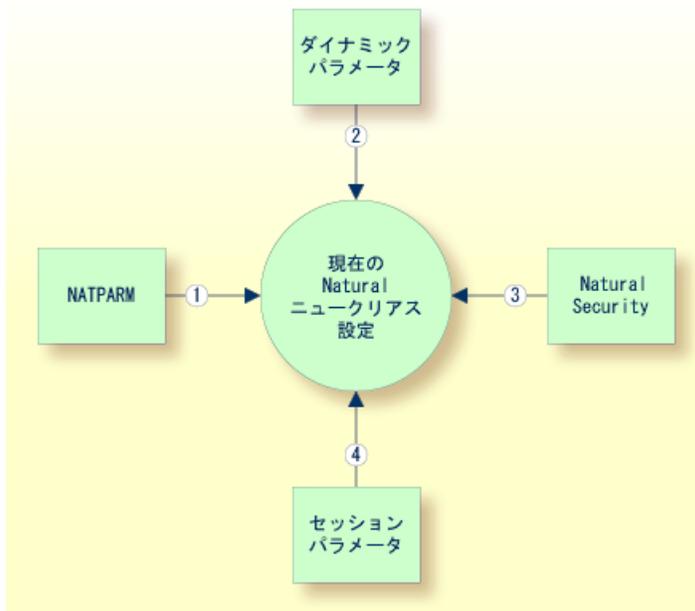
Naturalパラメータの値は、さまざまなソースから取得されます。パラメータの優先順位は次のとおりです。

1. スタティックな割り当て
最も低い優先順位。スタティックな割り当ては、Natural パラメータファイル NATPARM に指定されたパラメータによって行われます。
2. ダイナミックな割り当て
ダイナミックな割り当ては、Naturalの起動時に代替パラメータファイルや個別のパラメータを指定することによって行われます。
3. ランタイム割り当て
最も高い優先順位。ランタイム割り当ては、セッションパラメータを指定することによってセッション中に行われます。

各タイプの割り当ての詳細については、このセクションで後述します。

 **Note:** Natural Security がアクティブな場合は、特定のパラメータの使用が制限されることがあります。

次の図は、パラメータ階層を示したものです。



パラメータ値のスタティックな割り当て

デフォルトでは、パラメータファイル NATPARM のパラメータ指定は Natural 環境の特性を決定するために使用されます。初期状態では、このファイルには Software AG が提供するデフォルト値が含まれます。この値は、コンフィグレーションユーティリティを使用して変更できます。

 **Tip:** デフォルトのパラメータファイル NATPARM を変更することはお勧めしません。Natural にデフォルト値以外のパラメータ値を使用する場合は、ユーザー独自のパラメータファイルを作成してください（次のセクションも参照）。

パラメータ値のダイナミックな割り当て

ダイナミックパラメータを使用することによって、Natural の起動時に独自の環境を設定できます。セッションが開始されると、オペレーティングシステムはダイナミックパラメータの値を Natural に渡します。

ダイナミックパラメータは、現在の Natural セッションに対して有効です。これらは、デフォルトのパラメータファイル NATPARM に指定されているスタティックな割り当てを上書きします。

コンフィグレーションユーティリティを使用して、独自のパラメータファイルを作成することもできます。独自のパラメータファイルを使用するには、Natural の起動時にその名前を指定する必要があります。

▶手順 3.1. ダイナミックパラメータ値を使用して Natural を起動するには

- Natural の起動に使用するコマンドに、ダイナミックパラメータおよびその値を追加します。

例：プロファイルパラメータ PARM を使用して、代替パラメータファイル MYPARM で Natural を呼び出します。プロファイルパラメータ SM および DTFORM の値が、MYPARM に定義されている値の代わりに使用されます。

```
natural PARM=MYPARM SM=ON DTFORM=I
```

Or:

ショートカットを使用して Natural を起動する場合は、ダイナミックパラメータを次の例のように指定します。

```
"C:\Program Files\Software AG\Natural\n.n\bin\natural.exe" PARM=MYPARM SM=ON  
DTFORM=I
```

n.n は現在のバージョン番号です。

特殊文字

カッコやアスタリスクなどの特殊文字は、オペレーティングシステムによって解釈されます。そのため、これらの特殊文字を使用するパラメータは二重引用符で囲む必要があります。例えば、次のようになります。

```
natural "FNAT=(99,30) FUSER=(99,32)"
```

この規則の例外として、パラメータ FNAT、FDIC、FSEC、FDDM、および FUSER は、カッコなしで指定して引用符の使用を避けることもできます。例えば、次のようになります。

```
natural FNAT=99,30 FUSER=99,32
```

パラメータ値のランタイム割り当て

ランタイム割り当ては、セッションパラメータを設定することによってセッション中に行われます。セッションパラメータ値は、スタティックおよびダイナミックな割り当てを上書きします。

Natural スタジオを使用する場合は、ウィンドウでセッションパラメータを設定できます（これは、パラメータなしにシステムコマンド GLOBALS を発行することに相当します）。『Natural スタジオの使用』ドキュメントの「セッションパラメータの使用」を参照してください。

セッションパラメータは、システムコマンド GLOBALS で設定することもできます。例えば、次のようになります。

```
GLOBALS SA=ON,IM=D
```

レポートモードでは、セッションパラメータをプログラムの SET GLOBALS ステートメントで設定することもできます。例えば、次のようになります。

```
SET GLOBALS SA=ON,IM=D
```

 **Note:** セッションパラメータは、セッションレベルでの設定（上記参照）の他に、プログラム、ステートメント、またはフィールドレベルで設定することもできます。詳細については、『パラメータリファレンス』の「セッションパラメータについて」を参照してください。

4 システムファイル

| | |
|---|----|
| ■ システムファイル構造 | 14 |
| ■ システムファイル <code>FNAT</code> と <code>FUSER</code> | 15 |
| ■ システムファイル <code>FDDM</code> | 17 |
| ■ 重要な情報および警告 | 21 |
| ■ ファイル <code>FILEDIR.SAG</code> | 21 |
| ■ 移植可能な Natural システムファイル | 22 |

Natural for Windows では、オペレーティングシステム機能からアクセスできるファイルにオブジェクトを保存します。オブジェクトが Adabas システムファイルに保存される Natural for Mainframes とは異なり、Natural for Windows ではオブジェクトをディスク上の特定のディレクトリに保存します。このため、Natural for Windows の実行に Adabas などのデータベースは必要ありません。

このchapterでは、次のトピックについて説明します。

システムファイル構造

デフォルトでは、Natural ライブラリは、特定の Natural バージョンの Natural ルートディレクトリにサブディレクトリとして作成されます。このサブディレクトリには、ライブラリと同じ名前が付けられます。

Natural オブジェクトは、サブディレクトリにファイルとして保存されます。Natural オブジェクトのファイル名の形式は次のとおりです。

`file-name.NKT`

| | | |
|------------------------|--|---------|
| <code>file-name</code> | オブジェクトの名前。『Natural スタジオの使用』の「オブジェクトの命名規則」も参照してください。 | |
| <code>N</code> | 拡張子の最初の文字は常に "N" です。これは「Natural」を示します。 | |
| <code>K</code> | 拡張子の 2 番目の文字は次のいずれかになります。 | |
| | <code>S</code> | ソースファイル |
| | <code>G</code> | 生成プログラム |
| | <code>R</code> | リソース |
| <code>T</code> | 拡張子の 3 番目の文字はオブジェクトのタイプを示します。有効な値については、次のリストを参照してください。 | |

例えば、ソースプログラム TESTPROG はファイル TESTPROG.NSP として保存されますが、マップ TESTMAP に対して生成されたコードはファイル TESTMAP.NGM として保存されます。

 **Note:** ファイル名は常にオブジェクト名と同一というわけではありません。現在のオブジェクト名および対応する内部オブジェクト名は両方とも **FILEDIR.SAG** ファイルに記述されます。

オブジェクトタイプを示す次の文字と数字を拡張子に使用できます。

| 文字または番号 | オブジェクトタイプ |
|---------|-------------------|
| A | パラメータデータエリア (PDA) |
| C | コピーコード |
| D | DDM |
| G | グローバルデータエリア (GDA) |
| H | ヘルプルーチン |
| L | ローカルデータエリア (LDA) |
| M | マップ |
| N | サブプログラム |
| P | プログラム |
| S | サブルーチン |
| T | テキスト |
| 3 | ダイアログ |
| 4 | クラス |
| 5 | コマンドプロセッサ |
| 6 | ビュー調整 |
| 7 | ファンクション |
| 8 | アダプタ |

システムファイル_{FNAT} と_{FUSER}

Natural システムファイル _{FNAT} (システムプログラム用) と _{FUSER} (ユーザープログラム用) は、異なるサブディレクトリに置かれます。

_{FNAT} では、次のディレクトリ構造が想定されます。

```

FNAT
├── LIBDIR.SAG
├── SYSTEM
│   ├── FILEDIR.SAG
│   ├── SRC
│   ├── GP
│   ├── ERR
│   └── RES
├── SYS*
│   └── FILEDIR.SAG

```

```
└─ SRC
└─ GP
└─ ERR
└─ RES
```

FNAT のみで使用できるファイル *LIBDIR.SAG* には、Natural を使用する他のインストール済み Software AG 製品に関する情報がすべて含まれています。この情報は、システムコマンド *SYSPROD* を使用して表示できます。

FUSER では、次のディレクトリ構造が想定されます。

```
FUSER
└─ SYSTEM
    └─ FILEDIR.SAG
        └─ SRC
            └─ GP
                └─ ERR
                    └─ RES
└─ user-library1
    └─ FILEDIR.SAG
        └─ SRC
            └─ GP
                └─ ERR
                    └─ RES
```

ユーザーライブラリの名前は "SYS" で始めることはできません。

ディレクトリ構造は、Natural のインストール中に生成されます。システムおよびユーザーライブラリを表すディレクトリには、以下が含まれています。

■ **FILEDIR.SAG**

このファイルには、Natural が使用する内部ライブラリ情報が含まれています。詳細については、後述の「[ファイル FILEDIR.SAG](#)」を参照してください。

■ **SRC**

このサブディレクトリには、ライブラリに保存された Natural ソースオブジェクトが含まれています。

■ **GP**

このサブディレクトリには、ライブラリに保存された生成済み Natural プログラムが含まれています。

■ **ERR**

このサブディレクトリには、ライブラリに保存されたエラーメッセージが含まれています。

■ RES

このサブディレクトリには、ライブラリに保存されたプライベートリソースと共有リソースが含まれています。

DDM は、ローカルライブラリに保存できます。Natural は、プログラムによって使用される DDM を、現在のライブラリ、steplib、ライブラリ SYSTEM の順に検索します。DDM が検出されない場合は、プログラムはコンパイルされず、エラーメッセージが表示されます。ただし、**FDDM モード**がアクティブになっている場合は、Natural はシステムファイル FDDM のみで DDM を検索します。

システムファイル FNAT、FUSER および FDDM へのパスは、コンフィグレーションユーティリティで定義されます。システムファイルはバージョンに依存します。そのため、Natural は、現在の Natural バージョンのシステムファイルにのみアクセスできます。FNAT システムファイルは 1 つだけ使用することをお勧めします。ただし、複数の FUSER システムファイルを定義することは可能です（例えば、目的に応じて異なる開発エリアがある場合など）。

システムファイル FDDM

システムファイル FDDM は、すべての DDM を保存できるコンテナです。

FDDM では、次のディレクトリ構造が想定されます。

```
FDDM
├── SYSTEM
│   ├── FILEDIR.SAG
│   ├── SRC
│   └── GP
```

デフォルトでは、システムファイル FDDM はアクティブではありません。これを使用するには、次の手順に従って、FDDM モードをアクティブにする必要があります。

- [FDDM モードのアクティブ化](#)
- [システムファイル FDDM への DDM の移行](#)

■ システムファイル FDDM が使用されているかどうかの確認

FDDM モードのアクティブ化

FDDMモードがアクティブになっている場合（データベースIDとファイル番号の両方がグローバルコンフィグレーションファイルで0ではない場合）は、すべてのDDMはシステムファイル FDDM に保存され読み込まれます。ライブラリに格納された DDM は、Natural からアクセスできません。これは、すべての DDM がシステムファイル FDIC に保存されるメインフレームと同様です。

FDDMシステムファイルがグローバルコンフィグレーションファイルで定義されていない場合は、DDM は Natural ライブラリ FUSER および FNAT に保存され、FDDM システムファイルは非アクティブな環境として表示されます。

▶手順 4.1. FDDM モードをアクティブにするには

- 1 DDM が FDDM モードで保存される空のディレクトリを作成します。ディレクトリには、Natural 命名規則に従って任意の名前を付けることができます。
- 2 コンフィグレーションユーティリティを呼び出します。
- 3 グローバルコンフィグレーションファイル（ [システムファイル (System Files)] カテゴリ）で、システムファイル FDDM にデータベース ID とファイル番号を割り当て、最初の手順で作成したディレクトリへのパスを定義します。
- 4 必要なパラメータファイルを選択します。
- 5 パラメータ FDDM を見つけます。



Tip: このパラメータを見つけるには、"FDDM"で検索します。詳細については、『コンフィグレーションユーティリティ』ドキュメントの「パラメータの検索」を参照してください。

- 6 ドロップダウンリストボックスから、パラメータ FDDM に必要なパスを選択します。
- 7 変更内容を保存します。
- 8 以下の説明に従って、必要なすべての DDM をシステムファイル FDDM に移行します。

システムファイル FDDM への DDM の移行

FDDM モードで使用可能にするすべての DDM を、システムファイル FDDM に含める必要があります。特に、Natural で提供されたサンプル DDM (ライブラリ SYSEXDDM 内) は、システムファイル FDDM で使用可能である必要があります。

DDM を FDDM システムファイルに移行する方法は、次の 2 通りあります。

- オブジェクトハンドラを使用できます。オブジェクトハンドラは、FDDM システムファイルをサポートし、DDM を FDDM システムファイルに移行する機能を提供します。DDM を Natural ライブラリからアンロードし、アクティブな Natural セッションで FDDM システムファイルに保存できます。



Important: 完全な開発環境を移行するために、オブジェクトハンドラの使用をお勧めします。

- また、SYSMAIN のコピー機能や移動機能で DDM を移行したり、以下に記載されているように Natural スタジオのコピー／移動 (ドラッグ&ドロップ) 機能を使用したりすることも可能です。この場合は、古い環境を再使用できるように、FDDM パラメータを最初に非アクティブにする必要があります。

これらの方法については、以下で詳細に説明します。



Note: INPL ユーティリティは、DDM を、Natural ライブラリ (FDDM モードが非アクティブな場合) またはシステムファイル FDDM (FDDM モードがアクティブな場合) のいずれかにロードします。これは、ロードされる INPL ファイルが両方のモードで機能する場合にいくらか影響することがあります。DDM が Natural ライブラリと FDDM システムファイルの両方で使用可能である必要がある場合があります。

▶手順 4.2. オブジェクトハンドラを使用して DDM をシステムファイル FDDM に移行するには

- 1 前述したように、FDDM モードを有効にします。
- 2 修正したパラメータファイル (つまり、パラメータ FDDM のパスが定義されたパラメータファイル) を使用して、Natural スタジオを開始します。
- 3 [ツール] メニューの [開発ツール] > [オブジェクトハンドラ] を選択して、オブジェクトハンドラを開始します。
- 4 オブジェクトハンドラで [Options] メニューの [Settings] を選択します。
- 5 表示されたダイアログボックスで、[Additional Options] オプションボタンを選択し、[Set] ボタンをクリックします。
- 6 表示されたダイアログボックスで、[Special] ページを選択します。
- 7 [Use FDDM file for DDMs] チェックボックスをオフにします。

これにより、（移行する DDM が含まれる）古い環境が有効になります。このチェックボックスをオフにしないと、移行する DDM にアクセスできません。

- 8 Natural ライブラリに保存されている DDM をアンロードします（ウィザードを使用するか、上級ユーザーモードで行います）。
- 9 **[Use FDDM file for DDMs]** チェックボックスをオンにします（上記の手順を参照）。

これにより、FDDM システムファイルが含まれる新しい環境が有効になります。

 **Note:** ライブラリが異なれば、DDM の名前を同一にすることができます。FDDM システムファイルで DDM を上書きするのを防ぎ、同一の名前を持つ DDM を検出するために、**[Do not replace]** オプションを使用して DDM をロードすることをお勧めします。このオプションは、**[Use FDDM file for DDMs]** チェックボックスと同じページにあります。

- 10 DDM を FDDM システムファイルにロードします（ウィザードを使用するか、上級ユーザーモードで行います）。

▶手順 4.3. Natural スタジオのコピーまたは移動機能を使用して DDM をシステムファイル FDDM に移行するには

- 1 次に示すように、**ダイナミック**パラメータ `FDDM=0,0` を使用して、Natural を起動します。

```
natural FDDM=0,0
```

これにより、移行する DDM が含まれている古い環境が有効になります。修正したパラメータファイルで新しい FDDM 指定を上書きしないと、これらの DDM にアクセスできません。

- 2 必要なすべての DDM を Natural ライブラリから指定の FDDM ファイルのライブラリ SYSTEM にコピーまたは移動します。このファイルは、Natural スタジオの非アクティブな環境として表示されます。
- 3 Natural を終了します。

上記のダイナミックパラメータを使用せずに Natural を次回起動するときに、FDDM システムファイルが使用されます。

システムファイル FDDM が使用されているかどうかの確認

すべての DDM をシステムファイル FDDM に移行したら、FDDM が使用されているかどうかを確認します。

▶手順 4.4. FDDM が使用されているかどうかを確認するには

- 1 Natural を起動します。

- 2 [ツール] メニューの [システム情報] > [システムファイル] を選択します。『Natural スタジオの使用』ドキュメントの「システムファイル」も参照してください。
[SYSPROF] ダイアログボックスが表示されます。
- 3 FDDM ファイルが表示される場合は、Naturalはこのシステムファイルに保存されているDDMのみにアクセスします。
FDDM ファイルが表示されないか予期したファイルが表示されない場合は、セッションに使用されるパラメータファイルを修正してください。

重要な情報および警告

Natural 開発者は、すべてのオブジェクトに対する読み取り、書き込み、および削除の権限を持っている必要があります。

エンドユーザーは、生成プログラムに対する読み取り権限のみ（および特別な場合はソースに対する読み取り権限）を持っている必要があります。

オペレーティングシステムユーティリティで Natural ファイルにアクセスしないでください。これらのユーティリティは、Naturalディレクトリ情報を変更したり破壊したりする可能性があります。

FNAT、FUSER、および FDDM ディレクトリにプライベートデータファイルを保存しないでください。Natural によって予期せぬ方法で削除されたり変更されたりするおそれがあります。

FNAT、FUSER、および FDDM ディレクトリを Windows アプリケーション用の作業ディレクトリとして使用しないでください。Naturalシステムコマンドを発行するときに問題が発生するおそれがあります。

Natural によってアクセスされるオブジェクトのファイル名（8.3 フォーマットのファイル名を含むパス）は 255 バイト以下にする必要があります。

ファイル FILEDIR.SAG

ファイル FILEDIR.SAG は、最大 60000 オブジェクトをサポートします。このファイルには、オブジェクトのプログラミングモード（ストラクチャードまたはレポートング）や、内部的に変換されたオブジェクト名など、Natural で使用される内部ライブラリ情報が含まれます。Natural オブジェクトを次の名前前でディスクに保存するときに、これらの内部オブジェクト名が自動的に作成されます。

- 8 文字を超える名前（DDM の場合など）

- **Natural**ではサポートされるがオペレーティングシステムではサポートされない特殊文字を含む名前

内部オブジェクト名は一意であり、現在のオブジェクト名の省略形と任意の番号で構成されます。現在のオブジェクト名および対応する内部オブジェクト名は両方とも *FILEDIR.SAG* に記述されます。

オブジェクトが正しいディレクトリに配置されている場合でも、このライブラリ情報が *FILEDIR.SAG* に含まれた後のみ、**Natural** はそれを使用できます。**Natural** 内で作成されたオブジェクトについては、ライブラリ情報は自動的に含まれます。その他のオブジェクトをインポートする方法については、『*Natural* スタジオの使用』ドキュメントの「オブジェクトのインポート」セクションを参照してください。

FTOUCH ユーティリティを使用すると、**Natural** に入らずに *FILEDIR.SAG* を更新できます。

移植可能な **Natural** システムファイル

Natural バージョン 6.2 以降では、**Natural** ライブラリ内のディレクトリファイル *FILEDIR.SAG* と、**Natural** エラーメッセージファイルは、プラットフォームに依存しない移植可能なフォーマットで作成されます。これにより、例えば、オペレーティングシステムコマンドでライブラリをコピーするだけで、Windows、UNIX、および OpenVMS プラットフォーム間で FUSER ライブラリを交換できます。

FNAT システムファイルは **Natural** インストールに属し、バージョン固有であると同時にプラットフォーム固有でもあります。そのため、異なるプラットフォーム間で FNAT システムファイルを共有することはお勧めしません。特に、Windows プラットフォーム上の FNAT システムファイルには、まったく異なる一連のユーティリティが UNIX または OpenVMS プラットフォーム上の FNAT システムファイルとして含まれています。

異なるプラットフォーム間で FUSER システムファイルを共有できるようにはなりませんが、この機能の扱いには注意が必要です。**Natural** のロック機構はマシン境界を超えないため、異なるプラットフォーム上の 2 つの **Natural** セッションが同じオブジェクトを同時に変更して予期できない結果が生じる可能性があります。

Natural バージョン 6.2 以降で新しく作成されたすべてのライブラリには、新しい *FILEDIR.SAG* 構造が含まれています。特に、**Natural** バージョン 6.2 以降に提供およびインストールされた FNAT システムファイルには、新しい構造のライブラリしか含まれていません。

以下では次のトピックについて説明します。

- **言語依存オブジェクト**

■ 古いFILEDIR.SAG ファイルの移行

言語依存オブジェクト

移植するアプリケーションでシステム変数 *LANGUAGE が使用される場合は、以下の情報に注意する必要があります。

ほとんどすべての Natural オブジェクトは、大文字のみの名前のシステムファイルに保存されます。例外は、言語依存オブジェクト（特定の言語用に作成されたオブジェクト）です。言語依存オブジェクトの名前には、小文字が含まれている場合があります。Windows は大文字と小文字を維持するオペレーティングシステムであるため（UNIX は大文字と小文字を区別するオペレーティングシステム）、UNIX で作成された名前が Windows で競合を発生させたり、UNIX で開発されたアプリケーションが Windows で予期せぬ結果を生じさせたりすることがあります。



Note: OpenVMS の動作は Windows の動作と似ています。大文字と小文字は区別されません。ただし、ファイル名は常に大文字で作成されます。

例

コマンド `SAVE PGM&` は、オブジェクト名に言語識別子が含まれるオブジェクトを作成します。結果のオブジェクト名は、次のように *LANGUAGE の設定によって異なります。

| *LANGUAGE の設定 | 作成されるオブジェクトの名前 |
|---------------|----------------|
| 33 | PGMX (X が大文字) |
| 59 | PGMx (x が小文字) |

UNIX で作成された個別のオブジェクト (`PGMX.NGP` および `PGMx.NGP`) は、ファイル `FILEDIR.SAG` に `PGMX` および `PGMx` という名前でエントリを取得します。これらの2つのオブジェクトは、次のように Natural が実行される環境に応じて処理が異なります。

- Natural for UNIX で `PGMX` を実行すると、ファイル `PGMX.NGP` がバッファプールにロードされ実行されます。
- Natural for Windows で `PGMX` を実行すると、ファイル `PGMX.NGP` または `PGMx.NGP` がバッファプールにロードされ実行されます。これは、Windows がこれらの2つのオブジェクトを区別せずに1つの同じオブジェクトとして扱うためです。そのため、FUSER または FUSER のコピーを共有するアプリケーションの動作が異なる場合があります。

古いFILEDIR.SAG ファイルの移行

バージョン 6.2 以降の Natural では、プラットフォーム固有の古い FILEDIR.SAG ファイルをそれが以前に生成されたプラットフォームで読み取ることができますが、古い FILEDIR.SAG ファイルを変更することはできません。古い FILEDIR.SAG ファイルを含むライブラリが変更のためにアクセスされると、変更が行われる前に、FILEDIR.SAG は新しいフォーマットに変換されます。

 **Important:** 以下に挙げられた (FILEDIR.SAG が変換される) 手順を実行する前に、古い FUSER システムファイルのバックアップコピーを作成することをお勧めします。

次のように、古い FILEDIR.SAG ファイルが新しいフォーマットに変換される可能性は多数あります。

- バージョン 6.2 以降の Natural では、変更のためにアクセスされると、古い FILEDIR.SAG フォーマットが自動的に変換されます。この動作はユーザーには完全に透過的で、どのような場合でも強制されません。変更のためのアクセスとは、新しいソースの SAVE、ソースの CATALOG、CATALL などです。コピー操作では、コピー先ライブラリとコピー先ライブラリの FILEDIR.SAG ファイルが変更されます。移動操作では、オブジェクトがソースライブラリで削除される必要があるため、ソースライブラリが追加で変更されます。いずれの場合でも、元の FILEDIR.SAG ファイルはライブラリディレクトリに FILEDIR.BCK として保存されます。
- 変換されるすべてのライブラリは、オブジェクトハンドラ (SYSOBJH) でアンロードできます。結果のワークファイルが新しい FUSER システムファイルに再ロードされると、すべてのライブラリが新しい FILEDIR.SAG 構造で生成されます。
- ユーティリティ FTOUCH は、指定されたライブラリの FILEDIR.SAG ファイルを新しい構造に変換するオプション convert を提供します。元の FILEDIR.SAG ファイルは、ライブラリディレクトリに FILEDIR.BCK として残ります。構文および使用例については、FTOUCH ユーティリティの説明を参照してください。
- ユーティリティ SYSMAIN のコピー機能または Natural スタジオのドラッグ&ドロップ (コピーおよび貼り付け) 機能を使用して、ライブラリの完全なセットを古いフォーマットの FUSER システムファイルから新しい FUSER システムファイルにコピーできます。コピー先の FUSER では、FILEDIR.SAG ファイルが新しい構造で自動的に生成されます。

Natural バージョン 6.2 より前の Natural では、移植可能な新しい FILEDIR.SAG ファイルのライブラリにアクセスできません。そのため、Natural バージョン 6.2 以降とそれより前のバージョンでシステムファイルを共有することはできません。共有が可能なのは、すべてのライブラリが古い FILEDIR.SAG フォーマットのままで、変更のためのアクセスが行われたことがない場合に限られます。例えば、実稼働環境でライブラリに変更が加えられたことがない場合に、Natural バージョン 6.1 の FUSER を Natural バージョン 6.2 で使用できます。ただし、Natural バージョン 6.2 以降とそれより前のバージョンでシステムファイルを共有することはお勧めしません。

5 ワークファイル

- ワークファイルの定義 26
- ワークファイルフォーマット 29
- 拡張子 NCD を持つワークファイルについての特別な考慮 33

ワークファイルは、Naturalプログラムでデータを読み書きできるファイルです。これらのファイルは、データの間中ストレージやプログラム間のデータ交換に使用されます。データは、Naturalステートメント `READ WORK FILE` および `WRITE WORK FILE` を使用してワークファイルとの間で転送できます。

このchapterでは、次のトピックについて説明します。

ワークファイルの定義

コンフィグレーションユーティリティまたは `DEFINE WORK FILE` ステートメントを使用して、最大 32 のワークファイルに名前（パスを含む）を割り当てることができます。

使用できるワークファイルの最大数は、パラメータ `WORK` の設定によって異なります。

名前およびパスが割り当てられていないワークファイルを使用するプログラムを実行すると、Naturalはファイル名を自動的に作成し、ローカルコンフィグレーションファイルに指定された一時ディレクトリにワークファイルを書き込みます。このファイル名は、指定されたワークファイル番号と、オペレーティングシステムによって割り当てられた任意の番号で構成されます。ワークファイル名の生成は、一意の名前を生成するアルゴリズムに基づいています。Naturalパラメータ `TMPSORTUNIQ` に応じて、命名規則は異なります。ワークファイル名がNatural以外から参照される場合は、ファイルの識別上の問題を回避するために、名前を明示的に指定することをお勧めします。

以下では次のトピックについて説明します。

- [コンフィグレーションユーティリティを使用したワークファイル名の定義](#)
- [環境変数を使用したワークファイル名の定義](#)
- [アプリケーションプログラミングインターフェイスを使用したワークファイル名の定義](#)

コンフィグレーションユーティリティを使用したワークファイル名の定義

コンフィグレーションユーティリティでは、ワークファイル名はパラメータファイルの [ワークファイル設定 (**Work Files**)] カテゴリで割り当てられます。上記のパラメータ `WORK` および `TMPSORTUNIQ` もこのカテゴリにあります。詳細については、『コンフィグレーションユーティリティ』ドキュメントの「ワークファイルの割り当て」を参照してください。



Tip: ワークファイルの割り当てを確認するには、"ワークファイル (Work Files)" で検索します。詳細については、『コンフィグレーションユーティリティ』ドキュメントの「パラメータの検索」を参照してください。

環境変数を使用したワークファイル名の定義

以下では次のトピックについて説明します。

- 全般的な情報
- 環境変数のデリミタ
- ファイル名のドル記号 (\$)

全般的な情報

ワークファイルは、Windowsの環境変数を使用して定義することもできます。パラメータファイルにワークファイル名を定義すると、パラメータファイルをそれ以上変更せずにワークファイル名を設定できます。例えば、パラメータファイル（または DEFINE WORK FILE ステートメント）にワークファイルの名前を次のように指定します。

```
%Natural%\%myfile%
```

オペレーティングシステムに次のように設定されているとします。

```
set Natural=D:\natural
set myfile=sub\test
```

この場合は、次のファイル名に展開されます。

```
D:\natural\sub\test
```

環境変数のデリミタ

環境変数名は特殊文字で区切られます。左デリミタは変数の左側に、右デリミタは変数の右側にあります。

例えば、文字列 %TEMP% の場合は、TEMP という名前の環境変数を識別し、% が左右両方のデリミタとして使用されています。

有効なデリミタは次のとおりです。

| デリミタのタイプ | 有効なデリミタ |
|----------|------------------|
| 左デリミタ | % \$ |
| 右デリミタ | % / . \ |

 **Note:** 文字列の末尾の印はデフォルトでは右デリミタです。%TEMP は TEMP という名前の環境変数であると認識されます。

Windows では "%" のみが環境変数に有効な左デリミタですが、Natural for Windows では以前のバージョンとの上位互換性を維持するために左デリミタとして "%" と "\$" が許可されます。この設定により、Windows セッションでの UNIX 系ワークファイル名割り当てが可能です。\$TEMP は、Natural for Windows と Natural for UNIX の両方で環境変数 TEMP として認識されます。

例：

以下の Natural コード行は同じであると解釈されます。

```
DEFINE WORK FILE 1 '$TEMP\myfile.dat'
```

および

```
DEFINE WORK FILE 1 '%TEMP%\myfile.dat'
```

TEMP は環境変数として認識されます。文字列 \$TEMP (または%TEMP%) は、ランタイム時に環境変数 TEMP の内容に置き換えられます。

ファイル名のドル記号 (\$)

ファイル名のドル記号 (\$) には、2つの意味があります。

- "\$" は、デリミタに埋め込まれた文字列の左または中央にある場合は、使用されている環境変数の左デリミタとして解釈されます。左デリミタの後ろから右デリミタまたは EOS までのすべての文字は、環境変数の名前であるとみなされます。
- "\$" は、文字列の最後の文字である場合は、デリミタ文字とはみなされません。スキャンされる文字列の一部です。

例：

次の Natural コード行は、エラーにはなりません。

```
DEFINE WORK FILE 1 '\\MYPC\C$\myfile.dat'
```

\\MYPC\C\$ はデフォルトの共有とみなされます。"C\$" は有効なディレクトリです。

ただし、次のNaturalコード行は、Aが定義されているかどうかによってエラーになる可能性があります。

```
DEFINE WORK FILE 1 '\\MYPC\C$A\myfile.dat'
```

"A" は前にドル記号があるために環境変数として解釈されます。"A" が定義されていない場合は、エラーが発生します。"A" が定義されている場合は、エラーは発生しません。

アプリケーションプログラミングインターフェイスを使用したワークファイル名の定義

ライブラリ SYSEXT のアプリケーションプログラミングインターフェイス USR1050N を使用してワークファイルを定義することもできます。

ワークファイルフォーマット

ワークファイルのフォーマットは、定義されているワークファイルタイプによって異なります。さまざまなワークファイルフォーマットを使用できます。Naturalでは、ファイル名と拡張子をチェックすることによってフォーマットが認識されます。

```
file-name.extension
```

file-name は最大 8 文字、*extension* は最大 3 文字です。

ワークファイルフォーマットは以下のとおりです。

- バイナリフォーマット
- ASCII フォーマット
- Entire Connection フォーマット
- Portable フォーマット
- Unformatted フォーマット
- CSV フォーマット

『Unicode およびコードページのサポート』ドキュメントの「Windows、UNIX、およびOpenVMSプラットフォームでのワークファイルおよび出力ファイル」も参照してください。

バイナリフォーマット

使用可能なタイプ：SAG

このフォーマットは、Software AGに固有です。すべてのデータタイプに使用できるため、推奨されるフォーマットです。

書き込まれた各レコードの前には、レコードの長さを含める2バイトが付加されます。

ワークファイル用にバイナリフォーマットを定義するには、ピリオドと拡張子"SAG"を含むファイル名（例：<file-name>.SAG）またはピリオドを含まないファイル名（例：<file-name>）のいずれかを使用します。

ASCII フォーマット

使用可能なタイプ：ASCII および圧縮 ASCII

書き込まれた各レコードは改行（CR/LF）で終了されるため、ASCIIフォーマットは英数字データにのみ推奨されます。

ワークファイル用に ASCII フォーマットを定義するには、ピリオドと任意の拡張子（"SAG" と "NCD"を除く）を含むファイル名（例：<file-name>.<ext>）またはピリオドを含み拡張子を含まないファイル名（例：<file-name>）のいずれかを入力します。

Entire Connection フォーマット

使用可能なタイプ：Entire Connection

Entire Connection 製品では、データファイル（実際のデータを含む）とフォーマットファイル（データファイルのデータに関するフォーマット情報を含む）の2つのファイルを使用します。

Natural は、タイプ Entire Connection 用の対応するフォーマットファイルを自動的に生成します。フォーマットファイルは、名前はデータファイルと同じですが、拡張子は "NCF" です。拡張子 "NCF" を持つフォーマットファイルの内容の詳細については、『Entire Connection』ドキュメントを参照してください。

ワークファイル用に Entire Connection フォーマットを定義するには、ピリオドと拡張子 "NCD" を含むファイル名（例：<file-name>.NCD）を入力します。

Entire Connection フォーマットのワークファイルはローカルディスクとの間で直接読み書きできます。

「[拡張子 NCD を持つワークファイルについての特別な考慮](#)」も参照してください。



Notes:

1. READ WORK FILE ステートメントの RECORD オプションは、フォーマット Entire Connection のワークファイルを読み込むには使用できません。
2. ワークファイルタイプ Entire Connection では、オペラントフォーマット U (Unicode) はサポートされていません。このワークファイルタイプで U が使用された場合は、ランタイムエラーメッセージが表示されます。

Portable フォーマット

使用可能なタイプ：Portable

タイプ Portable は、ワークファイルが別のマシンに転送されるときに、ワークファイルの自動エンディアン変換を実行します。例えば、PC (リトルエンディアン) で書き込まれたワークファイルは、RS6000 または HP マシン (ビッグエンディアン) で正しく読み取ることができます。エンディアン変換は、フィールドフォーマット I2、I4、F4、F8、および U にのみ適用されます。浮動小数点フォーマットは、IEEE であるとみなされます。ただし、ハードウェアシステムによって、IEEE 浮動小数点表現にはわずかな違いがあります。ルールとして、これらの違いは、ワークファイルに通常書き込まれない無限と NaN 表現にのみ適用されます。詳細については、ハードウェア記述を確認してください。

異なる表記のマシンによってファイルが読まれる場合にのみ変換が実行されるように、ファイルは常にマシン固有の表記で記述されます。これにより、速いパフォーマンスが保たれます。

上記の変換以外には、このフォーマットにはその他の変換はありません。

READ WORK FILE ステートメントがダイナミック変数に使用される場合は、変数は現在のレコード長にサイズ変更されます。

Unformatted フォーマット

使用可能なタイプ：Unformatted

タイプ Unformatted は、1つのダイナミック変数と1つのレコードを持つ完全なファイルを読み書きします (例えば、データベースから読み込んだビデオを保存するため)。フォーマット情報は挿入されず、すべてそのまま読み書きされます。

CSV フォーマット

使用可能なタイプ：CSV (コンマ区切りの値)

 **Note:** ワークファイルタイプ CSV を使用する場合は、CATALOG または STOW コマンドを使用してソースを再カタログする必要があります。ワークファイルタイプ CSV を Natural バージョン 4 の生成プログラムで使用することはできません。

Natural フィールドは、以下のように CSV ワークファイルに保存されます。

1. 最初の手順では、内部フィールドデータが可読フォーマットに変換されます。

- 内部 Natural データフォーマット B (バイナリ)、O (オブジェクトハンドル)、G (GUI ハンドル)、および C (属性制御) のフィールドデータは、フィールド変換なしでレコードにコピーされます。データがそのまま取得されます。
- 内部 Natural データフォーマット A (英数字) のフィールドデータは、指定されたワークファイルコードページ (『コンフィグレーションユーティリティ』ドキュメントの「ワークファイル」を参照) に変換されます。コンフィグレーションユーティリティでワークファイルコードページが指定されていない場合は、パラメータ CP で定義されたデフォルトのコードページが使用され、変換は行われません。

内部 Natural データフォーマット U (Unicode) のフィールドデータは、指定されたワークファイルコードページ (『コンフィグレーションユーティリティ』ドキュメントの「ワークファイル」を参照) に変換されるか、ワークファイルコードページが指定されていない場合はパラメータ CP で定義されたデフォルトのコードページに変換されます。

- 内部 Natural フォーマット D (日付) および T (時刻) の値は、英数字出力フォーマットに変換されます。DTFORMパラメータは、ユーザー指定の日付および時刻フォーマットが使用されるように評価されます。
- 数値タイプの内部フィールド値は、英数字出力フォーマットに変換されます。

2. 次の手順では、可読フォーマットのフィールドデータが CSV ワークファイルレコードにコピーされます。ワークファイル内のフィールドは、指定されたセパレータ文字で区切られています。フィールドに特殊文字が含まれている場合は、フィールドは二重引用符で区切られます。書き込まれた各レコードは、改行 (CR/LF) で終了されます。

Natural フィールド名を含むヘッダーがワークファイルに書き込まれるように定義している場合は (『コンフィグレーションユーティリティ』ドキュメントの「ワークファイルの割り当て」を参照)、以下が適用されます。

- WRITE WORK FILE ステートメントを使用すると、最初に書き込まれたレコードのフィールド名を含むヘッダー行がワークファイルの最初の行に保存されます。後続の CSV レコードに異なる数のフィールドが含まれている場合は、ヘッダー行がそれらの後続の CSV レコードに対応しないことがあります。
- READ WORK FILE ステートメントを使用すると、CSV ワークファイルの最初の行がヘッダー行であるとみなされます。そのため、最初の行がスキップされます (つまり、最初の行にあるレコードデータが返されません)。

拡張子 NCD を持つワークファイルについての特別な考慮

拡張子 "NCD" を持つファイルが Entire Connection で作成されて READ WORK FILE ステートメントで Natural に読み込まれる場合は、セッションプロパティで Entire Connection オプションの [Keep trailing blanks] がアクティブになっている必要があります。詳細については、『Entire Connection』ドキュメントを参照してください。

 **Note:** Entire Connection を使用して NCD ファイルを作成し、オブジェクトハンドラを使用してこのファイルをロードすると、ソース制御レコードが見つからないことを示すエラーが表示されることがあります。これを回避するには、NCD ファイルを作成するときに [Keep trailing blanks] オプションがアクティブになっていることを確認します。

Entire Connection フォーマットのワークファイルについては以下の点を考慮してください。

- READ WORK FILE ステートメントで NCD ファイルを読み込むときに、対応する NCF フォーマットファイルが使用できないか無効な情報を含んでいる場合は、NCD ファイルは ASCII ワークファイルとみなされます。
- データを NCD ファイルに追加するために APPEND 属性が使用されている場合は、新旧データのレコードレイアウト（NCF フォーマットファイルに書き込まれるフィールドフォーマットおよび長さの情報）が一致する必要があります。レコードレイアウトが異なる場合は、ランタイム中にエラーが発生します。
- Entire Connection で処理できる WRITE WORK FILE VARIABLE の最大ワークファイルレコードサイズは、32767 バイトです。
- 拡張子が "NCD" の「古い」ワークファイルがある場合は、拡張子を変更する必要があります。
- 以下の各プロファイルパラメータは、読み書き両方の操作に対して同じ値に設定する必要があります。
 - DC (小数点文字)
 - IA (INPUT 割り当て文字)
 - ID (INPUT 区切り文字)
- メインフレームコンピュータでの浮動小数点変数に有効な値の範囲は、他のプラットフォームと異なることに注意してください。メインフレームの F4 および F8 変数に有効な値の範囲は次のとおりです。

$$\pm 5.4 * 10^{-79} \text{ to } \pm 7.2 * 10^{75}$$

他のほとんどのプラットフォームで F4 変数に有効な値の範囲は次のとおりです。

$$\pm 1.17 * 10^{-38} \text{ to } \pm 3.40 * 10^{38}$$

他のほとんどのプラットフォームで F8 変数に有効な値の範囲は次のとおりです。

$$\pm 2.22 * 10^{-308} \text{ to } \pm 1.79 * 10^{308}$$

6 Natural バッファプール

| | |
|--|----|
| ▪ 全般的な情報 | 36 |
| ▪ バッファプールの設定 | 40 |
| ▪ Natural バッファプールサービスの使用 | 40 |
| ▪ ユーティリティ NATBPSRV を使用したバッファプールの作成 | 44 |
| ▪ バッファプールのモニタリング | 44 |
| ▪ トラブルシューティング | 45 |

このchapterでは、次のトピックについて説明します。

全般的な情報

Natural バッファプールは、同じコンピュータ上で Natural オブジェクトにアクセスする複数の Natural プロセス間で Natural オブジェクトを共有するのに使用されます。バッファプールとは、コンパイルされた Natural プログラムを実行に備えて配置するストレージエリアのことです。Natural ユーザーからの Natural オブジェクト要求に応じて、プログラムはバッファプールにまたはバッファプールから移動されます。

Natural ではリエントラントな Natural オブジェクトコードが生成されるので、Natural プログラムの1つのコピーを複数のユーザーが同時に実行できます。この目的のため、各オブジェクトは、呼び出し元が呼び出すたびにロードされるのではなく、システムファイルから Natural バッファプールに1回だけロードされます。

以下では次のトピックについて説明します。

- バッファプール内のオブジェクト
- リソース処理
- 複数のバッファプール
- バッファプールへのオブジェクトの保存
- 読み取り専用バッファプール
- 制限

バッファプール内のオブジェクト

バッファプール内のオブジェクトは、プログラムやダイアログなどの実行可能なオブジェクトです。実行可能なオブジェクトのうち、ローカルデータエリア、パラメータデータエリア、およびコピーコードは、コンパイル目的でのみバッファプールに配置されます。

Natural オブジェクトがバッファプールにロードされると、ディレクトリエントリと呼ばれるコントロールブロックがこのオブジェクトに割り当てられます。このコントロールブロックには、オブジェクトの名前、オブジェクトが属するライブラリまたはアプリケーション、オブジェクトの取得元のデータベース ID および Natural システムファイル番号などの情報と、いくつかの統計情報（例えば、プログラムを同時に実行しているユーザーの数）が含まれます。

リソース処理

リソースは、リソースが Natural システムファイル（FUSER など）のライブラリにあり、名前がファイル拡張子も含めて 32 文字以内の場合に、バッファプールにロードされます。

環境変数 NATGUI_BMP に割り当てられたディレクトリにあるか名前が 32 文字を超えるリソースは、アクセスされるたびに Natural プロセスに直接ロードされます（つまり、リソースはバッファプールにロードされません）。

複数のバッファプール

個別の要件に応じて、同じコンピュータ上で同じ Natural バージョンの異なるバッファプールを同時に実行できます。

コンフィグレーションユーティリティで、各バッファプールに対して同期を有効にできます（後述の「[バッファプールの設定](#)」も参照）。同じシステムファイルからのオブジェクトを含み、同期が有効にされているすべてのバッファプールは、自動的に同期されます。

 **Important:** システムファイルが共有ドライブにある場合は、同期はサーバー上のファイルシステムが NTFS の場合にのみ実行されます。

同期が有効にされている場合：複数のバッファプールにロードされたオブジェクトが 1 つの Natural プロセスで変更されると、まずそのオブジェクトは無効としてマークされます。オブジェクトがプロセスで使用されなくなると、バッファプールから削除されます。次回このオブジェクトがプロセスで要求されると、オブジェクトはバッファプールに再度ロードされます。

バッファプールへのオブジェクトの保存

ユーザーがプログラムを実行すると、バッファプールマネージャが呼び出されます。プログラムがすでにバッファプールロードされているかどうかを確認するためにディレクトリエントリが検索されます。まだロードされていない場合は、コピーが適切なライブラリから取得され、バッファプールにロードされます。

Natural オブジェクトをバッファプールにロードしているときに、新しいディレクトリエントリがこのプログラムを識別するために定義され、新しくロードするオブジェクトを収容するために、現在実行中ではない 1 つまたは複数の他の Natural オブジェクトがバッファプールから削除されることがあります。

この目的のために、バッファプールは現在どのユーザーがどのオブジェクトを使用しているかの記録を管理し、それにより、ユーザーがそのオブジェクトを解放しないで Natural を終了する状況を検出します。他のアプリケーションに属している新しいオブジェクトを収容するために、未使用または期限切れのオブジェクトがダイナミックに削除されます。

読み取り専用バッファプール

読み取り専用バッファプールとは、読み取りアクセスのみを許可する特別なバッファプールです。オブジェクトが読み取り専用のバッファプールにない場合、Naturalはエラー82（オブジェクトが見つからない）を発行します。システムファイルで見つからないオブジェクトの取得が試行されないため、システムファイルおよびバッファプールに対するすべてのロック操作がスキップされます。アカウントデータは収集されません。読み取り専用バッファプールにアクセスできるユーザー数に制限はありません。

読み取り専用バッファプールは、コンフィグレーションユーティリティで定義されます（後述の「[バッファプールの設定](#)」も参照）。バッファプールが読み取り専用バッファプールとして定義されている場合は、ユーザー最大数に定義されている値は無視されます。

ユーティリティ `NATBPSRV` は、セマフォを読み取り専用バッファプールに割り当てません。ただし、Natural パラメータファイルの位置に `<バッファプール名>.PRL` ファイル内のプリロードリストが必要です。これは、ローカルコンフィグレーションファイルに定義されます（インストール指定）。例えば、読み取り専用バッファプールの名前が "ROBP" である場合は、ファイル名は `ROBP.PRL` になります。

プリロードリストは、ライブラリ `SYSBPM` にある Natural プログラム `CRTPRL`（プリロードリストリストの作成）を使用して生成できます。このプログラムは、バッファプールの内容を抽出してバッファプールの既存のプリロードデータとマージします。

`PRL` ファイル内のプリロードリストには、コンマ区切りデータのレコードが次の形式で含まれています。

```
database-ID, file-number, library, object-name, kind, type
```

ファイル内のキーワードは、`NATBPMON` ユーティリティの `DIR` コマンドで表示されるキーワードと同じ意味を持ちます。

ディレクトリ記述レコード（オブジェクトの種類は、オブジェクトが `FILEDIR.SAG` の一部であることを意味する "D"）を除き、すべてのキーワードに値が割り当てられている必要があります。例：

| キーワード | NATBPSRV によってバッファプールにロードされる内容 |
|-------------------------|---|
| 222,111,MY_LIB,PGM1,G,P | データベース 222 およびファイル番号 111 にあるライブラリ MY_LIB のプログラム PGM1 のオブジェクトコード |
| 222,113,*,*,D | FNAT = 222,113 にある <code>LIBDIR.SAG</code> |
| 222,111,MY_LIB,*,D | FUSER = 222,111 にあるライブラリ MY_LIB の <code>FILEDIR.SAG</code> |

読み取り専用バッファプールの使用には、アプリケーションの詳細が認識されていなければならないという短所があります（不明オブジェクトはロードできないため）。これは、アプリケーションで必要なすべてのオブジェクトをプリロードリストに指定する必要があることを意味しま

す。まれに、アプリケーションで必要なオブジェクトの完全なセットを事前に決定できることがあります。

セカンダリ読み取り／書き込みバッファプール

Naturalでは、読み取り専用バッファプールをプライマリバッファプールとして実行できます。このようなバッファプールは変更できません。読み取り専用バッファプールにないオブジェクトはロードできません。オブジェクトが読み取り専用のバッファプールにない場合、Naturalはエラー 82 (オブジェクトが見つからない) を発行します。これを回避するには、Natural を実行時にセカンダリ標準バッファプール (読み取り／書き込みアクセス可能) に接続し、そこで不明オブジェクトをアクティブにします。ダイナミックパラメータ BPID2 は、セカンダリバッファプールを識別します。

読み取り専用バッファプールの場合を除き、セカンダリバッファプールに接続できるユーザー数には上限があり、セカンダリバッファプールがアクセスされるたびにセマフォを使用したオブジェクトのロックが実行されます。

読み取り専用バッファプールのプリロードリストは、ライブラリ SYSBPMにあるプログラム CRTPRL を使用して、セカンダリ読み取り／書き込みバッファプールの内容を読み取り専用バッファプールのプリロードリストとマージすることによって更新／拡張できます。

代替の読み取り専用バッファプール

読み取り専用バッファプールには、コンフィグレーションユーティリティで代替バッファプールの名前を定義できます (後述の「[バッファプールの設定](#)」も参照)。

読み取り専用バッファプールにのみ使用できる、NATBPMONユーティリティのSWAPコマンドを使用して、読み取り専用バッファプールに「現在使用されていない」というタグを付けることができます。現在使用されていないバッファプールに接続されたすべてのNaturalセッションは、このバッファプールから切断され、代替バッファプールに接続されます。ただしこれは、代替バッファプールも読み取り専用バッファプールの場合に限られます。Naturalが新しいオブジェクトのロードを試行するとき (例えば、CALLNAT または RETURN ステートメントの実行時)、またはNaturalがスタックに配置されたコマンドの解釈を試行するときに、1つのバッファプールから別のバッファプールへのスワップが実行されます。現在使用されていないというタグが付けられているバッファプールのIPCリソース (共有メモリセグメント) は、NATBPMONユーティリティのSWAPコマンドを発行した後に削除できます。この機能により、Naturalセッションを中止せずに、バッファおおよびその内容を別の読み取り専用バッファプールの更新された内容と交換できます。

制限

Natural バッファプールの使用時に考慮する必要がある最低限の制限は以下のとおりです。

- Natural セッションがハングアップしたときに、Windows タスクマネージャを使用して終了しないでください。

このセッションでバッファプール内部データ構造を現在変更している場合は、更新が完全に終了しない段階で割り込みが発生することがあります。バッファプール内部データ構造に矛盾があると、問題が発生することがあります。



Note: これは、Natural ニュークリアスがバッファプールルーチンを実行しているときにのみ発生します。

バッファプールの設定

バッファプール割り当ては、ローカルコンフィグレーションファイルに保存されています。バッファプールを設定するには、コンフィグレーションユーティリティを使用して、ローカルコンフィグレーションファイルに特定の値を指定する必要があります。これらの値のリストについては、『コンフィグレーションユーティリティ』ドキュメントの「バッファプール割り当て」を参照してください。

Natural バッファプールサービスの使用

Natural は、PC の起動時に Windows サービスの **Software AG Natural *n.n* Buffer Pool Service** を使用してバッファプールサーバーを起動します。

Natural のインストール時に、デフォルトバッファプール NATBP が設定されます。NATBP は、Natural 起動時のデフォルトバッファプール名としても使用されます（別のバッファプールを定義するには、プロファイルパラメータ BPID を使用します）。

必要に応じて、サービス設定を変更できます。これについては、以降のトピックを参照してください。

- [バッファプールサービスコマンド](#)

■ 例：独自のバッファプールを使用した Natural の開始

バッファプールサービスコマンド

Natural *bin* ディレクトリに保存されている *natbpsvc.exe* ファイルは、サービスコマンドを実行するのに使用されます。

次のサービスコマンドは、Windows のコマンドプロンプトウィンドウで指定できます。

| コマンド | 説明 | | | | |
|--|---|--------|--|-----------|---------------------------------|
| NATBPSVC INSTALL <i>mode</i> | <p>バッファプールサービスをインストールします。 <i>mode</i> は、次のいずれかです。</p> <table border="1"> <tr> <td>manual</td> <td>デフォルトです。 サービスがインストールされ、手動で開始する必要があります (START コマンドを使用するか、Windows で Software AG Natural n.n Buffer Pool Service を開始することによって行います)。</td> </tr> <tr> <td>automatic</td> <td>サービスがインストールされ、PCの起動時に自動的に起動します。</td> </tr> </table> | manual | デフォルトです。 サービスがインストールされ、手動で開始する必要があります (START コマンドを使用するか、Windows で Software AG Natural n.n Buffer Pool Service を開始することによって行います)。 | automatic | サービスがインストールされ、PCの起動時に自動的に起動します。 |
| manual | デフォルトです。 サービスがインストールされ、手動で開始する必要があります (START コマンドを使用するか、Windows で Software AG Natural n.n Buffer Pool Service を開始することによって行います)。 | | | | |
| automatic | サービスがインストールされ、PCの起動時に自動的に起動します。 | | | | |
| NATBPSVC CREATE <i>buffer-pool-name</i> | サービスによって開始されるバッファプールを新規作成します。 サービスは、指定された名前を持つバッファプールが Natural パラメータファイルに定義されているかどうかをチェックします。 | | | | |
| NATBPSVC START | サービス (停止している場合)、および <i>start</i> パラメータが "yes" に設定されている (SET コマンドを参照) すべての作成されたバッファプール (CREATE コマンドを参照) を開始します。 | | | | |
| NATBPSVC START <i>buffer-pool-name</i> | 指定バッファプールを開始します。 サービスが (PCの起動時に自動的に、またはユーザーによって手動で) 開始されていなかった場合は、エラーメッセージが表示されます。 | | | | |
| NATBPSVC SET <i>buffer-pool-name</i> start= <i>mode</i> | <p>サービスの開始時に指定バッファプールを開始するかどうかを定義します。 <i>mode</i> は、次のいずれかです。</p> <table border="1"> <tr> <td>yes</td> <td>バッファプールが開始されます。</td> </tr> <tr> <td>no</td> <td>デフォルトです。 バッファプールは開始されません。</td> </tr> </table> | yes | バッファプールが開始されます。 | no | デフォルトです。 バッファプールは開始されません。 |
| yes | バッファプールが開始されます。 | | | | |
| no | デフォルトです。 バッファプールは開始されません。 | | | | |
| NATBPSVC STOP | サービスを停止し、以前に開始されたすべてのバッファプールを停止します。 | | | | |
| NATBPSVC STOP <i>buffer-pool-name</i> | 指定バッファプールを停止します。 | | | | |
| NATBPSVC SHOW | サービス用に定義されたすべてのバッファプールの設定パラメータを表示して、サービスの開始時にこれらのバッファプールを開始するかどうかを示します。 | | | | |
| NATBPSVC SHOW <i>buffer-pool-name</i> | 指定バッファプールの設定パラメータを表示して、サービスの開始時にこのバッファプールを開始するかどうかを示します。 | | | | |

| コマンド | 説明 |
|--|---|
| NATBPSVC STATUS | サービス用に定義されたすべてのバッファプールのステータスを表示して、これらのバッファプールがアクティブか非アクティブかを示します。 |
| NATBPSVC STATUS <i>buffer-pool-name</i> | 指定バッファプールのステータスを表示して、このバッファプールがアクティブか非アクティブかを示します。 |
| NATBPSVC DELETE <i>buffer-pool-name</i> | 指定バッファプールをサービスから削除します。 注意: デフォルトのバッファプール NATBP は削除しないでください。削除すると、Natural が正常に機能しなくなる可能性があります。 |
| NATBPSVC REMOVE | サービスをシステムから削除します。 |

例：独自のバッファプールを使用した Natural の開始

この例では、MYBP という名前の新しいバッファプールを作成し、新しいバッファプールで Natural を開始する方法について説明します。

▶手順 6.1. 独自のバッファプールで Natural を開始するには

- 1 コンフィグレーションユーティリティを使用して、ローカルコンフィグレーションファイルに追加のバッファプールを MYBP という名前で定義します。『コンフィグレーションユーティリティ』ドキュメントの「バッファプール割り当て」を参照してください。

 **Note:** この例では、デフォルトバッファプール NATBP に定義された値と同じ値を使用できます。

- 2 Windows のコマンドプロンプトウィンドウを起動します。
- 3 *natbpsvc.exe* ファイルが保存されている Natural *bin* ディレクトリに移動します。
- 4 次のコマンドを入力して、バッファプールを MYBP という名前で作成します。

```
NATBPSVC CREATE MYBP
```

次の情報が表示されます。

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: New buffer pool 'MYBP' created
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

 **Important:** 指定した名前のバッファプールがローカルコンフィグレーションファイルにまだ定義されていない場合は、対応するメッセージが代わりに表示されます。次の手順に進む前に、ローカルコンフィグレーションファイルにバッファプールを定義してください。

- 5 次のコマンドを入力して、サービスの開始時にバッファプールが開始されるように定義します。

```
NATBPSVC SET MYBP start=yes
```

次の情報が表示されます。

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: Configuration successfully set
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

- 6 次のコマンドを入力して、バッファプールを直ちに開始します（サービスを再スタートする必要はありません）。

```
NATBPSVC START MYBP
```

次の情報が表示されます。

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: Send request to Natural n.n Buffer Pool Service
%NATBPSVC-I: Buffer pool 'MYBP' started
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

- 7 次のコマンドを入力して、サービス用に現在定義されているすべてのバッファプールのステータスを表示します。

```
NATBPSVC STATUS
```

次の情報が表示されます。

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: Send request to Natural n.n Buffer Pool Service
%NATBPSVC-I:   MYBP is active
                NATBP is active
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

- 8 次のようにダイナミックパラメータ BPID を使用して、Natural を開始します。

```
natural BPID=MYBP
```

ユーティリティ NATBPSRV を使用したバッファプールの作成

バッファプールの作成には、ユーティリティ NATBPSRV を使用します。

バッファプールサーバーは、**Natural バッファプールサービス**によって自動的に起動されます。



Note: ユーティリティ NATBPSRV を使用すると、他のバッファプールユーザーの作業を破壊する可能性があるため、すべてのNaturalユーザーがこのユーティリティにアクセスできるようにはしないでください。

NATBPSRV は、バッファプールが必要とするリソースを割り当て、バッファプール用の恒久的な通信機能（共有メモリとセマフォ）を作成します。リソースと機能に必要な指定は、コンフィグレーションユーティリティで行います（「[バッファプールの設定](#)」を参照）。

デフォルトでは、バッファプール NATBP が開始されます。他のバッファプールを開始するには、次の NATBPSRV コマンド行オプションで名前を指定します。

```
NATBPSRV BP = buffer-pool-name
```

バッファプールのモニタリング

バッファプールモニタでは、バッファプールの操作中の動作を監視します。バッファプールモニタは、コンピュータ上でNaturalを停止しなければならないときにバッファプールをシャットダウンするのにも使用できます。

バッファプールモニタでは、Natural バッファプールの現在の状態に関する情報が収集されます。

同じコンピュータ上で複数のバッファプールがアクティブになっていて、複数のバッファプールにロードされたオブジェクトがNatural プロセスで変更されると、オブジェクトはそのNatural プロセスが接続先されているバッファプールからのみ削除されます。変更されたオブジェクトが、それが現在ロードされている同じコンピュータ上の他のバッファプールからも削除されるようにするには、コンフィグレーションユーティリティでバッファプールの同期を有効にします。

Natural では、2つのバージョンのバッファプールモニタを使用できます。グラフィカルユーザーインターフェイスと、コマンド行バージョンの NATBPMON ユーティリティです。

バッファプールモニタの異なるバージョンの使用方法については、「[GUIバージョンのバッファプールモニタの使用](#)」および「[コマンド行バージョンのバッファプールモニタの使用 \(NATBPMON\)](#)」を参照してください。

トラブルシューティング

このセクションでは、Natural バッファプールの使用時に発生する可能性のある問題とその解決方法について説明します。

以下は、典型的なコマンド出力の例と、実行中に発生した問題に関する説明です。

問題

Natural または Natural バッファプールモニタを起動できない。

例

以降の例では、Natural 管理者またはユーザーが経験する可能性がある最も一般的な問題について説明します。これらの問題は、Natural または NATBPMON ユーティリティを起動するときにバッファプールが非アクティブな場合に発生します。

- Natural を起動しようとする、次のメッセージが表示されます。

```
Natural Startup Error 16: Unable to open buffer pool.  
Buffer pool error: "unexpected system call error occurred" (20)  
Buffer pool could not attach to the global shared memory.
```

- Natural バッファプールモニタを起動しようとする、次のメッセージが表示されます。

```
Cannot get shared memory  
Buffer pool error: "unexpected system call error occurred" (20)  
Buffer pool could not attach to the global shared memory.
```

解決策

「[Natural バッファプールサービスの使用](#)」の説明に従って、バッファプールサービスを開始します。

7 GUIバージョンのバッファプールモニタの使用

| | |
|------------------------------------|----|
| ■ バッファプールモニタの開始と終了 | 48 |
| ■ Natural バッファプールモニタウィンドウの要素 | 49 |
| ■ バッファプールの切断と接続 | 52 |
| ■ バッファプールサーバーのシャットダウン | 53 |
| ■ バッファプールサーバーの開始 | 54 |
| ■ バッファプールモニタのプロパティの変更 | 55 |
| ■ 一般情報 | 56 |
| ■ バッファプールコンテンツ | 58 |
| ■ グラフィックアナライザ | 61 |
| ■ レポート | 66 |

このchapterでは、次のトピックについて説明します。

バッファプールの一般的な情報およびバッファプールの起動方法について記載されている「[Natural バッファプール](#)」も参照してください。

 **Caution:** このユーティリティを使用すると、バッファプールの他のユーザーの作業を破損する可能性があるため、Naturalのすべての一般ユーザーがこのユーティリティにアクセスできるようにはしないでください。

バッファプールモニタの開始と終了

Natural フォルダは、Natural のインストール後に [スタート] メニューの [すべてのプログラム] フォルダに自動的に表示されます。これには、バッファプールモニタを含む、Natural のショートカットが含まれます。

▶手順 7.1. バッファプールモニタを開始するには

- Windows の [スタート] メニューの [すべてのプログラム] > [Software AG Natural *n.n*] > [Buffer Pool Monitor] を選択します。

[Natural バッファプールモニタ] ウィンドウが表示されます。



Note: バッファプールは、Natural *bin* ディレクトリに保存されている実行ファイル *natbpmong.exe* を使用して開始することもできます。

▶手順 7.2. バッファプールモニタを終了するには

- [モニタ] メニューの [終了] を選択します。

Or:

Alt キーを押したまま F4 キーを押します。

Or:

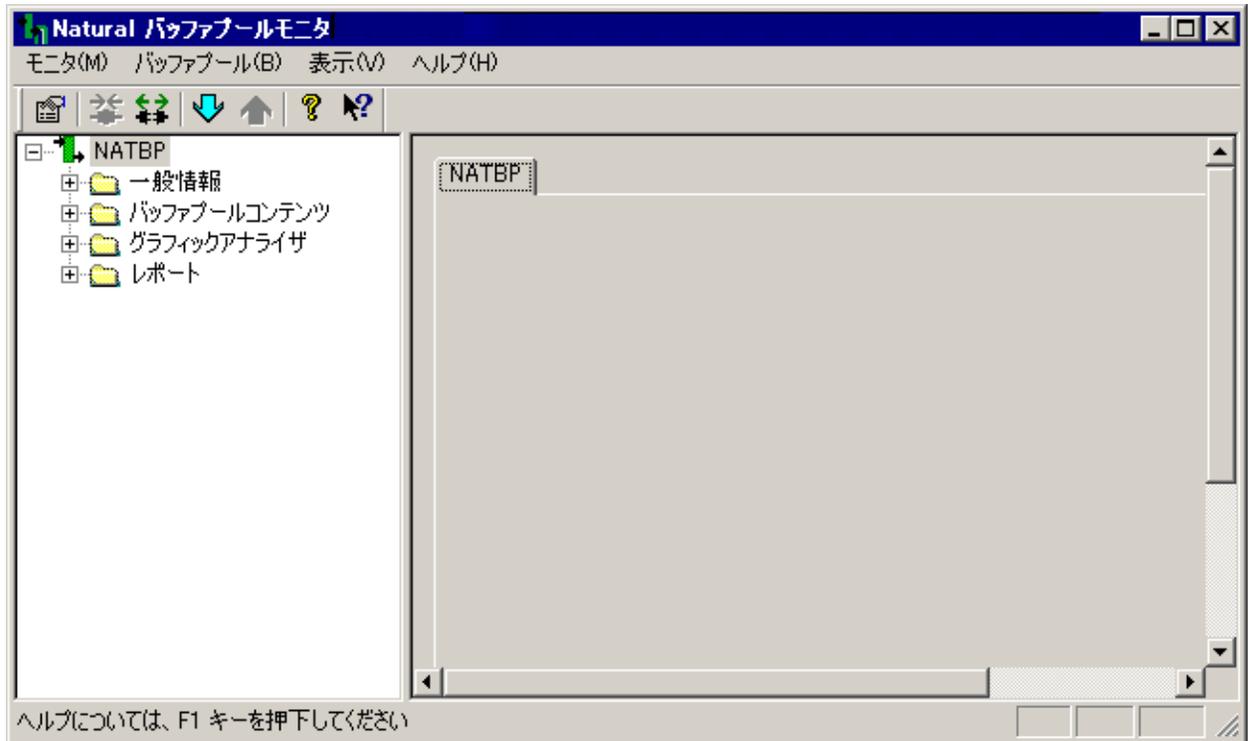
コントロールメニューの [閉じる] を選択します。

Or:

タイトルバーの右側にある対応する標準ボタンを選択します。

Natural バッファプールモニタウィンドウの要素

バッファプールモニタを開始すると、Naturalのデフォルトバッファプール NATBP に自動的に接続します。現在使用されているバッファプールの名前がタイトルバーに表示されます。これはツリーでも選択されています。



以下では次のトピックについて説明します。

- メニューバー
- ツールバー
- ツリー

■ ステータスバー

メニューバー

次のメニューを使用できます。

| メニュー | このメニューのコマンドで実行できる操作 |
|---------|--|
| モニタ | プロパティを変更するか、バッファプールモニタを終了します。 |
| バッファプール | バッファプールを切断および接続します。バッファプールサーバーをシャットダウンおよび起動します。 |
| 表示 | [Natural バッファプールモニタ] ウィンドウのさまざまな要素を表示または非表示にします。 |
| ヘルプ | オンラインドキュメントを呼び出し、バッファプールモニタに関する情報を表示します。 |

ツールバー

重要な機能はツールバーボタンを使用して実行できます。マウスポインタをツールバーの上に移動させると、ステータスバーにそのボタンの簡単な説明が表示されます。

次のツールバーボタンを使用できます。

-  プロパティの変更
-  他のバッファプールへ接続
-  現在のバッファプールを切断
-  バッファプールサーバーのシャットダウン
-  バッファプールサーバーの起動
-  オンラインヘルプの表示

▶手順 7.3. ツールバーの表示のオンとオフを切り替えるには

- [表示] メニューの [ツールバー] を選択します。

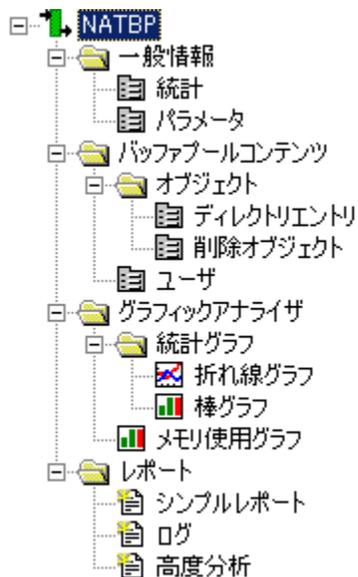
ツールバーが [Natural バッファプールモニタ] ウィンドウに表示されると、このメニューコマンドの横にチェックマークが付きます。

ツリー

〔**Natural** バッファプールモニタ〕ウィンドウの左側にあるツリーには、ローカルコンフィグレーションファイルに現在割り当てられているすべてのバッファプールが表示されます。『コンフィグレーションユーティリティ』ドキュメントの「バッファプール割り当て」を参照してください。

一度にモニタリングできるのは1つのバッファプールのみです。他のバッファプールに接続するには、「[バッファプールの切断と接続](#)」を参照してください。

現在使用されているバッファプール（以下の例の**NATBP**）のすべてのノードが展開されている場合は、ツリーは次のように見えます。



ツリーでノードを選択すると、ウィンドウの右側に対応するページが表示されます。各ページの詳細については、以降のセクションを参照してください。

- [一般情報](#)
- [バッファプールコンテンツ](#)
- [グラフィックアナライザ](#)
- [レポート](#)

ステータスバー

ステータスバーとは、[Natural バッファプールモニタ] ウィンドウの下部にある横方向の情報行のことです。ステータスバーには、メニューバーとツールバーのコマンドについての短いヘルプテキストが表示されます。

▶ **手順 7.4. ステータスバーの表示/非表示を切り替えるには、次の手順に従います。**

- [表示] メニューの [ステータスバー] を選択します。

ステータスバーが [Natural バッファプールモニタ] ウィンドウに表示されると、このメニューコマンドの横にチェックマークが付きます。

バッファプールの切断と接続

一度に接続できるのは1つのバッファプールのみです。環境内で別のバッファプールに切り替えるには、現在使用しているバッファプールを切断してから新しいバッファプールに接続します。



Note: 別のバッファプールに接続すると、前に接続されていたバッファプールは自動的に切断されます。そのため、**Disconnect** コマンドを最初に使用する必要があります。

バッファプール名の横のアイコンは、次のいずれかの状態を示します。



バッファプールモニタはバッファプールに接続されています (緑のアイコン)。



バッファプールモニタはバッファプールに接続されていません (グレーのアイコン)。

▶ **手順 7.5. 現在使用しているバッファプールを切断するには**

- 1 現在接続されているバッファプールの名前をツリーで選択してください。
- 2 [モニタ] メニューの [切断] を選択します。

Or:

コンテキストメニューを表示し、[切断] を選択します。

Or:

次のツールバーボタンを選択します。



このバッファプールのツリーは表示されなくなります。

▶手順 7.6. バッファプールを接続するには

- 1 バッファプールの名前をツリーで選択してください。
- 2 [モニタ] メニューの [接続] を選択します。

Or:

コンテキストメニューを表示し、 [接続] を選択します。

Or:

次のツールバーボタンを選択します。



このバッファプールのツリーが表示されます。

バッファプールサーバーのシャットダウン

接続しているバッファプールをシャットダウンできます。これは、バッファプールがバッファプールサービスによって開始されなかった場合に限り可能です。

例えば、バッファプールを初期化する必要がある場合は、バッファプールをシャットダウンしてから再起動します。

まだ接続されている Natural プロセスがある間は、バッファプールサーバーはシャットダウンされません。最後のプロセスがバッファプールから切断された後にのみシャットダウンされます。プロセスが接続されている間は、バッファプールのステータスは「シャットダウン中」になります。これは、ツリーのバッファプール名の横に表示されます。

▶手順 7.7. バッファプールサーバーをシャットダウンするには

- 1 現在接続されているバッファプールの名前をツリーで選択してください。
- 2 [モニタ] メニューの [サーバのシャットダウン] を選択します。

Or:

コンテキストメニューを表示し、[サーバのシャットダウン] を選択します。

Or:

次のツールバーボタンを選択します。



このバッファプールのツリーは表示されなくなります。

バッファプールサーバーの開始

バッファプールサーバーがまだ開始されていない場合は、バッファプールサーバーを開始できるのはバッファプールモニタからのみです。

▶手順 7.8. バッファプールサーバーを開始するには

- 1 バッファプールの名前をツリーで選択してください。
- 2 [モニタ] メニューの [サーバの起動] を選択します。

Or:

コンテキストメニューを表示し、[サーバの起動] を選択します。

Or:

次のツールバーボタンを選択します。



バッファプールサーバーが開始されます。

この場合は、バッファプールは自動的に接続されません。「[バッファプールの切断と接続](#)」の説明に従って、バッファプールを手動で接続する必要があります。

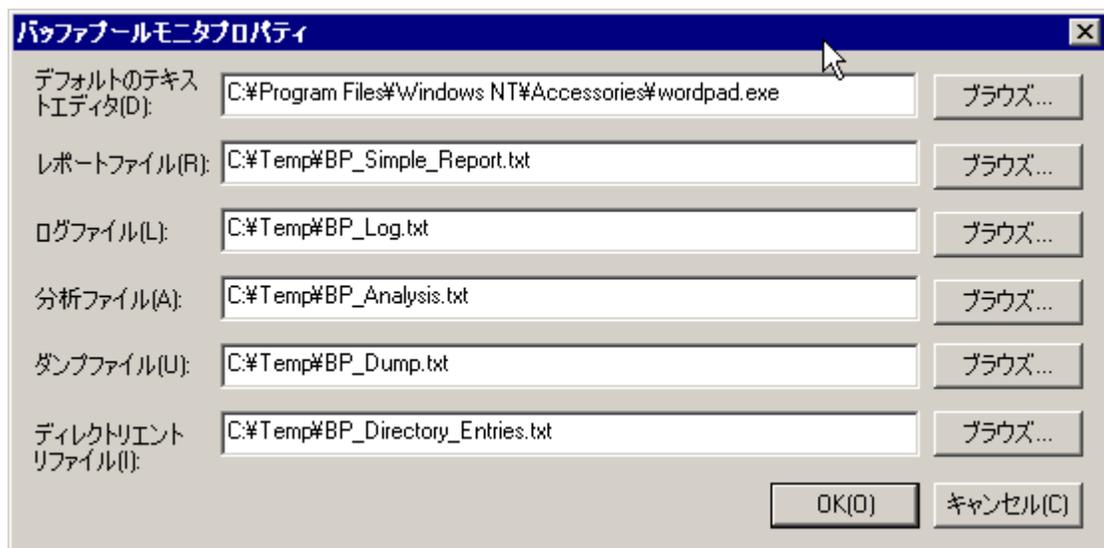
バッファプールモニタのプロパティの変更

バッファプールモニタのいくつかのページにデフォルトとして表示されるファイルを定義できます。また、使用されるデフォルトのテキストエディタも定義できます。

手順 7.9. プロパティを変更するには

- 1 [モニタ] メニューの [プロパティ] を選択します。

次のダイアログボックスが表示されます。デフォルトでは、一時ディレクトリが現在のユーザー用に定義されています。例えば、ユーザー独自の一時ディレクトリが定義されている場合は、ダイアログボックスに次のように表示されます。



次の情報を変更できます。

- 以下のページでテキストファイルを開くときに使用されるデフォルトのテキストエディタ
- [シンプルレポート] ページで使用されるレポートファイル
- [ログ] ページで使用されるログファイル
- [高度分析] ページで使用される分析ファイル
- [高度分析] ページで使用されるダンプファイル
- [ディレクトリエントリ] ページで使用されるディレクトリエントリファイル

- 2 エントリを変更するには、該当するテキストボックスにパスとファイル名を指定します。

Or:

対応する [ブラウズ] ボタンをクリックして、ダイアログボックスからファイルを選択します。

3 [OK] ボタンを選択します。

一般情報

ツリーで [一般情報] ノードを展開すると、バッファプールとそのパラメータの統計データを表示できます。

次のページを使用できます。

- [統計](#)
- [パラメータ](#)

統計

ツリーで [統計] を選択すると、次のページが表示されます。

| 統計 | | パラメータ | |
|---------------------|-------------------------|---|-------------------------|
| 一般情報 | | | |
| BPID: | NATBP | 開始日時: | 14-AUG-2007 7:22:31 |
| シャットダウン状態: | active | 最終クリア日時: | 14-AUG-2007 7:22:31 |
| メモリ割り当て | | ユーザ統計 | |
| 割り当てメモリ: | 1,737,980 bytes 1.66 MB | 現在のユーザ数: | 1 |
| 最小割り当てメモリ: | 20 bytes 0.02 KB | 最大ユーザ数: | 3 |
| 最大割り当てメモ: | 145,404 bytes 142.00 KB | 削除された停止ユーザ | 0 |
| 空きメモリサイズ: | 1,407,760 bytes 1.34 MB | オブジェクトの使用統計 | |
| 最小連続メモリサイズ: | 92 bytes 0.09 KB | 未使用オブジェクト数: | 213 |
| 最大連続メモリサイズ: | 848,496 bytes 828.61 KB | アクティブなオブジェクト数: | 0 |
| オブジェクトのロード統計 | | 生成オブジェクト数: | 0 |
| 格納されたオブジェクト数: | 0 | 削除予定オブジェクト数: | 0 |
| ロードされたオブジェクト数: | 905 | オブジェクトサイズ統計 | |
| 起動されたオブジェクト数: | 9548 | 最大オブジェクトサイズ: | 40,257 bytes 39.31 KB |
| 中断されたロード数: | 317 | 最小オブジェクトサイズ: | 108 bytes 0.11 KB |
| ロケット統計 | | 合計オブジェクトサイズ: | 1,564,515 bytes 1.49 MB |
| ロケット回数: | 11378 | 一般的なロード統計 | |
| 高速ロケットの試行回: | 4811 | 未使用オブジェクトの削除回数: | 0 |
| 高速ロケットの成功回: | 3458 | 並列起動の最大数: | 1 |
| 成功パーセント: | 71.876949 | オブジェクトの再起動回数の割合: | 10.550276 |
| [カウンタのクリア(C)] | | [更新(R)] <input checked="" type="checkbox"/> 自動更新(S) | |

このページには、バッファプールの全般的な情報と、メモリ、ユーザー、およびオブジェクトの詳細情報が表示されます。これは、NATBPMON ユーティリティの STATUS コマンドを使用した場合と同じ情報です。詳細については、「[バッファプールの統計情報](#)」を参照してください。

[自動更新] チェックボックスをオンにした場合は、ページは自動的に更新されます。このチェックボックスをオフにした場合は、[更新] ボタンをクリックして値を手動で更新する必要があります。

[カウンタのクリア] ボタンをクリックすると、バッファプールの内部統計が0にリセットされます。

パラメータ

ツリーで [パラメータ] を選択すると、次のページが表示されます。



| | |
|-------------|-------------------------|
| 統計 | パラメータ |
| バッファプール ID: | NATBP |
| 共有メモリー: | GNAT632BPMEM_0x16321111 |
| セマフォキー: | GNAT632BPSEM_0x16321111 |
| メモリーサイズ: | 3,145,728 bytes 3.00 MB |
| 最大ユーザ数: | 20 |

このページには、NATBPMON ユーティリティの PARAM コマンドを使用した場合と同じ情報が表示されます。「[バッファプール設定の表示](#)」を参照してください。

バッファプールコンテンツ

ツリーで [バッファプールコンテンツ] ノードを展開すると、バッファプールにロードされた Natural オブジェクトおよびそれらにアクセス中のユーザーに関する詳細を表示できます。

次のページを使用できます。

- [ディレクトリエントリ](#)
- [削除オブジェクト](#)
- [ユーザー](#)

ディレクトリエントリ

ツリーで [オブジェクト] ノードを展開し、[ディレクトリエントリ] を選択すると、次のページが表示されます。



このページには、現在ロードされているすべてのディレクトリエントリに関する情報の表が表示されます。これは、NATBPMON ユーティリティの DIR コマンドを使用した場合と同じ情報です。「[バッファプールのオブジェクトの表示](#)」を参照してください。

次のコマンドボタンを使用できます。

| コマンドボタン | 説明 |
|-------------------|--|
| ディレクトリエントリの更新 | 表を更新します。 |
| すべてのディレクトリエントリの削除 | バッファプールに現在ロードされているすべてのNaturalオブジェクトを削除します。 |

表の上にマウスがあるときは、コンテキストメニューを表示して次のコマンドを使用できます。

| コンテキストメニューのコマンド | 説明 |
|-----------------|---------------------|
| すべて選択 | 表内のすべてのエントリを選択します。 |
| 削除 | 表内の選択されたエントリを削除します。 |

フィルタオプション

フィルタを使用して、表に表示されるディレクトリエントリ数を減らすことができます。

▶手順 7.10. フィルタを定義するには

- 1 [フィルタの使用] チェックボックスをオンにします。
- 2 テキストボックス [DBID]、[FNR]、[ライブラリ]、[名前]、[種類]、および [タイプ] にフィルタ基準を指定します。

例えば、"MY" で始まるライブラリのプログラムをすべて表示するには、[ライブラリ] テキストボックスに「MY*」と入力し、[タイプ] テキストボックスに「P」と入力します。

- 3 [ディレクトリエントリの更新] ボタンをクリックして、表を更新します。

ファイルの書き込み

表に現在表示されているすべてのディレクトリエントリをファイルに書き込むことができます。必要に応じて、ディレクトリエントリのメモリもこのファイルに書き込むことができます。

▶手順 7.11. ディレクトリエントリをファイルに書き込むには

- 1 オプション。[ファイル名] テキストボックスに、ディレクトリエントリの書き込み先ファイルへのパスを指定します。

Or:

このテキストボックスの右にあるボタンをクリックして、ダイアログボックスからファイルを選択します。



Note: デフォルトで、[ファイル名] テキストボックスにはプロパティで定義したファイルへのパスが指定されています。

- 2 オプション。ディレクトリエントリのメモリをこのファイルに書き込む場合は、[ディレクトリエントリメモリの書き込み] チェックボックスをオンにします。
- 3 [ファイルの書き込み] ボタンをクリックします。

情報が指定のファイルに書き込まれます。このファイルの内容は、プロパティで定義されたテキストエディタに自動的に表示されます。

削除オブジェクト

ツリーで [オブジェクト] ノードを展開し、[削除オブジェクト] を選択すると、次のページが表示されます。

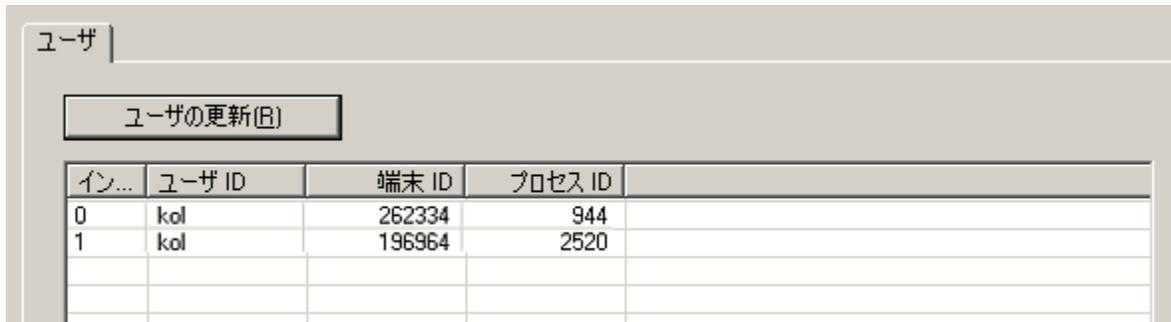
| Nr | DBID | FNR | ライブラリ (L) | 名前 (N) | 種類... | タイ... | ユ... | 最大ユ... | 使用数 | 生成 | サイズ [...] | GP バー... |
|----|------|-----|-----------|--------|-------|-------|------|--------|-----|----|-----------|----------|
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |

削除オブジェクトとは、バッファプールから削除されているがまだ使用中のオブジェクトのことです。削除オブジェクトがある場合は、表にそれらが表示されます。

[削除されるオブジェクトの更新] ボタンをクリックすると、表を更新できます。

ユーザー

ツリーで [ユーザ] を選択すると、次のページが表示されます。



| イン... | ユーザ ID | 端末 ID | プロセス ID |
|-------|--------|--------|---------|
| 0 | kol | 262334 | 944 |
| 1 | kol | 196964 | 2520 |
| | | | |
| | | | |

このページには、バッファプールを現在使用しているユーザーに関する情報の表が表示されません。

[ユーザーの更新] ボタンをクリックすると、表を更新できます。

グラフィックアナライザ

ツリーで [グラフィックアナライザ] ノードを展開すると、バッファプールメモリ内で発生している内容の統計的な数字とダイレクトビューをグラフ表示できます。

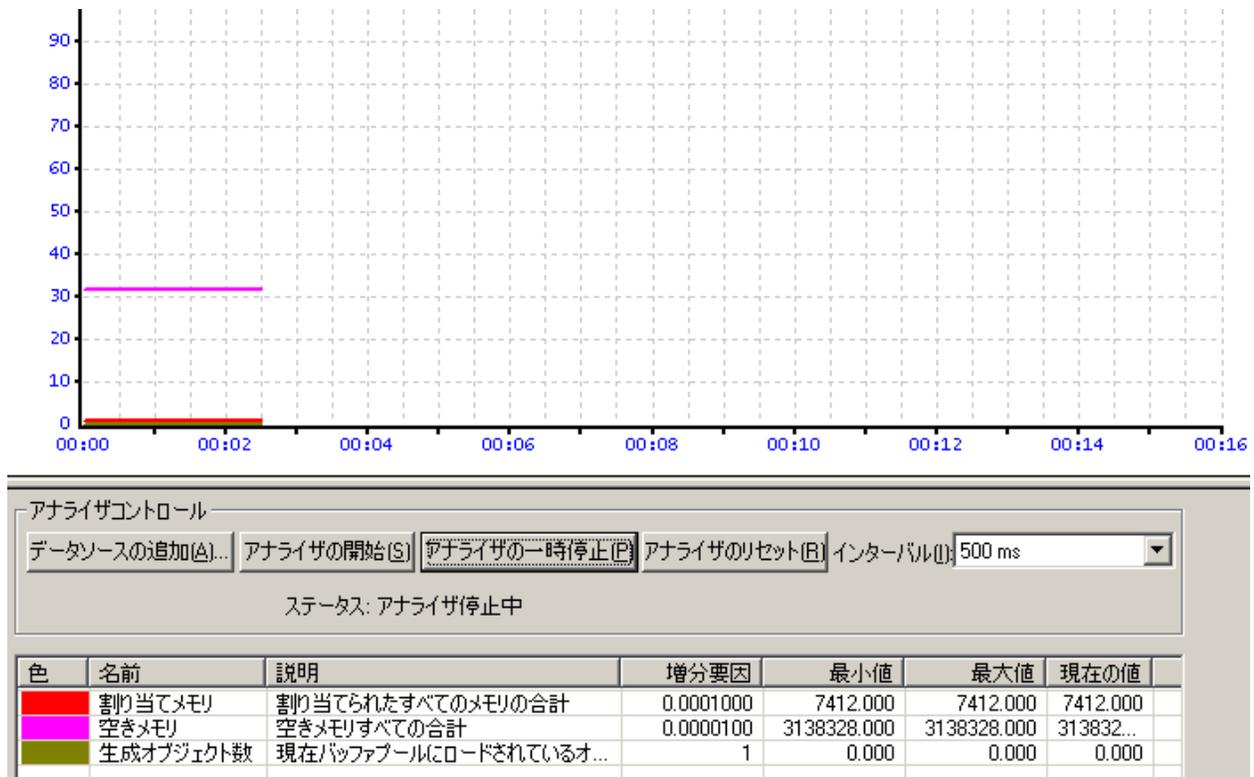
次のページを使用できます。

- 折れ線グラフ
- 棒グラフ

■ メモリ使用グラフ

折れ線グラフ

ツリーで [統計グラフ] ノードを展開し、 [折れ線グラフ] を選択すると、次のページが表示されます。 データソースを追加してアナライザを開始すると、このページは次のように表示されます。



折れ線グラフと棒グラフは、両方とも同じ統計データソースに基づいています。 次のいずれかの処理を折れ線グラフに適用すると、この処理は棒グラフにも適用されます。その逆も同様です。

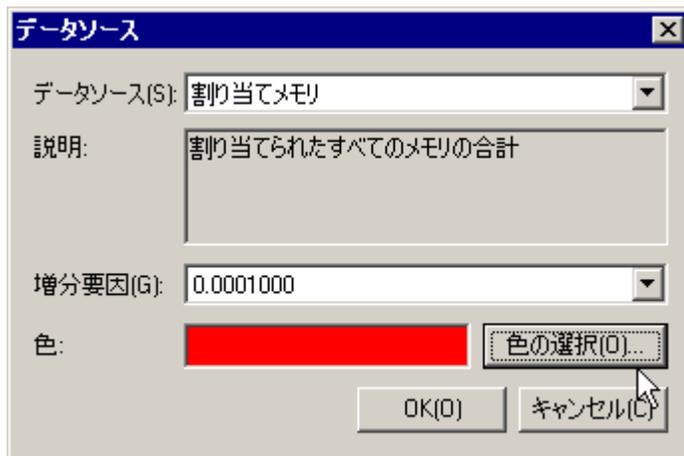
- データソースの追加、変更、または削除
- アナライザの開始、一時停止、またはリセット
- 更新間隔の調整

 **Note:** アナライザは、 [メモリ使用グラフ] ページでも使用されます。

▶手順 7.12. データソースを追加するには

- 1 [データソースの追加] ボタンをクリックします。

[データソース] ダイアログボックスが表示されます。



- 2 [データソース] ドロップダウンリストボックスから、グラフに表示するデータソースを選択します。

選択したデータソースの説明が表示されます。

- 3 オプション。 [増分要因] ドロップダウンリストボックスから、選択したデータソースに必要な値を選択します。

これにより、y 軸の範囲が調整されます。 グラフに適切な表現を完成するために、データソースの現在値がこの要因で乗算されます。

- 4 オプション。 選択したデータソースに別の色を定義する場合は、 [色の選択] ボタンをクリックします。

Windows の標準の [色] ダイアログボックスが表示され、データソースに使用する別の色を選択または定義できます。

- 5 [OK] ボタンをクリックして、ページの下部に表示される表にデータソースを追加します。これで、データソースを折れ線グラフと棒グラフの両方で使用できるようになります。

表に、追加した各データソースの色、名前、説明、および増分要因が表示されます。また、データソースの最小値、最大値、および現在値も含まれています。

- 6 オプション。 他のデータソースを表に追加する場合は、上記の手順を繰り返します。

▶手順 7.13. 定義したデータソースの管理

- 1 表でデータソースを選択し、コンテキストメニューを表示します。

コンテキストメニューには、次のコマンドがあります。

| コンテキストメニューのコマンド | 説明 |
|-----------------|--|
| プロパティ | 選択したデータソースの [データソース] ダイアログボックスを表示します。この場合は、ダイアログボックスは別の色を定義するためにしか使用できません。 |
| すべて選択 | 表内のすべてのデータソースを選択します。 |
| 削除 | 表内の選択されたデータソースを削除します。 |

2 上記のコマンドのいずれかを選択します。

▶ **手順 7.14. 更新間隔を調整するには**

- [インターバル] ドロップダウンリストボックスから、更新時間を選択します（ミリ秒単位、秒単位、または分単位で更新するための異なる値が表示されます）。

グラフの更新間隔が調整されます。

 **Note:** [インターバル] ドロップダウンリストボックスは、アナライザが非アクティブの場合にのみ使用できます。

▶ **手順 7.15. アナライザを開始するには**

- 必要なすべてのデータソースを表に追加したら、[アナライザの開始] ボタンをクリックします。

すべてのグラフでアナライザが開始されます。選択したデータソースが、折れ線グラフと棒グラフでグラフ表示されます。

▶ **手順 7.16. アナライザを一時停止するには**

- [アナライザの一時停止] ボタンをクリックします。

すべてのグラフでグラフ表示が停止します。

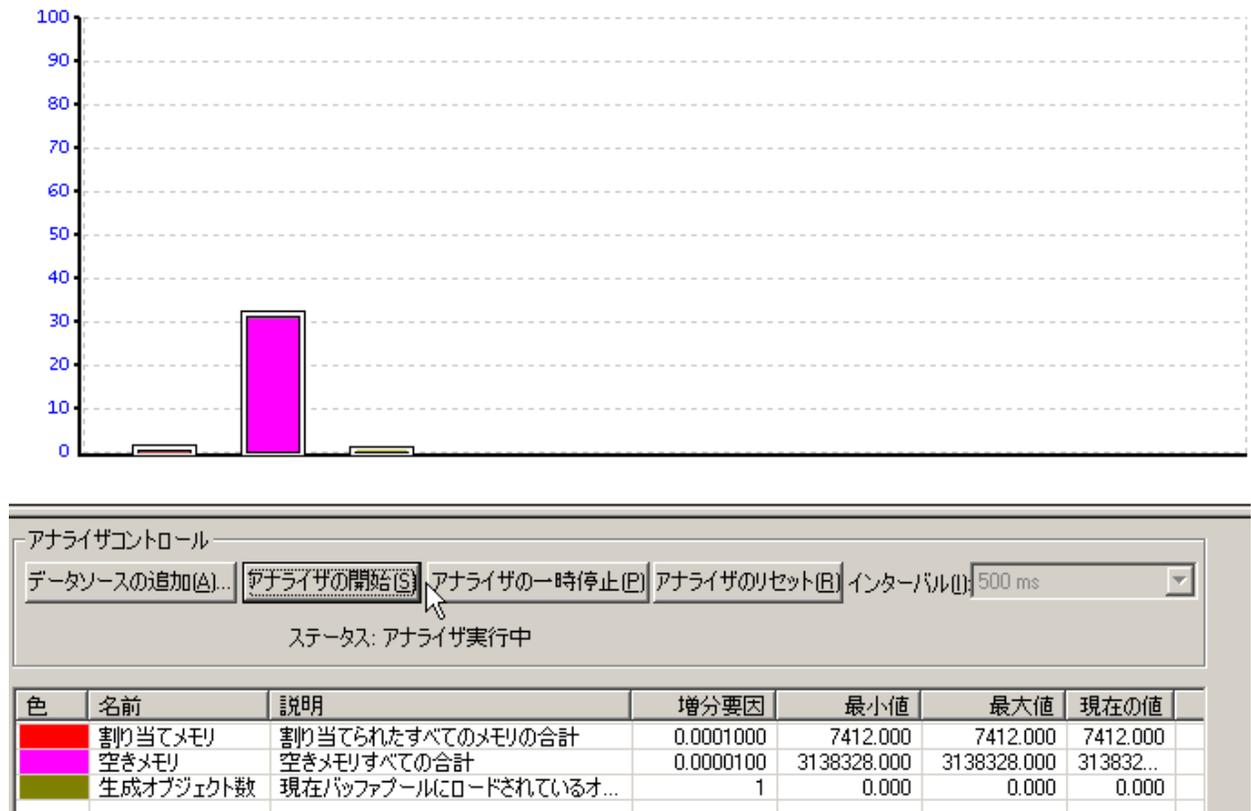
▶ **手順 7.17. アナライザをリセットするには**

- [アナライザのリセット] ボタンをクリックします。

すべてのグラフでグラフ表示がリセットされます。折れ線グラフと棒グラフで、表の最小値、最大値、および現在値がリセットされます。折れ線グラフに表示される時間ベースもリセットされます。

棒グラフ

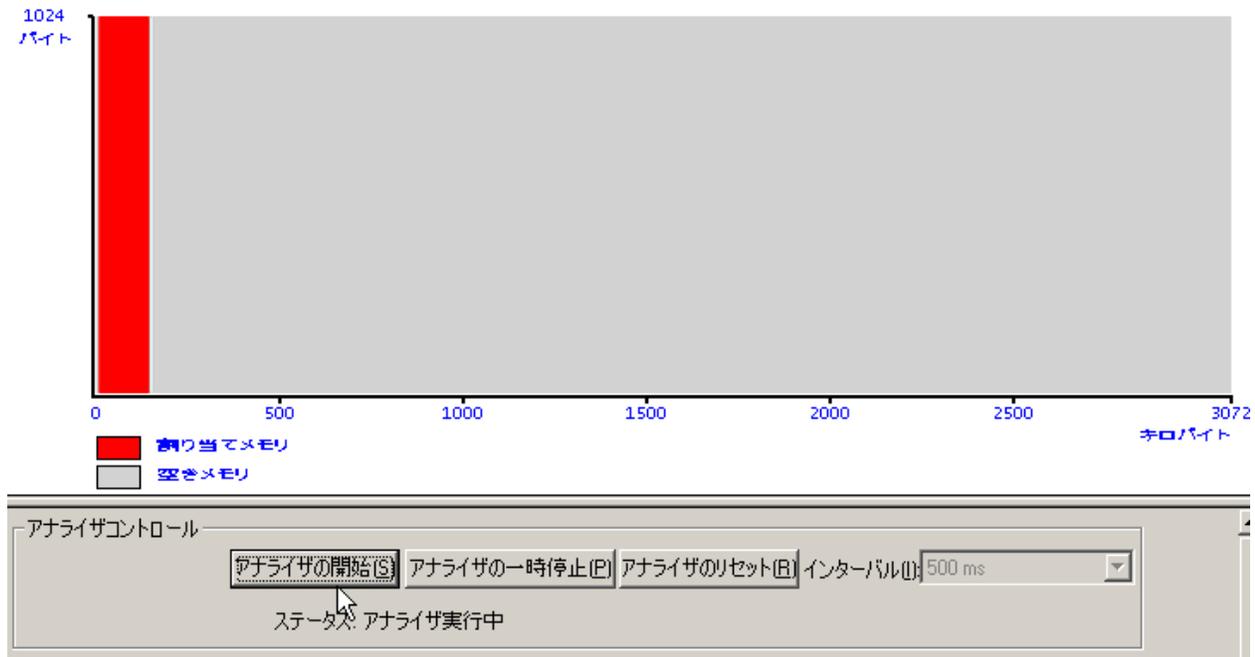
ツリーで「統計グラフ」ノードを展開し、「棒グラフ」を選択すると、次のページが表示されます。データソースを追加してアナライザを開始すると、このページは次のように表示されます。



折れ線グラフに処理を適用すると、この処理は棒グラフにも適用されます。その逆も同様です。データソースを追加、変更、または削除する方法、アナライザを開始、一時停止、またはリセットする方法、および更新間隔を調整する方法については、「[折れ線グラフ](#)」を参照してください。

メモリ使用グラフ

ツリーで「メモリ使用グラフ」を選択すると、次のページが表示されます。



このグラフには、バッファプールメモリの構造のみが表示されます。割り当てメモリと空きメモリが表示されます。

アナライザが折れ線グラフまたは棒グラフでアクティブになっている場合は、このページでもアクティブになっています。その逆の場合も同様です。アナライザを開始、一時停止、またはリセットする方法および更新間隔を調整する方法については、「[折れ線グラフ](#)」を参照してください。

レポート

ツリーで「レポート」ノードを展開すると、複数のページが使用可能です。これらは、バッファプールに関する特定のタイプの情報をファイルに書き込むために使用できます。

次のページを使用できます。

- [シンプルレポート](#)
- [ログ](#)

■ 高度分析

シンプルレポート

ツリーで [シンプルレポート] を選択すると、次のページが表示されます。



このページで選択したデータソースの情報を含むレポートを書き込むことができます。

▶手順 7.18. データソースを選択し、レポートを書き込むには

- 1 [有効なデータソース] リストボックスで、1つまたは複数のデータソースを選択します。
- 2 [>>] ボタンをクリックします。

選択したデータソースが [選択されたデータソース] リストボックスに移動されます。



Note: 誤って不適切なデータソースを [選択されたデータソース] リストボックスに移動した場合は、 [<<] ボタンを使用して、そのデータソースを [有効なデータソース] リストボックスに戻すことができます。

- 3 オプション。 [レポートファイル] テキストボックスに、レポートの書き込み先ファイルへのパスを指定します。

Or:

[ブラウズ] ボタンをクリックして、ダイアログボックスからファイルを選択します。

 **Note:** デフォルトで、[レポートファイル] テキストボックスにはプロパティで定義したファイルへのパスが指定されています。

4 [レポートの書き込み] ボタンをクリックします。

レポートが指定のファイルに書き込まれます。このファイルの内容は、プロパティで定義されたテキストエディタに自動的に表示されます。

ログ

ツリーで [ログ] を選択すると、次のページが表示されます。



シンプルレポート | ログ | 高度分析

データ選択

有効なデータソース(A):

- 割り当てメモリ
- 最小割り当てメモリ
- 最大割り当てメモリ
- 空きメモリ
- 最小連続空きメモリ
- 最大連続空きメモリ
- 現在のユーザ数
- 最大ユーザ数
- 削除された停止ユーザ数
- 未使用オブジェクト数
- アクティブなオブジェクト数

選択されたデータソース(S):

ファイル選択

ログファイル(F): C:\Temp\BP_Log.txt ブラウズ(B)...

ログコントロール

インターバル(I): 10 seconds すぐに開始(I) 停止(O)

開始時刻(B): 08/15/2007 09:40:01 スケジュールの開始(C)

停止時刻(E): 08/15/2007 09:40:01

ステータス:

このページの上部には、[シンプルレポート] ページと同じ情報が含まれています。唯一の違いは、異なるログファイルがデフォルトで使用されることです。

このページの上部でデータソースを選択する他に、ログファイルを直ちに（手動で）書き込むか、または特定の時間範囲でスケジュールするかを指定できます。

▶手順 7.19. ログ処理を手動で開始するには

- 1 **シンプルレポート**のページに記載されているように、すべてのデータソースおよび（オプションで）ログファイルを選択します。
- 2 [インターバル] ドロップダウンリストから、ログ情報をファイルに書き込む頻度を決める間隔を選択します。
- 3 [すぐに開始] ボタンをクリックして、ログファイルへの情報の書き込みを開始します。

[ログコントロール] グループボックスの下部に、終了数と経過時間を示すステータスメッセージが表示されます。
- 4 ログファイルへの情報の書き込みを停止するには、[停止] ボタンをクリックします。

ログ処理が停止すると、ログファイルの内容が、**プロパティ**で定義されたテキストエディタに自動的に表示されます。

▶手順 7.20. ログ処理を特定の時間範囲でスケジュールするには

- 1 **シンプルレポート**のページに記載されているように、すべてのデータソースおよび（オプションで）ログファイルを選択します。
- 2 [インターバル] ドロップダウンリストから、ログ情報をファイルに書き込む頻度を決める間隔を選択します。
- 3 開始日時を指定します。
- 4 停止日時を指定します。
- 5 [スケジュールの開始] ボタンをクリックします。

[ログコントロール] グループボックスの下部に、ステータスメッセージが表示されます。これは、ログ処理が開始されるまでの経過時間を示します。開始時刻になると、別のステータスメッセージが表示され、終了数と経過時間が示されます。



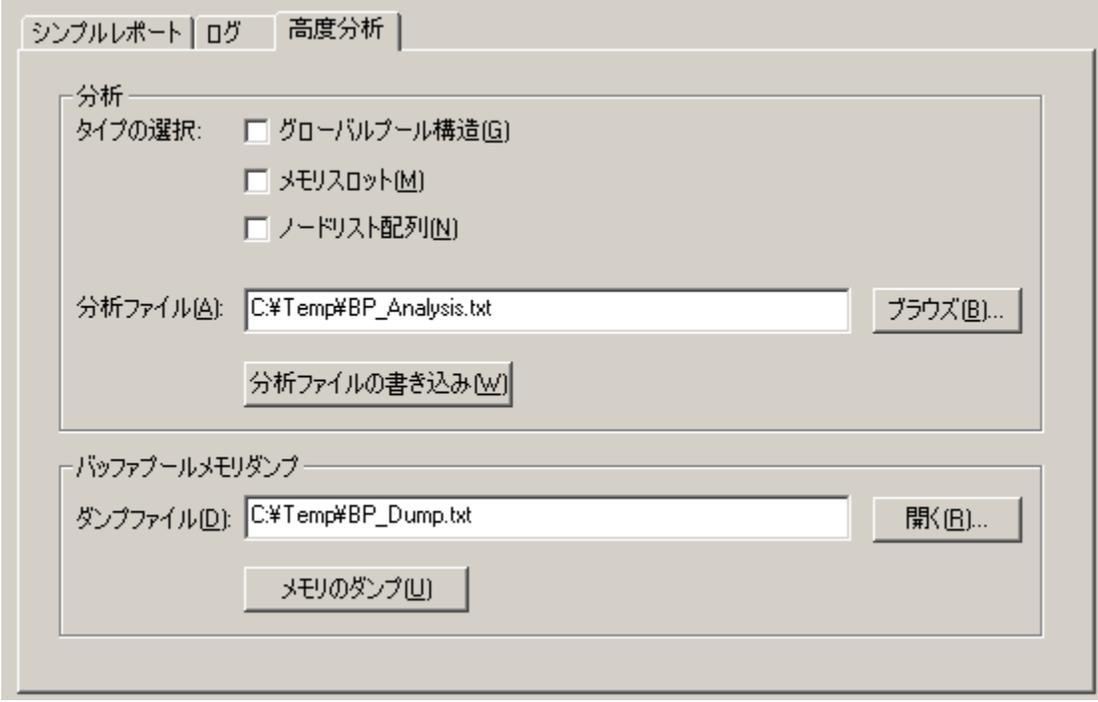
Note: 指定した開始時刻の前にスケジュールをキャンセルするには、[停止] ボタンをクリックします。

停止時刻になると（または開始時刻後に [停止] ボタンをクリックすると）、ログファイルの内容が、**プロパティ**で定義されたテキストエディタに自動的に表示されます。

高度分析

ツリーで「高度分析」を選択すると、次のページが表示されます。このページは、Software AG サポートに情報を提供します。

 **Important:** Software AG サポートに依頼されない限り、このページは使用しないでください。



シンプルレポート | ログ | 高度分析

分析
タイプの選択: グローバルプール構造(G)
 メモリスロット(M)
 ノードリスト配列(N)

分析ファイル(A): C:\Temp\BP_Analysis.txt ブラウズ(B)...

分析ファイルの書き込み(W)

バッファプールメモリダンプ
ダンプファイル(D): C:\Temp\BP_Dump.txt 開く(O)...

メモリのダンプ(D)

8 コマンド行バージョンのバッファプールモニタの使用 (NATBPMON)

| | |
|-------------------------------|----|
| ▪ NATBPMON ユーティリティの呼び出し | 72 |
| ▪ NATBPMON コマンド | 73 |
| ▪ バッファプールのオブジェクトの表示 | 74 |
| ▪ パターンの指定 | 75 |
| ▪ バッファプール設定の表示 | 76 |
| ▪ バッファプールの統計情報 | 77 |

このchapterでは、次のトピックについて説明します。

バッファプールの一般的な情報およびバッファプールの起動方法について記載されている「[Natural バッファプール](#)」も参照してください。

 **Caution:** このユーティリティを使用すると、バッファプールの他のユーザーの作業を破損する可能性があるため、Natural のすべての一般ユーザーがこのユーティリティにアクセスできるようにはしないでください。

NATBPMON ユーティリティの呼び出し

デフォルトバッファプール NATBP 用または別の既存のバッファプール用に、NATBPMON ユーティリティを呼び出すことができます。

▶手順 8.1. NATBPMON ユーティリティを呼び出すには

- 1 Windows のコマンドプロンプトウィンドウを起動します。
- 2 `natbpmon.exe` ファイルが保存されている Natural `bin` ディレクトリに移動します。
- 3 デフォルトバッファプール NATBP を使用する場合は、次のコマンドをコマンドプロンプトウィンドウに入力します。

```
NATBPMON
```

Or:

別のバッファプールを使用する場合は、次のコマンドをコマンドプロンプトウィンドウに入力します。

```
NATBPMON BP=buffer-pool-name
```

次のプロンプトが表示されます。

```
NATBPMON>
```

NATBPMON コマンド

NATBPMON プロンプトには、次のコマンドを入力できます。

| コマンド | 説明 |
|----------------------------------|--|
| CLEAR | これは、ZERO コマンドと同じです。 |
| CORPSES | 削除オブジェクトのリストを表示します。削除オブジェクトとは、削除済みだが削除を実行したときにバッファプールでまだ使用中だったオブジェクトのことです。このオブジェクトが使用されなくなると、削除オブジェクトのリストから自動的に消えます。 注意: DIR マンドで表示される列 cusr に、オブジェクトが使用中かどうかを示されます。 |
| DELETE { <i>pattern</i> [*]} | オブジェクトをバッファプールから削除します。アスタリスク (*) を使用すると、バッファプールからすべてのオブジェクトを削除できます。パターンを使用して、オブジェクトのコレクションを指定します (現在のオペレーティングシステムで可能なワイルドカードでのファイルのクラス指定と同様)。詳細については、「 パターンの指定 」を参照してください。 |
| DIR { <i>pattern</i> [*]} | バッファプール内のすべてのオブジェクトを含むディレクトリを表示します。詳細については、「 パターンの指定 」および「 バッファプールのオブジェクトの表示 」を参照してください。 |
| DUMP | エラー分析に使用されます。 重要: Software AG サポートに依頼されない限り、このコマンドは使用しないでください。 |
| EXIT | NATBPMON ユーティリティを終了します。 |
| FIN | NATBPMON ユーティリティを終了します。これは、EXIT コマンドと同じです。 |
| HELP | NATBPMON ユーティリティの使用可能なすべてのコマンドのリストを表示します。 |
| PARAM | バッファプール設定を表示します。詳細については、「 バッファプール設定の表示 」を参照してください。 |
| QUIT | NATBPMON ユーティリティを終了します。これは、EXIT コマンドと同じです。 |
| SHUTDOWN | バッファプールをシャットダウンします。このコマンドが発行されると、新しいプロセスでバッファプールを使用できません。NATBPMON ユーティリティは、ステータスが「シャットダウン中」のバッファプールで実行できます。この場合は、NATBPMON ユーティリティのすべてのコマンドが使用可能です。 注意: シャットダウン後にバッファプールを開始するには、ユーティリティ NATBPSRV を使用できます。 |
| STATUS | バッファプールの統計情報を表示します。詳細については、「 バッファプールの統計情報 」を参照してください。 |

| コマンド | 説明 |
|-------|---|
| SWAP | 読み取り専用バッファプールにのみ使用可能です。読み取り専用バッファプールに「現在使用されていない」というタグを付けます。このバッファプールに接続されたすべてのNaturalセッションは、このバッファプールから切断され、代替バッファプールに接続されます。 |
| WHO | バッファプールを使用しているすべてのユーザーのリストを表示します。NATBPMONユーティリティが各バッファプールユーザーに自動的に割り当てる番号（インデックス）と、バッファプールを使用しているプロセスのユーザーID、端末ID、およびプロセスID (tid) について、統計が表示されます。 |
| WRITE | バッファプールオブジェクトをディスクに書き込みます。インデックスとファイル名を指定する必要があります。 注意: DIR コマンドで表示される列「indx」に、インデックス番号が示されます。 |
| ZERO | STATUS コマンドで表示されるすべてのカウンタを0にリセットします。 |

バッファプールのオブジェクトの表示

DIR コマンドは、オブジェクトのリストを表示します。リストには、次の情報が含まれます。

| 列 | 説明 | |
|------|--|--|
| indx | オブジェクトがバッファプールにロードされるときに NATBPMON ユーティリティによってオブジェクトに自動的に割り当てられる番号です。 | |
| cusr | バッファプールでオブジェクトを使用している現在のユーザー数です。 | |
| pusr | バッファプールでオブジェクトが同時にアクティブにされた最大回数です。 | |
| nusg | バッファプールでオブジェクトがアクティブにされた回数です。 | |
| g | オブジェクトがシステムファイルからバッファプールにロードされているかどうかを示します。次のいずれかの値が表示されます。 | |
| | 0 | オブジェクトはロードされていません。 |
| | 1 | オブジェクトはロードされています。 |
| size | バッファプール内のオブジェクトのサイズをバイト単位で示します。 | |
| gpv | 生成プログラムのバージョン番号です。 | |
| key | オブジェクトに関する次の情報を示します。 | |
| | D | データベース ID。 |
| | F | ファイル番号。 |
| | L | オブジェクトが存在するライブラリ。 |
| | N | オブジェクトの名前です。番号および "@" は、現在ロードされているライブラリ用の一連の FILEDIR.SAG を示します。 |
| | K | オブジェクトの種類 ("G" = 生成されたオブジェクトモジュール、"S" = ソース、"D" = FILEDIR.SAG の一部、"R" = リソース)。 |

| 列 | 説明 |
|---|---------------------------------------|
| T | オブジェクトタイプ (" [K] "フィールドが "D" の場合は空白)。 |

DIRコマンドが発行されると、プール内のすべてのオブジェクトが以下のように表示されます。

```

indx:      index of the element
cusr:      current number of concurrent users
pusr:      peak number of concurrent users
nusg:      number of usages
g :        set if object is generating
gpv :      version of generated program

indx | cusr | pusr | nusg | g | size | gpv | key
-----+-----+-----+-----+-----+-----+-----+-----
1 | 0 | 1 | 4 | 0 | 920 | | (D=99 F=101 L="DEMO"
N="SEL-MAP" K='G' T='M')
2 | 1 | 7 | 2 | 0 | 3096 | | (D=99 F=101 L="DEMO" N="EMWND"
K='G' T='P')
3 | 4 | 9 | 4 | 0 | 604 | | (D=99 F=101 L="DEMO" N="HDR"
K='G' T='P')
4 | 2 | 3 | 7 | 0 | 412 | | (D=99 F=101 L="RPA"
N="MMUPROG1" K='G' T='P')
5 | 0 | 1 | 5 | 0 | 372 | | (D=99 F=101 L="RPA"
N="MMUPROG2" K='G' T='P')
6 | 0 | 5 | 4 | 0 | 372 | | (D=99 F=101 L="RPA"
N="MMUPROG3" K='G' T='P')

```

パターンの指定

パターンは、コマンド DIR および DELETE で指定できます。このセクションの例は、DIR コマンドに適用されます。

複数のオブジェクトを選択する際に、一致パターン表現を指定することによって、特定のキーフィールドの値に制限できます。

指定フィールドの対象フィールド値に制限するには、以下のパターン指定を使用する必要があります。

```
name=expression
```

複数のパターンをコンマで区切って指定できます。

指定したパターンはすべて、キー全体を受け入れるために対応するフィールドと一致する必要があります。

表現には、ワイルドカード文字 "*" および "?" を指定できます。

文字 "*" は、任意の文字シーケンスがある場合（または、ない場合）に一致します。ワイルドカード文字 "?" は、正確に 1 文字に一致します。

例

上記のサンプルのタイプ"P"のオブジェクトをすべて選択するには、以下のコマンドを使用します。

```
DIR T=P
```

DEMO ライブラリのプログラムをすべて選択するには、以下のコマンドを使用します。

```
DIR T=P, L=DEMO
```

名前に "M" を含むオブジェクトをすべて選択するには、以下のコマンドを使用します。

```
DIR N=*M*
```

バッファプール設定の表示

PARAM コマンドでは、次の設定が表示されます。

```
Active since .....: 4-JAN-2007 10:16:52, Version 6.3(631)
Last time cleared .....: 4-JAN-2007 10:16:52

Bpid .....: NATBP
Shmkey .....: Global\NAT631BPMEM_0x16221111
Semkey .....: Global\NAT631BPSEM_0x16221111
Memsized .....: 3145728
Maxusers .....: 20
```

| | |
|-----------------|--------------------------------------|
| Bpid | バッファプール ID。 |
| Shmkey | バッファプールの作成またはバッファプールへの接続に使用される一意の名前。 |
| Semkey | バッファプールメモリへのアクセスを同期するのに使用される一意の名前。 |
| Memsized | 使用可能な共有メモリのサイズ。 |
| Maxusers | バッファプールへ同時にアクセスすることができる最大ユーザー数。 |

『コンフィグレーションユーティリティ』ドキュメントの「バッファプール割り当て」を参照してください。

バッファプールの統計情報

STATUS コマンドでは、次の統計が表示されます。

```

Active since .....: 4-JAN-2007 10:16:52, Version 6.3(631)
Last time cleared .....: 4-JAN-2007 10:16:52
Bpid .....: NATBP
Allocated memory (bytes) ..: 18207752 Max users .....: 50
Smallest allocation .....: 32 Current users .....: 5
Largest allocation .....: 4707272 Peak users .....: 9
Free memory (bytes) .....: 2763768 Dead users purged .....: 623
Smallest free .....: 136
Largest free .....: 33136

Dormant objects .....: 2010 Smallest object (bytes) ..: 15
Active objects .....: 1 Largest object (bytes) ..: 49383
Generating objects .....: 0 Total object sizes .....: 12394644
Obsolete objects .....: 0

Attempted locates .....: 1646837910 Stored objects .....: 0
Attempted fast locates ....: 823725223 Loaded objects .....: 1520609
Successful fast locates ...: 822423506 Activated objects .....: 1645410434
Percent .....: 99.84 Aborted loads .....: 323104

Dormant objects purged ....: 51919 Peak parallel activations: 4
Object reuse factor .....: 1082.07
    
```

| 全般的な情報 | |
|---------------------------------|----------------------------------|
| Active since | バッファプールが開始された日時とバッファプールのバージョン番号。 |
| Last time cleared | バッファプールが最後にクリアされた日時。 |
| Bpid | バッファプール ID。 |
| メモリ割り当て | |
| Allocated memory (bytes) | 割り当てられたすべてのメモリの合計。 |
| Smallest allocatio | 割り当てられたメモリの最小量。 |
| Largest allocation | 割り当てられたメモリの最大量。 |
| Free memory (bytes) | すべての空きメモリの合計。 |
| Smallest free | 連続している空きメモリの最小量。 |
| Largest free | 連続している空きメモリの最大量。 |
| ユーザー統計 | |

| | |
|--------------------------------|--|
| Max. users | バッファプールへ同時にアクセスすることができる最大ユーザー数。『コンフィグレーションユーティリティ』ドキュメントの「バッファプール割り当て」を参照してください。 |
| Current users | バッファプールを現在使用しているユーザーの数。 |
| Peak users | バッファプールを使用していたユーザーの最大数。 |
| Dead users purged | バッファプールから削除された非アクティブユーザーの数。ゼロ (0) に近い数値である必要があります。この数値の増分は、バッファプールユーザーのエントリ (つまり、Natural セッション) が無条件でキャンセルまたは終了されたことを示します。バッファプールマネージャによってそのようなユーザーのエントリが識別されるたびに、この数値が増え、キャンセルされたセッションによってバッファプールに残された不要物を削除するためにクリーンアップが実行されます。 |
| オブジェクト使用統計 | |
| Dormant objects | 使用可能だがアクティブではないオブジェクトの数。これらのオブジェクトはバッファプールにあります。使用されていません。これらは後で使用することができ、バッファプールユーザーがその使用を要求するとすぐにアクティブになります。 |
| Active objects | アクティブなオブジェクトの数。これらのオブジェクトは、バッファプールユーザーに現在使用されています。 |
| Generating objects | バッファプールに現在ロードされているオブジェクトの数。これらのオブジェクトは、ロード操作が完了するとすぐに使用可能になります。 |
| Obsolete objects | バッファプールから削除されることになっているが、まだ使用されているオブジェクトの数。これらオブジェクトは、CORPSES コマンドを使用して表示できます。使用されなくなったオブジェクトは、このオブジェクトをアクティブにしたすべてのユーザーがこのオブジェクトを解放するとすぐにバッファプールから削除されます。実稼働環境では、この数値はゼロ (0) である必要があります。0 以外の値は、オブジェクトが NATBPMON の DELETE コマンドを使用して削除されたか、新しいオブジェクトが作成 (例えば、CATALOG コマンドによって) されたために使用されなくなったかのいずれかであることを示します。 |
| オブジェクトサイズ統計 | |
| Smallest object (bytes) | バッファプール内の最小オブジェクトのサイズ。 |
| Largest object (bytes) | バッファプール内の最大オブジェクトのサイズ。 |
| Total object sizes | バッファプール内のすべてのオブジェクトの合計サイズ。 |
| ロケート統計 | |
| Attempted locates | オブジェクトの検索が成功および失敗した回数。この数値は、バッファプールマネージャがバッファプールでオブジェクトを検索するように要求された回数を示します。 |
| Attempted fast locates | 既知のロットでアクティブにしようとした回数。オブジェクトがすでに検索されていた場合は、オブジェクトをアクティブにした回数になります。オブジェクトが再度アクティブにされるたびに、オブジェクトがバッファプール内の同じ場所で見つかる可能性は非常に高いです。 |
| Successful fast locates | 高速検索が正常に行われた回数。 |
| Percent | 高速検索が正常に行われた割合。 |

| オブジェクトのロード統計 | |
|----------------------------------|---|
| Stored objects | バッファプールに保存されたオブジェクトの数。これは、バッファプールに保存され、システムファイルからロードされなかったオブジェクトの数です。 |
| Loaded objects | システムファイルからロードされたオブジェクトの数。オブジェクトがバッファプールで見つからない場合は、常にオブジェクトはシステムファイルからロードされます。この数値は、オブジェクトがバッファプールに正常にロードされるたびに増えます。 |
| Activated objects | アクティブにされたオブジェクトの数。アクティブ化とは、バッファプールで見つかったオブジェクトがバッファプールユーザーによって「使用中」とマークされるプロセスです。 |
| Aborted loads | バッファプール内のメモリ不足のため、またはバッファプールへのオブジェクトのロード中に発生したエラーのため、中止されたロード操作の数。この数値は大幅に変化しないようにする必要があります。 |
| 全般的なロード統計 | |
| Dormant objects purged | 新しくロードされるオブジェクト用に領域を空けるため、バッファプールから削除された未使用のオブジェクトの数。 |
| Peak parallel activations | バッファプール内のオブジェクトの1つが同時にアクティブにされた最大回数。 |
| Object reusage factor | オブジェクトが再度アクティブにされた回数の平均値。この数値は、バッファプールにロードされたオブジェクト数に対するオブジェクトアクティブ化回数の割合です。 |

9 バッチモードでの Natural

| | |
|--|----|
| ■ バッチモードについて | 82 |
| ■ バッチモードの Natural セッションの開始 | 82 |
| ■ バッチモードの Natural セッションの終了 | 83 |
| ■ バッチモードでの Natural の使用 | 83 |
| ■ バッチモードのサンプルセッション | 85 |
| ■ バッチモード検出 | 88 |
| ■ バッチモードの制限事項 | 89 |
| ■ バッチモードで Natural マップおよびダイアログを使用するためのヒント | 89 |

このchapterでは、Natural をバッチモードで実行する場合の特別な考慮事項について説明します。次のトピックについて説明します。

バッチモードについて

Natural では、次の2つの処理モードが区別されます。

- 対話式モード (Natural スタジオで実行)
- バッチモード

2つのモードの主な違いは、次のとおりです。対話式モードでは、コマンドとデータはユーザーがキーボードから入力し、出力は画面に表示されます。バッチモードでは、ユーザーの操作なしに、入力はファイルから読み取られ、出力はファイルに書き込まれます。

Natural をバッチジョブとして実行する場合は、バッチジョブをサブミットしたユーザーと Natural との対話は不要です。バッチジョブは、順番の決まった入力データを受け取って順番に実行されるプログラムで構成されます。

バッチモードは、大量のデータ処理を定期的に行う場合に便利です。

バッチモードの Natural セッションの開始

バッチモードをアクティブにするには、パラメータ BATCHMODE を使用します。

▶手順 9.1. Natural セッションをバッチモードで開始するには

- 1 次のようにダイナミックパラメータ BATCHMODE を使用して、Natural を起動します。

```
natural BATCHMODE
```

上記の呼び出し (BATCHMODE パラメータのみの指定) は、必要な入力および出力チャンネルがコンフィグレーションユーティリティですでに定義されていることを前提にしています。入力および出力チャンネルの詳細については、このセクションで後述する「[バッチモードでの Natural の使用](#)」を参照してください。パラメータファイル内のバッチモード関連プロファイルパラメータの詳細については、『コンフィグレーションユーティリティ』ドキュメントの「バッチモード」を参照してください。

必要な入力および出力チャンネルをダイナミックパラメータとして上記の呼び出しに追加することもできます。これについては、このセクションで後述する「[バッチモードのサンプルセッション](#)」で示します。上記の呼び出しでダイナミックパラメータとして指定された入力および出力チャンネルは、パラメータファイル内のチャンネル定義を上書きします。

- 出力チャンネルとして定義されているファイルを確認します。このファイルの末尾に、セッションが正常に終了したことを示すメッセージが含まれている必要があります。

バッチモードの Natural セッションの終了

バッチモードの Natural セッションは、セッション中に次のいずれかを検出すると終了します。

- **バッチ入力ファイル**内のシステムコマンド `FIN`
- 実行中の Natural プログラム内の `TERMINATE` ステートメント



Note: バッチ入力ファイルで入力終了条件が発生した場合も、バッチセッションは終了します。この場合は、出力チャンネルとして定義されているファイルに、予期しない終了を示すメッセージが含まれます。

バッチモードでの Natural の使用

Natural セッションをバッチモードで開始するには、ダイナミックパラメータ `BATCHMODE` を指定する必要があります。さらに、入力および出力チャンネルを定義する必要があります（下記参照）。

! **Important:** 入力チャンネル `CMSYNIN` または `CMOBJIN` と出力チャンネル `CMPRINT` は、バッチモードに常に必要です。

以下では次のトピックについて説明します。

- [入力および出力チャンネル](#)
- [入力および出力ファイル用のコードページ](#)

入力および出力チャンネル

バッチモードでは、次のパラメータを使用できます。

| パラメータ | 説明 |
|----------------------|---|
| <code>CMSYNIN</code> | Natural コマンドを含むバッチ入力ファイル（およびオプションで Natural プログラムの実行中に <code>INPUT</code> ステートメントによって読み込まれるデータ）を定義します。 |
| <code>CMOBJIN</code> | <code>INPUT</code> ステートメントによって読み込まれるデータを含むバッチ入力ファイルを定義します。このデータは、関連する <code>RUN</code> または <code>EXECUTE</code> コマンドの直後に、パラメータ <code>CMSYNIN</code> で定義されたファイルに配置することもできます。 |
| <code>CMPRINT</code> | Natural プログラムの <code>DISPLAY</code> 、 <code>PRINT</code> 、および <code>WRITE</code> ステートメントから生成される出力用のバッチ出力ファイルを定義します。 |

| パラメータ | 説明 |
|------------|---|
| CMPRT nn | セッション中に実行される Natural プログラムによって参照される追加のレポート用の出力ファイルを定義します。 nn は、DISPLAY、PRINT、または WRITE ステートメントで使用されるレポート番号に対応する 01~31 の 2 桁の 10 進数です。 |
| CMWRK nn | セッション中に実行される Natural プログラムによって参照されるワークファイルを定義します。 nn は、READ WORK FILE または WRITE WORK FILE ステートメントで使用される番号に対応する 01~32 の 2 桁の 10 進数です。 |
| NATLOG | パラメータ CMPRINT で定義されたバッチ出力ファイルに書き込めなかったメッセージをログに記録します。 バッチモードでは NATLOG を有効にすることをお勧めします。 |

入力および出力ファイル用のコードページ

次のパラメータは、入力ファイルがエンコードされるコードページと出力ファイルがエンコードされるコードページを指定します。

| パラメータ | 説明 |
|---------|--|
| CPSYNIN | コマンドのバッチ入力ファイルのエンコードに使用するコードページを指定します。このファイルは、パラメータ CMSYNIN で定義されます。 |
| CPOBJIN | データのバッチ入力ファイルのエンコードに使用するコードページを指定します。このファイルは、パラメータ CMOBJIN で定義されます。 |
| CPPRINT | バッチ出力ファイルのエンコードに使用するコードページを指定します。このファイルは、パラメータ CMPRINT で定義されます。 |

CMSYNIN および CMOBJIN のエンコード：

- コードページが入力ファイル CMSYNIN または CMOBJIN のいずれかに指定されている場合は、入力ファイル内のデータはコードページを使用してエンコードされているとみなされます。
- コードページが入力ファイル CMSYNIN または CMOBJIN のいずれにも指定されていない場合は、入力ファイル内のデータは Natural パラメータ CP で指定されたデフォルトのコードページを使用してエンコードされているとみなされます。
- コードページが Natural パラメータ CP で指定されていない場合は、入力ファイル内のデータは現在のシステムコードページを使用してエンコードされているとみなされます。

CMPRINT のエンコード：

- コードページが出力ファイル CMPRINT に指定されている場合は、出力データはこのコードページを使用してエンコードされます。
- コードページが出力ファイル CMPRINT に指定されていない場合は、Natural パラメータ CP で指定されたデフォルトのコードページを使用してエンコードされます。
- コードページが Natural パラメータ CP で指定されていない場合は、出力データは現在のシステムコードページを使用してエンコードされます。

エンコード／デコードに失敗すると（例えば、ファイルのエンコードに使用されたコードページに含まれていない CMPRINT に文字が書き込まれる場合）、バッチジョブはエンコード／デコードエラーが発生したファイルを示すスタートアップエラー 42（バッチモードドライバエラー）で終了します。

これらの各パラメータで、特に、UTF-8 をコードページとして指定することができます。これにより、UTF-8 でエンコードされた Unicode データの読み書きが可能になります。

バッチモードのサンプルセッション

この例は、Natural をバッチモードで開始する方法を示します。簡単な Natural プログラムが実行され、データ項目がバッチ入力ファイルから取得されます。項目が INPUT ステートメントで処理されると、続けて DISPLAY ステートメントによってデータがバッチ出力ファイルに書き込まれます。その後、Natural が終了します。

この例では、ライブラリ SYSEXBAT に保存されているプログラム RECCONT を使用します。

 **Note:** このライブラリに保存されているオブジェクトについては、ライブラリ SYSEXBAT のテキスト A-README を参照してください。

以下の呼び出しによって、サンプルセッションが呼び出されます。

```
natural BATCHMODE CMSYNIN=cmd.txt CMOBJIN=data.txt CMPRINT=out.txt NATLOG=ALL
```

 **Note:** この呼び出しでは、すべてのファイルが現在のディレクトリにあり出力がこのディレクトリに書き込まれることを前提としています。ファイルが別のディレクトリにあるか、出力が別のディレクトリに書き込まれる場合は、パスを指定する必要があります。

上記の呼び出しのパラメータについて、以下で説明します。

BATCHMODE

パラメータ BATCHMODE は、バッチモードを有効にし、システム変数 *DEVICE の値を "BATCH" に設定します。

CMSYNIN=cmd.txt

バッチ入力ファイル *cmd.txt* は、ファイルシステムに保存されるテキストファイルです。このファイルの内容を以下に示します。これには、ライブラリ SYSEXBAT へのログイン用、Natural プログラム RECCONT の実行用、および Natural セッションの終了用の Natural システムコマンドが含まれます。

```
LOGON SYSEXBAT
EXECUTE RECCONT
FIN
```

Natural プログラム RECCONT には、以下の内容が含まれます。

```
DEFINE DATA
LOCAL
  1 #firstname   (A10)
  1 #lastname    (A10)
END-DEFINE
INPUT (IP=OFF AD=M) #firstname #lastname
DISPLAY #firstname #lastname
END
```

CMOBJIN=data.txt

プログラム RECCONT の INPUT ステートメントは、バッチ入力ファイル *data.txt* で定義されたデータを使用します。これは、ファイルシステムに保存されるテキストファイルです。このファイルの内容を以下に示します。

```
Ben %
Smith
```



Note: 文字 "%" は、情報が次の行に続くことを示します。

CMPRINT=out.txt

プログラム RECCONT の DISPLAY ステートメントは、ファイルシステムに作成されたバッチ出力ファイル *out.txt* にデータを書き込みます。このファイルの内容を以下に示します。

```

NEXT LOGON SYSEXBAT
Logon accepted to library SYSEXBAT.
NEXT EXECUTE RECCONT

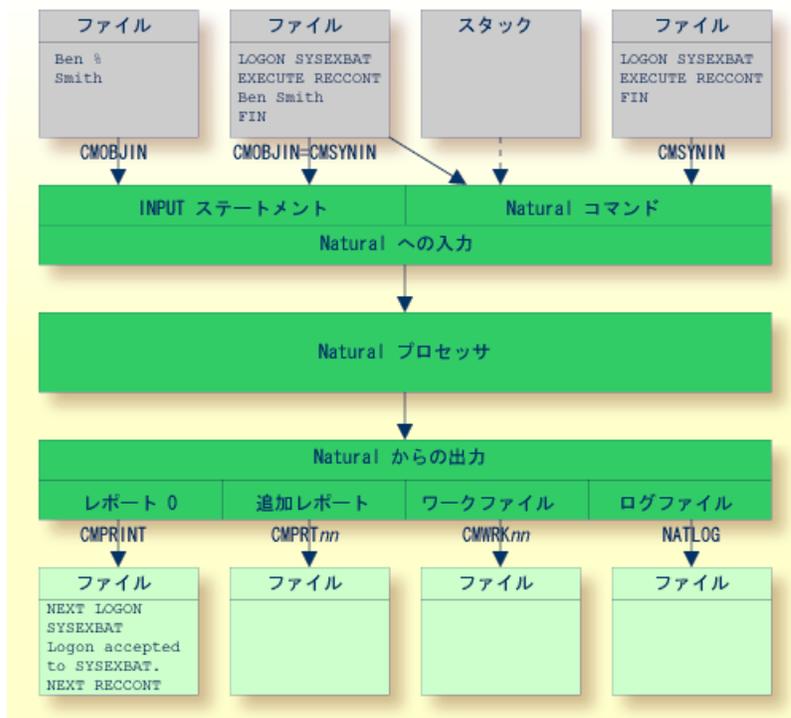
DATA Ben %
DATA Smith
Page      1                                25.04.05  13:39:09

#FIRSTNAME #LASTNAME
-----
Ben        Smith
NEXT FIN
NAT9995 Natural session terminated normally.
    
```

NATLOG=ALL

上記の呼び出しでサンプルセッションを呼び出すと、すべてのタイプのメッセージが含まれるログファイルが作成されます（これには、バッチ入力および出力ファイルの名前も含まれます）。ログファイルは通常、ローカルコンフィグレーションファイルで定義される Natural の一時ディレクトリに作成されます。NATLOG パラメータの説明も参照してください。

以下の図は、Natural がバッチモードで入力の読み取りおよび出力の書き込みを行うさまざまな方法を示しています。



上の図に示されているように、以下のいずれかの方法で続行できます。

■ CMOBJIN およびCMSYNIN

異なるファイルがバッチ入力に使用されます。1つのファイルには Natural コマンドが含まれ、他のファイルにはデータが含まれます。

```
natural BATCHMODE CMSYNIN=cmd.txt CMOBJIN=data.txt CMPRINT=out.txt
```

■ CMSYNIN

1つのファイルがバッチ入力に使用されます。これには Natural コマンドとデータの両方が含まれます。

```
natural BATCHMODE CMSYNIN=data.txt CMOBJIN=data.txt CMPRINT=out.txt
```



Note: バッチ入力ファイルを1つだけ使用する場合でも、パラメータ CMSYNIN および CMOBJIN の両方を指定する必要があります。両方のパラメータは同じファイルを参照する必要があります。

■ CMOBJIN およびSTACK

1つのファイルがバッチ入力に使用されます。これにはデータが含まれます。Natural コマンドは、プロファイルパラメータ STACK で指定されます。

```
natural BATCHMODE CMOBJIN=data.txt STACK="(LOGON SYSEXBAT; RECCONT;FIN)"
```

バッチモード検出

システム変数 *DEVICE は、Natural がバッチモードまたは対話式モードのどちらで実行されているかを示します。

| モード | 説明 |
|--------|---|
| バッチモード | *DEVICE には、値 "BATCH" が含まれます。この値は、パラメータ BATCHMODE で設定されます。 |
| 対話式モード | *DEVICE には、"BATCH" 以外の値が含まれます。ほとんどの場合は、値 "VIDEO" が含まれます。 |

例：

```
IF *DEVICE = "BATCH" THEN
  WRITE 'This is the background task'
ELSE
  WRITE 'This is the interactive session'
END-IF
```

バッチモードの制限事項

Natural をバッチモードで実行している場合は、以下のような制限があります。

- 対話式の入力または出力はできません。
- マウスサポートはありません。
- 異なる文字フォントは使用できません。
- INPUT ステートメントに対するデータのみが処理可能です。ダイアログ入力は条件付きでのみサポートされます（「バッチモードで Natural マップおよびダイアログを使用するためのヒント」を参照）。
- 出力表示は GUI 形式ではありません（文字ベースの出力です）。
- 色およびビデオ属性は、CMPRINT で定義されるバッチ出力ファイルに書き込まれます。
- 充填文字は INPUT ステートメント内に表示されません。
- 特定の Natural システムコマンドはバッチモードでは実行できず、無視されます。この制限が適用される各システムコマンドの注意事項については、『システムコマンド』ドキュメントを参照してください。

バッチモードで Natural マップおよびダイアログを使用するためのヒント

アプリケーションがバッチモードでも対話式モードでも実行できるように設計されている場合は、以下の点を考慮する必要があります。

Natural 内では、入力データを読み込む方法が2つあります。

- マップ（INPUT ステートメントまたは Natural オブジェクトのマップ）を使用
- ダイアログ（Natural オブジェクトのダイアログ）を使用

バッチモードでは、INPUT ステートメントを使用してデータを処理する必要があります。ダイアログではバッチモードでデータを処理できません。ダイアログではデータを移動したり制御し

たりする端末コマンドもサポートされません。それでも、ダイアログをバッチモードで実行できます。ただしこの場合は、ダイアログを以下の方法で変更する必要があります。

- ダイアログ属性 `VISIBLE` を `"FALSE"` に設定する必要があります。
- `AFTER-OPEN` イベント内に、バッチモード処理中にデータを読み込むためのコードを挿入する必要があります。Natural をバッチモードで実行する場合は、入力データを取得するために `INPUT` ステートメントをコード化します。対話式モードでダイアログが表示されるように、ダイアログ属性 `VISIBLE` を `"TRUE"` に設定する必要があります。

`AFTER-OPEN` イベントの例：

```
IF *DEVICE EQ "BATCH" THEN
/* Batch mode processing: call a map */
  INPUT USING MAP "BATCHINP" #p1 #p2 #p3

/* ... further data processing ... */

/* Close dialog immediately */
  CLOSE DIALOG *DIALOG-ID
  ELSE
/* Interactive mode processing: make dialog visible */
  #DLG$WINDOW.VISIBLE = TRUE
END-IF
```

- `CLOSE` イベントがある場合は、バッチモードで該当のコードに GUI アクションが含まれていないことを確認します。

`CLOSE` イベントの例：

```
IF *DEVICE NE "BATCH" THEN
/* ... GUI actions ... */
END-IF
```

10 出力ウィンドウ

- 出力ウィンドウについて 92
- 出力ウィンドウでの作業 92
- 出力ウィンドウプロファイルの変更 93
- 出力ウィンドウ用の独自のアイコンの使用 94

このchapterでは、次のトピックについて説明します。

出力ウィンドウについて

ローカル環境では、Naturalプログラムによって出力が画面に書き込まれると出力ウィンドウが表示されます。例：



 **Note:** リモート開発環境では、出力ウィンドウではなく、端末エミュレーションウィンドウまたはWeb I/O インターフェイスクライアントが表示されます。『SPoDを使用したリモート開発』ドキュメントの「端末エミュレーション」および「Web I/O インターフェイスクライアント」を参照してください。

出力ウィンドウでの作業

Windows タスクバーに出力ウィンドウのボタンが表示されます。出力ウィンドウを最小化するときはこのことに注意してください。

出力ウィンドウはサイズ変更および移動できます。出力ウィンドウが Natural 出力ページより小さい場合は、スクロールバーが表示されます。

Natural プログラムで定義された PF キーはコマンドボタンに変換されます。これらのコマンドボタンかキーボードの PF キーのいずれかを使用できます。

Natural 出力の入力フィールドの情報は、切り取って Windows クリップボードにコピーでき、Windows クリップボードから出力ウィンドウの入力フィールドに貼り付けることができます。

カーソルは、マウスを使用して配置できます。ダブルクリックすると、Enter キーがシミュレートされます。システム変数 *CURSOR、*CURS-COL、および *CURS-LINE は常に、現在のマウス位置に設定されます。

ステートメント DEFINE WINDOW または端末コマンド %W を使用して作成されたウィンドウは、出力ウィンドウに配置されます。これらのウィンドウは、出力ウィンドウの子ウィンドウとして、移動、サイズ変更、およびスクロールできます。

出力ウィンドウは、プログラムを終了すると自動的に閉じます。Windows の標準の閉じるボタンを使用して手動で閉じることはできません。プログラムの通常の終了の前に出力ウィンドウを閉じるには、Esc キーを押します（この機能が無効になっていない場合。下記参照）。

出力ウィンドウプロファイルの変更

複数のオプションを出力ウィンドウに設定できます。これらのオプションは、出力ウィンドウで直接設定するか（下記参照）、Natural スタジオで設定できます（『Natural スタジオの使用』ドキュメントの「オプションの設定」を参照）。

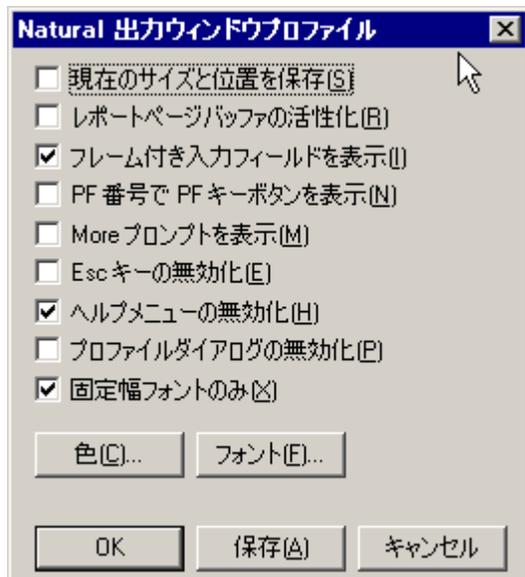
▶手順 10.1. 出力ウィンドウプロファイルを変更するには

- 1 出力ウィンドウのコントロールメニューの [プロファイル] を選択します。



Note: プロファイルダイアログボックスの表示が無効になっている場合は、このコマンドは使用できません。この場合は、出力ウィンドウオプションは Natural スタジオでのみ設定できます。

次のダイアログボックスが表示されます。



このダイアログボックスには、Naturalスタジオの [オプション] ダイアログボックスと同じオプションが含まれています。ただし例外があります。次のコマンドは、実行中のアプリケーションにしか指定できないため、上記のダイアログボックスでしか使用できません。

現在のサイズと位置を保存

このチェックボックスをオンにすると、出力ウィンドウの現在のサイズと位置が保存されます。

- 2 必要なオプションをすべて設定します。各オプションの詳細については、『Natural スタジオの使用』ドキュメントの「出力ウィンドウオプション」を参照してください。
- 3 [保存] ボタンをクリックすると、変更内容が永続的に保存されます。

Or:

[OK] ボタンをクリックすると、変更内容が現在のセッションにのみ保存されます。

出力ウィンドウ用の独自のアイコンの使用

出力ウィンドウに、Naturalのアイコンを使用する代わりに、独自のアイコンを使用できます。このアイコンは、出力ウィンドウのコントロールメニューとWindowsタスクバーに表示されます。

▶手順 10.2. 独自のアイコンを使用するには

- 1 アイコンファイル (*.ico) を作成します。
- 2 アイコンファイルを、Naturalで検索できるディレクトリに保存します。

Naturalは、アイコンファイルを、最初にログオンライブラリのRESサブディレクトリで検索し、次に各 `steplib` のRESサブディレクトリ、次に Windows 環境変数 `NATGUI_BMP` に割り当てられたディレクトリで検索します。

- 3 独自のアイコンファイルを使用するには、プログラムで以下のステートメントを使用します。

```
SET CONTROL 'I=icon-file-name.ICO'
```

端末コマンド `%I` の説明も参照してください。

11 Natural ランタイム

| | |
|--|-----|
| ▪ Natural ランタイムの対象外について | 98 |
| ▪ 移植手順の概要 | 99 |
| ▪ 手順 1：開発ワークステーションでのアプリケーションのパッケージ | 100 |
| ▪ 手順 2：Natural ランタイムのインストール | 105 |
| ▪ 手順 3：ランタイムワークステーションへのアプリケーションのインストール | 105 |
| ▪ 手順 4：ランタイムワークステーションでのアプリケーションの開始 | 107 |
| ▪ Natural ランタイム起動サービスの使用 | 108 |

Natural ランタイムは、Natural for Windows の開発バージョンを使用して作成されたアプリケーションを実行するために使用します。

このchapterでは、アプリケーションを Natural 開発ワークステーションから Natural ランタイムワークステーションに移植する方法について説明します。この移植プロセスは、Natural ランタイムの初回インストールと Natural ランタイムワークステーションの更新に使用できます。

また、Natural ランタイムプロセスを開始するためのサービスの使用方法についても説明します。

次のトピックについて説明します。

 **Important:** アプリケーションを Natural ランタイムワークステーションに移植する前に、すべてのオブジェクトが、同じバージョンの Natural および Natural ランタイムを使用してコンパイルされていることを確認してください。

Natural ランタイムの対象外について

システムコマンド

以下の Natural システムコマンドは Natural ランタイムではサポートされません。

CATALL
CATALOG
CHECK
CLEAR
COMPOPT
DEBUG
DELETE
EDIT
GLOBALS
PURGE
READ
RENUMBER
RUN
SAVE
SCAN
SCRATCH
STOW
STRUCT
SYSDDM
SYSMAIN
UNCATALOG
UNLOCK

エディタ

Natural エディタは Natural ランタイムではサポートされません。

ユーティリティ

開発者用の機能を提供する Natural ユーティリティは Natural ランタイムではサポートされません。

移植手順の概要

アプリケーションをランタイムワークステーションに移植するには、次の手順に従います（詳細については、このセクションで後述します）。

1. 開発ワークステーションでアプリケーションをパッケージします。
 - a. ファイルシステムに集合ディレクトリを作成します。
 - b. グローバルコンフィグレーションファイルをカスタマイズして集合ディレクトリにコピーします。
 - c. Natural パラメータファイルをカスタマイズして集合ディレクトリにコピーします。
 - d. 必要なオブジェクトをすべて集合ディレクトリにコピーまたはアンロードします。
 - e. 集合ディレクトリの内容を送信媒体（CD など）にコピーします。
2. Natural ランタイムをランタイムワークステーションにインストールします。
3. アプリケーションをランタイムワークステーションにインストールします。
 - a. グローバルコンフィグレーションファイルを送信媒体からランタイムワークステーションにコピーします。
 - b. Natural パラメータファイルを送信媒体からランタイムワークステーションにコピーします。
 - c. Natural オブジェクトを送信媒体からランタイムワークステーションにコピーまたはロードします。
4. アプリケーションをランタイムワークステーションで開始します。

『プログラミングガイド』の「Natural 生成プログラムの転送」も参照してください。

手順1：開発ワークステーションでのアプリケーションのパッケージ

以下では次のトピックについて説明します。

- 集合ディレクトリの作成
- グローバルコンフィグレーションファイルのカスタマイズおよびコピー
- Natural パラメータファイルのカスタマイズおよびコピー
- オブジェクトのコピーまたはアンロード
- 送信媒体への集合ディレクトリのコピー

集合ディレクトリの作成

Windows エクスプローラを使用して、開発ワークステーションのファイルシステムに新しいディレクトリを作成します。この一時ディレクトリに、アプリケーションに属しているすべてのファイルを集めます。

グローバルコンフィグレーションファイルのカスタマイズおよびコピー

アプリケーションをランタイムワークステーションで実行するのに必要なすべての設定が含まれるグローバルコンフィグレーションファイルを作成する必要があります。これを行うには、現在のグローバルコンフィグレーションファイルのバックアップバージョンを作成し、ランタイムバージョンに必要なすべての変更を行い、カスタマイズされたグローバルコンフィグレーションファイルを集合ディレクトリにコピーします。その後、古いグローバルコンフィグレーションファイルを復元します。詳細については、以下で説明します。

▶手順 11.1. グローバルコンフィグレーションファイルのカスタマイズしコピーするには

- 1 Windows エクスプローラを使用して、既存のグローバルコンフィグレーションファイルをバックアップします。

 **Note:** グローバルコンフィグレーションファイルの場所がわからない場合は、コンフィグレーションユーティリティを呼び出し、[ローカルコンフィグレーションファイル] ノードを展開し、[インストール指定] ノードを選択します。すると、グローバルコンフィグレーションファイルの完全なパスおよび名前が表示されます。

- 2 コンフィグレーションユーティリティを呼び出し、[グローバルコンフィグレーションファイル] ノードを展開します。
- 3 アプリケーションに応じてグローバルコンフィグレーションファイルの設定を調整し、変更内容を保存します。以下の注も参照してください。



Important: ランタイム環境用の設定が含まれるグローバルコンフィグレーションファイルがアクティブである場合は、開発バージョンの Natural で作業することはできません。

- 4 Windows エクスプローラを使用して、カスタマイズされたグローバルコンフィグレーションファイルを集合ディレクトリにコピーします。
- 5 Windows エクスプローラを使用して、グローバルコンフィグレーションファイルのバックアップバージョンを復元します。

注：

- FNAT および FUSER システムファイルへのデフォルトパスは、開発環境とランタイム環境では異なります。ランタイム環境でのアプリケーションのインストールを容易にするには、パスを調整します。これを行うには、グローバルコンフィグレーションファイルの [システムファイル] ノードを選択し、ランタイムバージョンの設定への FNAT および FUSER パスを変更します。

例：以下の Natural 開発バージョンインストールのデフォルトパスを変更します。

| DBID | FNR | パス |
|------|-----|--|
| 99 | 100 | C:\Program Files\Software AG\Natural\ <i><n.n></i> \Fnat |
| 99 | 101 | C:\Program Files\Software AG\Natural\Natapps\Fuser |

これらを、以下の Natural ランタイムインストールのデフォルトパスに変更します。

| DBID | FNR | パス |
|------|-----|--|
| 99 | 100 | C:\Program Files\Software AG\Natural Runtime\ <i><n.n></i> \Fnat |
| 99 | 101 | C:\Program Files\Software AG\Natural Runtime\Natapps\Fuser |

n.n はバージョン番号です。

- 移植するすべての新しいアプリケーションで、新しい設定と古い設定に互換性があることを確認してください。

例：最初のアプリケーションは SQL データベースにアクセスし、DBID エントリはこの SQL データベースに適用されます。後の日付で移植する 2 番目のアプリケーションは、Adabas C データベースにアクセスします。この場合は、Adabas C 用に 2 番目の DBID エントリを追加する必要があります。2 番目のエントリを追加しないと、新しいグローバルコンフィグレーションファイルが SQL データベースの DBID を上書きし、最初のアプリケーションがそのデータベースにアクセスできなくなります。

Natural パラメータファイルのカスタマイズおよびコピー

アプリケーションをランタイムワークステーションで実行するのに必要なすべての設定が含まれる Natural パラメータファイルを作成する必要があります。

▶手順 11.2. Natural パラメータファイルのカスタマイズするには

- 1 コンフィグレーションユーティリティを呼び出し、必要なパラメータファイルのノードを展開します。
- 2 ランタイム環境で実行するアプリケーションに応じて、パラメータファイルの設定を調整します。

開始するプログラムの名前がパラメータファイルに含まれていることを確認します。以下の例では、この目的用のさまざまな選択肢を示しています。

例 1：

| パラメータ | 必須設定 |
|----------|--|
| AUTO | Natural セッションの開始時に自動ログオンが実行されるように、"ON" に設定します。 |
| INIT-LIB | アプリケーションの移動先ライブラリの名前。 |
| STARTUP | 開始するプログラムの名前。 |
| USER | Natural の起動時に設定されるデフォルトユーザー ID。 |

例 2：

| パラメータ | 必須設定 |
|-------|--|
| STACK | ライブラリと開始するプログラムが含まれている必要があります。次に例を示します。 LOGON MYLIB;EXECUTE MYAPP |

- 3 修正したパラメータファイルを、ランタイム環境で使用する名前（例：RUNPARM）で保存します。
- 4 Windows エクスプローラを使用して、カスタマイズされたパラメータファイル（拡張子 "SAG"）を集合ディレクトリにコピーします。



Note: パラメータファイルの場所がわからない場合は、コンフィグレーションユーティリティの [ローカルコンフィグレーションファイル] ノードを展開し、[インストール指定] ノードを選択します。すると、Natural パラメータファイルの位置が、[パラメータモジュールへのパス] として表示されます。

オブジェクトのコピーまたはアンロード

コンパイル済みコードを Natural ランタイムで使用できるようにするには、カタログ化オブジェクトを Natural 開発環境からランタイム環境にコピーする必要があります。

Natural アプリケーションが完全な Natural ライブラリで構成される場合は、Windows エクスプローラのコピーアンドペースト機能を使用してライブラリをコピーできます。

オブジェクトを移植する別の方法としては、オブジェクトハンドラを使用して、オブジェクトを Natural 開発環境でアンロードし、ランタイム環境でロードします。

▶手順 11.3. オブジェクトをコピーするには

- 1 Natural スタジオを使用して、ランタイムバージョン用のすべてのオブジェクトを含める新しいライブラリを作成します。



Important: アプリケーションが複数のライブラリで構成される場合は、アプリケーションで使用される各ライブラリ用に新しいライブラリを作成し、以下の手順を続行します。

- 2 Natural スタジオを使用して、すべてのカタログ化オブジェクト、リソース、およびエラーメッセージを開発ライブラリから新しいライブラリにコピーします。ソースはコピーしないでください。
- 3 Windows エクスプローラを使用して、新しいライブラリに対応するディレクトリ全体（ファイル *FILEDIR.SAG* とサブディレクトリ *GP*、*RES*、および *ERR* を含む）を集合ディレクトリにコピーします。



Notes:

- 1 このディレクトリの場所がわからない場合は、Natural スタジオでシステムコマンド `SYSPROF` を実行します。表示されたダイアログボックスの **[Files in File System]** タブに、システムファイル `FUSER` に作成されたディレクトリへのパスが表示されます。新しいライブラリは、`FUSER` ディレクトリのサブディレクトリになり、Natural スタジオで定義された名前と同じ名前が付きます。
- 2 コンフィグレーションユーティリティを使用して、`FUSER` ディレクトリへのパスを検索することもできます。前の手順で作成したパラメータファイル（つまり、アプリケーションをランタイム環境で開始するのに使用されるパラメータファイル）を選択し、**[Natural 実行コンフィグレーション]** ノードを展開し、**[システムファイル]** ノードを選択します。`FUSER` ディレクトリへのパスが **[FUSER]** タブに表示されます。
- 4 必要に応じて、集合ディレクトリにコピーしたディレクトリの名前を変更します。ランタイム環境で使用されるライブラリの名前を入力します。

▶手順 11.4. オブジェクトをアンロードするには

- 1 [ツール] メニューの [開発ツール] > [オブジェクトハンドラ] を選択して、オブジェクトハンドラを開始します。
- 2 アンロードウィザードを開始します。
- 3 アンロードウィザードの最初のダイアログボックスで、[**Unload objects into Natural work file(s)**] オプションボタンを選択します。
- 4 次のダイアログボックス（オプション設定を指定する画面）で、[**Unload file**] テキストボックスに Natural ワークファイルを定義します。このワークファイルは、前に作成した集合ディレクトリに保存されている必要があります。
- 5 アプリケーションが両方の環境で同じライブラリ名を使用する場合は、次のダイアログボックス（パラメータを指定する画面）には情報を指定しないでください。

ただし、アプリケーションが開発環境とランタイム環境で異なるライブラリ名を使用する場合は、[**Use global parameters**] オプションボタンを選択し、[**Set**] ボタンをクリックして、表示されたダイアログボックスで名前を設定します。

- 6 次のダイアログボックス（オブジェクトタイプを指定する画面）で、[**Natural library objects**] オプションボタンを選択します。
- 7 次のダイアログボックス（アンロードする Natural ライブラリオブジェクトを選択する画面）で、[**Details**] ボタンをクリックします。
- 8 表示されたダイアログボックスで、アプリケーションに含まれるすべてのカタログ化オブジェクト、リソース、およびエラーメッセージを指定します。ソースはアンロードしないでください。[**S/C-Kind**] ドロップダウンリストボックスから、[**Gp**] を選択します。
- 9 次のダイアログボックスに進み、オブジェクトをアンロードします。
- 10 アンロードを完了したら、集合ディレクトリに作成されたワークファイルを確認します。ワークファイルですべてのオブジェクトをスキャンするには、ロードウィザードを使用します。

送信媒体への集合ディレクトリのコピー

集合ディレクトリ内のすべてのファイルを移植する準備が整ったら、Windows エクスプローラを使用して、集合ディレクトリの内容をすべてのサブディレクトリを含めて送信媒体（CD など）にコピーします。

手順 2 : Natural ランタイムのインストール

Natural ランタイムをランタイムワークステーションにインストールします。詳細については、『インストール』ドキュメントを参照してください。



Note: Natural ランタイムワークステーションでアプリケーションを更新する場合は、この手順は必要ありません。

手順3 : ランタイムワークステーションへのアプリケーションのインストール

以下では次のトピックについて説明します。

- グローバルコンフィグレーションファイルのコピー
- Natural パラメータファイルのコピー
- オブジェクトのコピーまたはロード

グローバルコンフィグレーションファイルのコピー

Windows エクスプローラを使用して、必要な DBID が定義されているグローバルコンフィグレーションファイルを送信媒体からグローバルコンフィグレーションファイルが含まれるランタイムワークステーション上のディレクトリにコピーします。



Caution: 既存のグローバルコンフィグレーションファイルは上書きされます。



Note: グローバルコンフィグレーションファイルの場所がわからない場合は、ランタイムワークステーション上でコンフィグレーションユーティリティを呼び出し、[ローカルコンフィグレーションファイル] ノードを展開し、[インストール指定] ノードを選択します。すると、グローバルコンフィグレーションファイルの完全なパスおよび名前が表示されます。

Natural パラメータファイルのコピー

Windows エクスプローラを使用して、Natural パラメータファイルを送信媒体から Natural パラメータファイルが含まれるランタイムワークステーション上のディレクトリにコピーします。



Caution: 既存のパラメータファイルは上書きされます。



Note: パラメータファイルの場所がわからない場合は、上記の説明に従って、コンフィグレーションユーティリティの [インストール指定] ノードを選択します。すると、Natural パラメータファイルの位置が、[パラメータモジュールへのパス] として表示されます。

オブジェクトのコピーまたはロード

オブジェクトがパッケージされた方法に応じて（「[オブジェクトのコピーまたはアンロード](#)」を参照）、送信媒体には完全なライブラリか Natural ワークファイルのいずれかが含まれています。

Windows エクスプローラを使用して完全なライブラリがコピーされた場合は、送信媒体には Natural ライブラリ名の付いたディレクトリが含まれています。各ディレクトリは Natural ライブラリ構造を反映しており、ファイル FILEDIR.SAG とサブディレクトリ GP、RES、および ERR が含まれます。この場合は、以下で説明する手順に従って、ライブラリをコピーする必要があります。

オブジェクトハンドラを使用してオブジェクトが Natural ワークファイルにアンロードされた場合は、送信媒体にはこのワークファイルが含まれています。この場合は、以下で説明する手順に従って、オブジェクトハンドラを使用してオブジェクトをロードする必要があります。

▶手順 11.5. ライブラリをコピーするには

- Windows エクスプローラを使用して、ライブラリ（すべてのサブディレクトリを含む）を FUSER システムファイルのディレクトリにコピーします。



Note: このディレクトリの場所がわからない場合は、コンフィグレーションユーティリティを呼び出し、前の手順でコピーしたパラメータファイルを選択し、**[Natural 実行コンフィグレーション]** ノードを展開し、**[システムファイル]** ノードを選択します。FUSER ディレクトリへのパスが **[FUSER]** タブに表示されます。

▶手順 11.6. オブジェクトをロードするには

- 1 コンフィグレーションユーティリティを呼び出し、パラメータファイル NATPARM の FUSER 設定に前の手順でコピーしたパラメータファイルと同じ DBID および FNR が含まれていることを確認します。



Note: FUSER の DBID および FNR は、上記の手順で呼び出した **[FUSER]** タブに表示されます。

- 2 Windows の **[スタート]** メニューの **[すべてのプログラム]** > **[Software AG Natural Runtime n.n]** > **[Natural Runtime]** を選択します。

Or:

Windows のデスクトップで次のショートカットを使用します。



これにより、Natural ランタイムが標準パラメータファイル NATPARM で呼び出されます。

- 3 ライブラリ SYSOBJH にログオンします。
- 4 ライブラリ SYSOBJH のプログラム MENU を実行します。
オブジェクトハンドラウィンドウが表示されます。
- 5 ロードウィザードの最初のダイアログボックスで、**[Load objects from Natural work file(s)]** オプションボタンを選択します。
- 6 次のダイアログボックス（オプション設定を指定する画面）で、**[Load file]** テキストボックスに Natural ワークファイルを定義します。これは、送信媒体上のワークファイルである必要があります。

 **Note:** 新しいパラメータファイルの DBID および FNR が標準の NATPARM 設定と異なる場合は、次のダイアログボックス（パラメータを指定する画面）で、新しいパラメータファイルで使用される値を入力します。**[Use global parameters]** オプションボタンを選択し、**[Set]** ボタンをクリックします。表示されたダイアログボックスで、**[Load Target]** タブを選択し、**[Load FUSER]** グループボックスで DBID および FNR の対応する値を入力します。
- 7 次のダイアログボックス（オブジェクトタイプを指定する画面）で、**[Load all objects from work file]** オプションボタンを選択します。
- 8 次のダイアログボックスに進み、オブジェクトをロードします。
- 9 オブジェクトハンドラを終了し、Natural ランタイムを終了します。

手順4：ランタイムワークステーションでのアプリケーションの開始

必要なファイルがすべてランタイムワークステーションにコピーされたら、アプリケーションを開始できます。各アプリケーション用にショートカットを作成することをお勧めします。ショートカットには、アプリケーションを実行するのに必要なパラメータファイルの名前を定義できません。

ランタイムワークステーションでアプリケーションを開始する方法はいくつかあります。

■ *naturalr.exe* を使用する

naturalr.exe コマンドでパラメータファイル名を指定しなかったときに表示されるユーザーインターフェイスが、バックグラウンドで非表示で実行されます。次に例を示します。

```
"C:\Program Files\Software AG\Natural Runtime\n.n.\bin\naturalr.exe" PARM=file-name
```

n.n は現在のバージョン番号を表し、*file-name* はカスタマイズされた **Natural** パラメータファイルに割り当てた名前（ファイル拡張子なし）を表します。

ユーザーインターフェイスが開始されたアプリケーションでこのランタイムプロセスが正常に終了しなかった場合にのみ、ユーザーインターフェイスは表示されます（例えば、アプリケーションが TERMINATE ステートメントを発行しなかった場合）。このユーザーインターフェイスは、表示されない場合でも、システムリソースを消費します。

■ *natrt.exe* を使用する

このいわゆる「ミニランタイム」には、ユーザーが実行するアプリケーションを選択できるユーザーインターフェイスが含まれていません。この場合は、開始するプログラムの名前をパラメータファイルで定義する必要があります。次に例を示します。

```
"C:\Program Files\Software AG\Natural Runtime\n.n.\bin\natrt.exe" PARM=file-name
```

n.n は現在のバージョン番号を表し、*file-name* はカスタマイズされた **Natural** パラメータファイルに割り当てた名前（ファイル拡張子なし）を表します。

ミニランタイムは、パラメータファイル内のすべてのコマンドが処理されるとすぐに終了します。

プログラム名がパラメータファイルに指定されていない場合は、ミニランタイムは直ちに終了します。

ミニランタイムを使用すると、システムリソースの消費が抑えられ、アプリケーション処理の最後にランタイムプロセスを確実に終了できます。

Natural ランタイム起動サービスの使用

Natural ランタイム起動サービスがインストールされアクティブになっている場合は、PC の起動時に 1 つまたは複数の Natural ランタイムプロセスを自動的に開始できます。

Natural パラメータを保持するパラメータテンプレートを定義できます。これにより、テンプレートに定義されているすべてのパラメータを使用して Natural ランタイムプロセスを開始することが可能です。

Natural ランタイムプロセスは通常、Natural アプリケーションを実行するのに使用されます。この目的では、Natural ランタイムプロセスは、STACK パラメータが次のように定義されているテンプレートで開始される必要があります。

```
STACK=(LOGON library-name; program-name)
```

STACK パラメータが定義されていない場合は、Natural はアプリケーションを実行せずに起動されます。

デフォルトでは、Natural ランタイム起動サービスはインストールされていません。これは、以下で説明する手順に従ってインストールする必要があります。

以下では次のトピックについて説明します。

- Natural ランタイム起動サービスコマンド
- 例：Natural プロセスの自動的な開始

Natural ランタイム起動サービスコマンド

Natural ランタイムの *bin* ディレクトリに保存されている *natrtsvc.exe* ファイルは、サービスコマンドを実行するのに使用されます。

次のサービスコマンドは、Windows のコマンドプロンプトウィンドウで指定できます。

| コマンド | 説明 | | | | |
|-------------------------------------|---|---------------|---|------------------|----------------------------------|
| NATRTSVC INSTALL <i>mode</i> | Natural ランタイム起動サービスをインストールします。 <i>mode</i> は、次のいずれかです。 <table border="1" data-bbox="570 1209 1472 1476"> <tr> <td><i>manual</i></td> <td>デフォルトです。サービスがインストールされ、手動で開始する必要があります (START コマンドを使用するか、Windows で Software AG Natural Runtime n.n Startup Service を開始することによって行います)。</td> </tr> <tr> <td><i>automatic</i></td> <td>サービスがインストールされ、PC の起動時に自動的に起動します。</td> </tr> </table> | <i>manual</i> | デフォルトです。サービスがインストールされ、手動で開始する必要があります (START コマンドを使用するか、Windows で Software AG Natural Runtime n.n Startup Service を開始することによって行います)。 | <i>automatic</i> | サービスがインストールされ、PC の起動時に自動的に起動します。 |
| <i>manual</i> | デフォルトです。サービスがインストールされ、手動で開始する必要があります (START コマンドを使用するか、Windows で Software AG Natural Runtime n.n Startup Service を開始することによって行います)。 | | | | |
| <i>automatic</i> | サービスがインストールされ、PC の起動時に自動的に起動します。 | | | | |
| NATRTSVC REMOVE | Natural ランタイム起動サービスをシステムから削除します。 | | | | |
| NATRTSVC START | Natural ランタイム起動サービスを、開始されていない場合に開始します。サービスは、以前に作成されて <i>start</i> パラメータが "yes" に設定されているパラメータテンプレートを検索します。さらに、同じテンプレートに保存されている Natural パラメータで Natural ランタイムプロセスを開始します。 | | | | |
| NATRTSVC START <i>template-name</i> | 指定されたテンプレートに保存されている Natural パラメータで Natural ランタイムプロセスを開始します。Natural ランタイム起動サービスが (PC の起動時に自動的に、またはユーザーによって手動で) 開始されていなかった場合は、エラーメッセージが表示されます。 | | | | |
| NATRTSVC STOP | Natural ランタイム起動サービスおよびそれによって開始されたすべての Natural ランタイムプロセスを停止します。 | | | | |

| コマンド | 説明 | | | | |
|---|---|-----|---------------------------|----|-------------------------------------|
| NATRTSVC STOP <i>template-name</i> | Natural ランタイム起動サービスによって開始された Natural ランタイムプロセスを、指定されたテンプレートに保存されている Natural パラメータで停止します。 | | | | |
| NATRTSVC CREATE <i>template-name</i> | Natural ランタイム起動サービスによって開始される新しいパラメータテンプレートを作成します。 | | | | |
| NATRTSVC DELETE <i>template-name</i> | 指定されたテンプレートを Natural ランタイム起動サービスから削除します。 | | | | |
| NATRTSVC SET <i>template-name</i> start= <i>mode</i> | Natural ランタイム起動サービスの開始時に、指定されたテンプレートに保存されている Natural パラメータで Natural ランタイムプロセスを開始するかどうかを定義します。 <i>mode</i> は、次のいずれかです。 <table border="1" data-bbox="474 613 1385 793"> <tr> <td>yes</td> <td>Natural ランタイムプロセスが開始されます。</td> </tr> <tr> <td>no</td> <td>デフォルトです。 Natural ランタイムプロセスは開始されません。</td> </tr> </table> | yes | Natural ランタイムプロセスが開始されます。 | no | デフォルトです。 Natural ランタイムプロセスは開始されません。 |
| yes | Natural ランタイムプロセスが開始されます。 | | | | |
| no | デフォルトです。 Natural ランタイムプロセスは開始されません。 | | | | |
| NATRTSVC SET <i>template-name</i> <i>Natural-parameters</i> | Natural パラメータを指定されたテンプレートに保存します。有効な Natural パラメータについては、『パラメータリファレンス』を参照してください。複数のパラメータを指定する場合は、パラメータを空白で区切る必要があります。パラメータの代わりに、Natural パラメータファイル名を指定することもできます。 | | | | |
| NATRTSVC SHOW | すべてのテンプレートのスタートアップ設定および保存されている Natural パラメータを表示します。 | | | | |
| NATRTSVC SHOW <i>template-name</i> | 指定されたテンプレートのスタートアップ設定および保存されている Natural パラメータを表示します。 | | | | |
| NATRTSVC STATUS | すべてのテンプレートのステータスを表示して、これらのテンプレートがアクティブか非アクティブかを示します。 | | | | |
| NATRTSVC STATUS <i>template-name</i> | 指定されたテンプレートのステータスを表示して、このテンプレートがアクティブか非アクティブかを示します。 | | | | |

例：Natural プロセスの自動的な開始

この例では、Natural ランタイム起動サービスを開始し、新しいテンプレートを作成し、PC を起動するたびに対応する Natural プロセスを開始する方法について説明します。

▶手順 11.7. PC の起動時に Natural プロセスを開始するには

- 1 Windows のコマンドプロンプトウィンドウを起動します。
- 2 *natrtsvc.exe* ファイルが保存されている Natural *bin* ディレクトリに移動します。
- 3 次のコマンドを入力して、Natural ランタイム起動サービスをインストールします。

```
NATRTSVC INSTALL automatic
```

次の情報が表示されます。

```
%NATRTSVC-I: Natural Runtime n.n Startup Service
%NATRTSVC-I: Natural Runtime n.n Startup Service successfully installed
%NATRTSVC-I: Path of binary is C:\PROGRAM FILES\SOFTWARE AG\NATURAL
RUNTIME\n.n\BIN\NATRTSVC.EXE
%NATRTSVC-I: Startup mode of Natural Runtime n.n Startup Service is 'Automatic'
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

これ以降は、PC を起動するたびに Natural ランタイム起動サービスが自動的に開始されます。

- 4 次のコマンドを入力して、空のパラメータテンプレートを "exa_temp" という名前で作成します。

```
NATRTSVC CREATE exa_temp
```

次の情報が表示されます。

```
%NATRTSVC-I: Natural Runtime n.n Startup Service
%NATRTSVC-I: New Natural instance 'exa_temp' created
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

- 5 次のコマンドを入力して、Natural ランタイム起動サービスの開始時に、パラメータテンプレート "exa_temp" に保存されている Natural パラメータで Natural ランタイムプロセスを開始することを定義します。

```
NATRTSVC SET exa_temp start=yes
```

次の情報が表示されます。

```
%NATRTSVC-I: Natural Runtime n.n Startup Service
%NATRTSVC-I: Configuration successfully set
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

- 6 次のコマンドを入力して、パラメータテンプレート "exa_temp" に Natural パラメータファイル "myparm" の内容を保存します。

```
NATRTSVC SET exa_temp parm=myparm
```

次の情報が表示されます。

```
%NATRTSVC-I: Natural Runtime n.n Startup Service  
%NATRTSVC-I: Configuration successfully set  
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

- 7 PC を再起動します。

Natural ランタイム起動サービスの自動開始モードを定義したので、Windows の起動後、指定した Natural ランタイムプロセスが自動的に開始されます。

- 8 次のコマンドを入力して、現在定義されているすべてのパラメータテンプレートのステータスを表示します。

```
NATRTSVC STATUS
```

次の情報が表示されます。

```
%NATRTSVC-I: Natural Runtime n.n Startup Service  
%NATRTSVC-I: Send request to Natural Runtime n.n Startup Service  
%NATRTSVC-I:    exa_temp is active  
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

12 NATCONV.INI での異なる文字セットのサポート

- 異なる文字セットのサポートの重要性 114
- 異なる文字セットの使用方法 114

コンフィグレーションファイル NATCONV.INI の設定値が A フォーマットに適用されます。U フォーマットでは、ICU ライブラリを使用します。

このchapterでは、Naturalでの異なる文字セットのサポート方法について、次のトピックで説明します。

異なる文字セットのサポートの重要性

異なる文字セットによる複数の言語のサポートは、国際化対応へのNaturalのアプローチを示します。これは、以下を使用する場合に役立ちます。

- 言語固有文字の大文字／小文字変換
- Natural 識別子、オブジェクト名、およびライブラリ名の言語固有文字
- マスク定義と比較されるオペランドの言語固有文字（『プログラミングガイド』の「MASK オプション」を参照）

異なる文字セットの使用方法

言語固有文字をサポートするために Natural が使用するすべてのチェック、変換、および分類テーブルは、コンフィグレーションファイル NATCONV.INI にあります。デフォルトでは、このファイルは Natural の *etc* ディレクトリにあります。

ローカルまたはアプリケーション固有の文字セットをサポートするように、NATCONV.INI を修正できます。

標準アプリケーションでは、NATCONV.INI の修正によって重大な矛盾が発生する可能性があるため、すでに Natural オブジェクトとデータベースデータが存在する場合は特に、このファイルを修正しないでください。また、その必要もありません。

NATCONV.INI の修正は十分に考慮し、慎重に実行してください。不用意な修正は、修復不可能な問題を引き起こすことがあります。

NATCONV.INI は、セクションとサブセクションに分かれます。次のセクションが定義されています。

| セクション | 説明 |
|--------------------------|---|
| CHARACTERSET-DEFINITION | <p>このセクションでは、内部文字セットの名前が定義されます。デフォルトは "ISO8859_1" です。</p> <p>異なる文字セットを選択する場合は、その文字セットのサブセクションが以下のセクションに含まれている必要があります。</p> |
| CASE-TRANSLATION | <p>このセクションには、以下のいずれかが指定されている場合に実行される大文字から小文字への変換に必要なテーブルが含まれています。</p> <ul style="list-style-type: none"> ■ 端末コマンド %U ■ フィールド属性 AD=T ■ ステートメント EXAMINE TRANSLATE <p>この変換は内部文字セット内で行われます。例えば、内部文字セットが "ISO8859_5" の場合は、このセクションに以下の2つのサブセクションが含まれている必要があります。</p> <ul style="list-style-type: none"> ■ [ISO8859_5->UPPER] ■ [ISO8859_5->LOWER] |
| IDENTIFIER-VALIDATION | <p>このセクションには、識別子（ソースプログラム内のユーザー定義変数）、オブジェクト名、およびライブラリ名の検証に必要なテーブルが含まれています。これには、各定義済み内部文字セットのサブセクションが含まれています。</p> <p>このセクションでは、特殊文字の "#"（非データベース変数用）、"+"（アプリケーション非依存変数用）、"@（SQL と Adabas の空値/長さインジケータ用）、および "&"（ダイナミックソース生成用）を再定義できます。さらに、識別子、オブジェクト名、およびライブラリ名に有効な 1 番目および後続の文字のセットを修正できます。</p> <p>注意: オブジェクト名に有効な文字のセットを "x7f"（小数点 127）を超える値で拡張する場合は、オブジェクトのソート順（例えば、LIST * コマンド時）が番号順にならないことがあります。</p> |
| CHARACTER-CLASSIFICATION | <p>このセクションには、MASK オプションを評価する場合などに使用される文字の分類に必要なテーブルが含まれています。これには、各定義済み内部文字セットのサブセクションが含まれています。</p> |

セクション CHARACTERSET-DEFINITION と各サブセクションには、文字の変換方法および関連する属性を記述する行が含まれています。これらの行は次のように表示されます。

```

line      ::= key = value
key       ::= name_key | range_key
name_key  ::= keyword{ CHARS }
keyword   ::= INTERNAL-CHARACTERSET | NON-DB-VARI | DYNAMIC-SOURCE |
            GLOBAL-VARI | FIRST-CHAR | SUBSEQUENT-CHAR |
            LIB-FIRST-CHAR | LIB-SUBSEQUENT-CHAR | ALTERNATE-CARET |
            ISASCII | ISALPHA | ISALNUM | ISDIGIT | ISXDIGIT |
            ISLOWER | ISUPPER | ISCNTRL | ISPRINT | ISPUNCT |
            ISGRAPH | ISSPACE
range_key ::= hexnum | hexnum-hexnum
value     ::= val { , val }
val       ::= hexnum | hexnum-hexnum
hexnum    ::= xhexdigithexdigit | xhexdigithexdigit
    
```

 **Notes:**

1. 左辺に range_key 変数が指定された場合は、右辺に指定される値の数は、右辺に指定される値が 1 つだけ（キー範囲の各要素に割り当てられる）の場合を除いて、キー範囲に指定された値の数に一致している必要があります。
2. 左辺に name_key 変数が指定され、対応する文字コードのリストが 1 行に収まらない場合は、"name_key=" を再度指定することによって次の行に継続できます。行を空白またはタブで開始しないでください。

有効な行の例

| | |
|---|---|
| x00-x1f = x00 | "x00"~"x1f" のすべての文字が "x00" に変換されます。 |
| x00-x7f = x00-x7f | "x00"~"x7f" のすべての文字が変換されません。 |
| x00-x08 = x00,x01-x07,x00 | 文字 "x00" と "x08" が "x00" に変換され、"x01"~"x07" の文字が変換されません。 |
| ISALPHA = x41-x5a,x61-x7a,xc0-xd6,xd8 ISALPHA = xd9-xf6,xf8-xff | これらの 2 行に指定されたすべての文字に属性 ISALPHA が割り当てられます。 |

無効な行の例

| | |
|-------------------|---|
| x41 = 'A' | すべての文字は 16 進形式で指定する必要があります。 |
| 0x00-0x1f = 0x00 | 16 進値は以下のいずれかの形式で指定する必要があります。 <i>xdigitdigit</i> <i>Xdigitdigit</i> |
| x00-x0f = x00,x01 | 指定された値の数がキー範囲の要素の数と一致していません。 |

13 Natural 出口コード

- Natural スタートアップエラー 120

Natural 出口コードには、次の2つのタイプがあります。

- **スタートアップエラー**。出口コード0および1は成功を示し、その他のすべての出口コードはエラーを示します。
- TRMINATE ステートメントによって生成されたエラー。この場合の出口コードは0~255です。

Natural スタートアップエラー

以下の出口コードが Natural スタジオの開始時に発生する場合があります。

| | |
|-------|--|
| 2 | SAGtermcap または環境変数 NATTCHARSET に端末制御文字 (TCS) 機能が指定されました。 |
| 3 | 文字変換テーブルの初期化に失敗しました。 |
| 4 | 文字変換ファイル NATCONV.INI でエラーが発生しました。 |
| 5 | グローバルコンフィグレーションファイル NATCONF.CFG からデータベース割り当てを読み取ることができません。 |
| 6 | FNAT(<i>dbid</i> , <i>fnr</i>) または FUSER(<i>dbid</i> , <i>fnr</i>) が見つかりません。コンフィグレーションファイルを確認してください。 |
| 7 | LFILE テーブルを初期化できません。 |
| 8 | 現在は使用されていません。 |
| 9 | 現在は使用されていません。 |
| 10 | 現在は使用されていません。 |
| 11 | 現在は使用されていません。 |
| 12 | 指定されたパラメータファイルを読み込むことができません。パラメータファイルを確認してください。 |
| 13 | パラメータファイル NATPARM を読み込むことができません。 |
| 14 | ストレージマネージャの初期化に失敗しました。 |
| 15 | STDIN からの読み込み中にファイルの終わり (EOF) に到達しました。 |
| 16 | バッファプールを開くことができません。Natural システム管理者に連絡してください。 |
| 17 | NATURAL.INI ファイルからバッファプール割り当てを読み取ることができません。 |
| 18 | FDIC 割り当てが無効です。 |
| 19 | FNAT 割り当てが無効です。 |
| 20 | FSEC 割り当てが無効です。 |
| 21 | FUSER 割り当てが無効です。 |
| 22 | Natural ログインモジュールをロードできません。 |
| 23 | ローカルデータ用のメモリを割り当てることができません。USIZE または SSIZE パラメータを減らしてください。 |
| 24 | Natural 表示モジュールをロードできません。 |
| 25,26 | 共有可能なイメージまたは DLL のロード中にエラーが発生しました。 |

| | |
|----|--|
| 27 | ログインがキャンセルされました。Natural が終了します。 |
| 28 | Natural 起動時のセキュリティ違反。Natural が終了します。 |
| 29 | Natural 起動時のセキュリティ違反。ログイン失敗が多すぎるため、ログインは中止されました。 |
| 30 | Natural システムエラーメッセージが発行されました。 |
| 31 | NAT0866：Natural ニュークリアスが Natural Security ニュークリアスではありません。 |
| 32 | パスワードチェックに失敗しました。 |
| 33 | ロックマネージャがセマフォを作成または初期化できません。 |
| 34 | ライブラリにアクセスできないか、指定された FNAT/FUSER にライブラリが存在しません。FNAT と FUSER のシステムファイル割り当てとファイル属性（ディレクトリとファイル）を確認してください。 |
| 35 | 内部 wfc i/o 端末ドライバエラー。 |
| 36 | 内部 XVT エラー。 |
| 37 | DCOM スタートアップエラー。 |
| 38 | ランタイムコンテキストの作成に失敗しました。 |
| 39 | NATDIR または NATVERS 環境変数が見つかりません。NATDIR 環境変数が設定されている場合は、無効な文字や空白文字が含まれてないことを確認してください。NATVERS には、Natural バージョンのみが含まれている必要があります。パスには、有効なドライブ ID が含まれている必要があります。 |
| 40 | Natural zmodem エラー。 |
| 41 | TF テーブルの作成に失敗しました。古いパラメータモジュールの複数のデータベースタイプのエントリがあります。Natural コンフィグレーションユーティリティでパラメータモジュールを確認してください。 |
| 42 | バッチモードドライバエラー。 |
| 43 | 画面ウィンドウサイズが小さすぎます。 |
| 44 | SQL シグナルハンドラを終了します。 |
| 45 | アドオン製品をロードできません。 |
| 46 | FNAT ライブラリ SYSLIB にアクセスできません。特権が不適切であるか、ファイル保護違反です。 |
| 47 | NATURAL.INI ファイルから PARM_PATH エントリを読み取ることができないか、ディレクトリにアクセスできません。 |
| 48 | NATURAL.INI ファイルから CONFIG_NAME エントリを読み取ることができないか、ファイルにアクセスできません。 |
| 50 | NATURAL.INI ファイルから NATCONV エントリを読み取ることができないか、ファイルにアクセスできません。 |
| 51 | NATURAL.INI ファイルの TMP_PATH エントリを処理できません。パス 'path' にアクセスできません。 |
| 52 | NATURAL.INI ファイルから PROFILE_PATH エントリを読み取ることができないか、ディレクトリにアクセスできません。 |
| 53 | ローカルコンフィグレーションファイル NATURAL.INI を開くことができません。 |
| 54 | NATCONF.CFG で NATOSDEP を読み取ることができません。 |
| 55 | NATURAL.INI で NATEXTLIB を読み取ることができません。 |

| | |
|----|---|
| 56 | SAG.INI ファイルの NATDIR エントリを読み取ることができません。 |
| 57 | 未使用。 |
| 58 | SAG.INI ファイルの NATINI エントリを読み取ることができません。 |
| 59 | 認識されていないオプション ' <i>option</i> ' が指定されました。 |
| 60 | 内部テーブルを初期化するのに十分なメモリがありません。 |
| 61 | バッチエラーが発生しましたが、CC=ON パラメータにより処理は続行されました。 |
| 62 | アクティブなりポジトリを持つ複数の Natural セッションは許可されていません。 |
| 63 | アクティブなりポジトリを持つ Natural セッションはすでに実行されています。 |
| 64 | FNAT の LIBDIR.SAG を開くことができませんでした。ファイルの有無とアクセス保護を確認してください。 |
| 66 | ポート番号が不明です。 |
| 67 | 指定されたポート番号はすでに使用されています。 |
| 68 | 構文 ... が無効です。 |
| 69 | Pal パッケージの初期化に失敗しました。 |
| 70 | ポート番号が無効です。 |
| 71 | 指定されたポート番号での待機に失敗しました。 |
| 72 | これは Natural ... の評価版です。... まで有効です。 |
| 73 | この Natural ... 評価版の試用期間の期限が切れました。有効期限は、... まででした。 |
| 74 | ... の起動中に ... Natural エラーメッセージ ' <i>nnnn</i> ' を受け取りました。 |
| 75 | ポート番号が指定されていません。 |
| 76 | RPC バージョンが正しくありません。 |
| 77 | FDDM 割り当てが無効です。 |
| 78 | この Natural ... 評価版の試用期間の期限が切れました。有効期限は、... まででした。 |
| 79 | ポート番号 ... が上限 (99999) を超えました。 |
| 85 | コンテキスト初期化中に Natural ランタイムスタートアップエラーが発生しました。 |
| 86 | 無効なコードページ [<i>name</i>] が指定されました。 |
| 88 | ダイナミックパラメータ文字列の長さが上限を ... 文字超えています。 |



Note: リターンコードを受け取るには、*nderun.exe* を (*naturalr.exe* ではなく) 実行する必要があります。

14 Entire System Server インターフェイスの設定

| | |
|---|-----|
| ▪ 前提条件 | 124 |
| ▪ 有効化 | 124 |
| ▪ Entire System Server DDM のデータベース ID の変更 | 125 |

Entire System Server 製品を使用する場合は、Entire System Server インターフェイスが必要です。Entire System Server インターフェイスは Natural の一部であり、追加のインストールは必要ありません。

また、Natural にはライブラリ SYSNPE および SYSNPR が用意されています。

SYSNPE は、Entire System Server オンラインチュートリアルで、Entire System Server ユーザーのためのスタートヘルプとして提供されます。Entire System Server の詳細については、Entire System Server のドキュメントを参照してください。

ライブラリ SYSNPR には、Entire System Server DDM のデータベース ID を変更するために使用されるプログラム CHANGEDB が含まれています。

このchapterでは、次のトピックについて説明します。

前提条件

Entire System Server インターフェイスは、z/OS、z/VSE、および BS2000/OSD 上の Entire System Server へのアクセスを Entire Net-Work 経由で可能にします。Entire System Server インターフェイスを完全にサポートするには、メインフレームプラットフォームに Entire Net-Work バージョン 5.8.1 以降がインストールされている必要があります。

有効化

標準 Natural コンフィグレーション設定を使用している場合は、Entire System Server インターフェイスはアクティブになりません。Entire System Server インターフェイスのデータベースの値 (Natural プロファイルパラメータ ESXDB) がデフォルトで "0" に設定されます。Entire System Server インターフェイスを使用するには、コンフィグレーションユーティリティを使用して、パラメータ ESXDB の値を "148" に設定する必要があります。

コンフィグレーションユーティリティでは、パラメータ ESXDB はパラメータファイルのパラメータグループ [プロダクトコンフィグレーション (**Product Configuration**)] に割り当てられています。



Tip: このパラメータを見つけるには、"ESXDB" で検索します。詳細については、『コンフィグレーションユーティリティ』ドキュメントの「パラメータの検索」を参照してください。

ESXDB は、Entire System Server の DDM に使用されるデータベース ID を指定します。この DBID は、Entire System Server 要求のターゲット DBID を指定するのではなく、カタログされた Entire System Server DDM に使用される DBID を Natural に通知します。有効な Entire System Server ターゲット DBID は、すべての Entire System Server DDM の一部である NODE フィールドで指定されます。



Important: Entire System Server インターフェイスサポートで Natural を実行するために、ESXDB の値を "148" に変更します。すべての Entire System Server DDM は DBID 148 でカタログされます。

Natural を再起動すると、メインフレームで実行されている Entire System Server ノードに Entire Net-Work 経由でアクセスできます。

Entire System Server インターフェイスのカスタマイズは、Entire System Server DDM の修正のみ可能です。

Entire System Server DDM のデータベース ID の変更

ライブラリ SYSNPR には、すべての Entire System Server DDM のデータベース ID を変更するために使用するプログラム CHANGEDB が含まれています。すべての Entire System Server DDM はライブラリ SYSNPE にあります。新しい DBID 値としてプログラム CHANGEDB に入力したデータベース ID は、コンフィグレーションユーティリティでも Entire System Server インターフェイスのデータベースパラメータ (ESXDB) の値として指定する必要があります。

15 Web I/O インターフェイスサービスの設定

| | |
|-------------------------------------|-----|
| ■ Web I/O インターフェイスサービスのインストール | 128 |
| ■ Web I/O インターフェイスサービスの設定 | 129 |
| ■ イベントログ | 131 |
| ■ Natural を開始するためのバッチファイル | 132 |

Web I/O インターフェイスは、Natural アプリケーションを Web ブラウザで実行するために使
用します。次の場合は、Natural アプリケーションを Web ブラウザで実行する必要があります。

- アプリケーションがメインフレームまたは UNIX 上で実行され、Unicode データを含む文字
マップを使用する場合



Note: GUI コントロールを含むマップはサポートされません。

- アプリケーションが Natural for Ajax ページを使用する場合

Web I/O インターフェイスは、2つのコンポーネントで構成されています。Web I/O インター
フェイスクライアントと Web I/O インターフェイスサービスです。Natural アプリケーション
を Web ブラウザで実行するユーザーは、特定の URL を Web I/O インターフェイスクライアント
に送信します。Web I/O インターフェイスクライアントは、ユーザーに代わって、Web I/O
インターフェイスサービスに Natural セッションを作成するように要求します。その後、Natural
セッションの出力が Web ページとして Web ブラウザに送信されます。

Web I/O インターフェイスクライアントは、Web サーバーまたはアプリケーションサーバーに
インストールします。ネットワーク内の任意のマシンにインストールできます。Web I/O イン
ターフェイスクライアントのインストールおよび設定は、「[Web I/O インターフェイスクライアントの設定](#)」
で説明します。

Web I/O インターフェイスサービスは、Natural がインストールされているマシンにインストー
ルします。Web I/O インターフェイスサービスのインストールおよび設定は、このchapterで説
明します。

このchapterでは、次のトピックについて説明します。

Web I/O インターフェイスサービスのインストール

Natural for Windows のインストール時に該当のオプションを設定した場合は、Web I/O イン
ターフェイスサービスもインストールされます。これは、セットアップタイプ [**Development
Client for Single Point of Development (SPoD)**] の一部です。詳細については、『インストー
ル』ドキュメントを参照してください。

Web I/O インターフェイスサービスの設定

Web I/O インターフェイスサービスは、Windows サービスとしてインストールされます。Windows サービス管理コンソールに **[Software AG Natural n.n Web I/O Service]** とリストされます。このサービスは、Windows の起動時に自動的に開始されます。デフォルトでは、このサービスはポート番号 2900 で待機します。インストール時に他のポート番号を選択できます。

Web I/O インターフェイスサービスの設定は、Windows レジストリに保存されます。Web I/O インターフェイスサービスがインストールされたときのデフォルト値を使用しない場合は、必要に応じてサービスの設定を変更できます。これについては、以降のトピックを参照してください。

- [Web I/O インターフェイスサービスコマンド](#)
- [例：独自の設定を使用した Web I/O インターフェイスサービスの開始](#)

Web I/O インターフェイスサービスコマンド

Natural *bin* ディレクトリに保存されているプログラム *nwosvcd.exe* は、サービスコマンドを実行するのに使用されます。次のサービスコマンドは、Windows のコマンドプロンプトウィンドウで指定できます。

| コマンド | 説明 |
|-------------------------------|--|
| NWOSVCD -INSTALL | Web I/O インターフェイスサービスをインストールします。サービスはスタートアップの種類が [自動] でインストールされます。 スタートアップの種類を変更する場合は、Windows サービス管理コンソールを開く必要があります。Windows XP では、これは [管理ツール] にあります。 Software AG Natural n.n Web I/O Service のプロパティを開き、目的のスタートアップの種類を選択します。 |
| NWOSVCD -REMOVE | Web I/O インターフェイスサービスをシステムから削除します。 |
| NWOSVCD -CONSOLE | Web I/O インターフェイスサービスを Windows コンソールアプリケーションとして実行します。この場合は、アイコンがタスクバーの通知領域に表示されます。このアイコンのコンテキストメニューを表示すると、以下のコマンドを使用できます。 ■ バージョン情報 Web I/O インターフェイスサービスのバージョンが表示されます。 ■ 終了 Web I/O インターフェイスサービスを停止するかどうかを確認するダイアログボックスが表示されます。 |
| NWOSVCD -CONFIG -PORT=nnnn | Web I/O インターフェイスサービスの設定を変更します。以下のキーを指定できます。 |

| コマンド | 説明 |
|--|--|
| <p>-USEREXIT1=<i>pathname</i> -USEREXIT2=<i>pathname</i></p> | <p>-PORT=<i>nnnn</i> ポート番号。</p> <p>-USEREXIT1=<i>pathname</i> USEREXIT1によって定義されたライブラリには、以下の関数が含まれている必要があります。</p> <pre>int nwo_CheckUsernameAndPassword(const char *pUsername, const char *pPassword, const char *pNewPassword, char *pErrorMessage, HANDLE *pHandleUser)</pre> <p>キー USEREXIT1 が設定されている場合は、関数 <code>nwo_CheckUsernameAndPassword</code> がユーザー名とパスワードをチェックする役割を果たします。新しいパスワードを受け取った場合は、ユーザー出口1もパスワード変更の役割を果たします。</p> <p>エラーの場合は、関数のリターンコードは"0"である必要があります。この場合は、<code>pErrorMessage</code> がクライアントに返されます。ユーザー名およびパスワードが正しい場合は、リターンコードの値は"0"以外である必要があります。</p> <p>-USEREXIT2=<i>pathname</i> USEREXIT2によって定義されたライブラリには、以下の関数が含まれている必要があります。</p> <p>■ <code>int nwo_Messages(int *iNumberOfMessages, char *pMessage[])</code></p> <p>キー USEREXIT2 が設定されている場合は、新しいクライアントが受け入れられたときに関数 <code>nwo_Messages</code> が呼び出され、この関数が返したメッセージがクライアントに送信されます。ユーザー出口2を使用して、次のようなメッセージを送信できます。「メンテナンスのため、Natural アプリケーション XXXXX は、次の月曜日の 18:00 から 19:00 まで使用できません。」</p> <p>エラーの場合は、関数のリターンコードは "0" である必要があります。</p> <p>関数 <code>nwo_Messages</code> が呼び出された後、関数 <code>nwo_FreeMessages</code> が呼び出されます。</p> <p>■ <code>int nwo_FreeMessages(int iNumberOfMessages, char *pMessage[])</code></p> <p><code>iNumberOfMessages</code> : メッセージの数です。</p> <p><code>pMessage</code> : メッセージの配列です。</p> <p>キー USEREXIT2 が定義されている場合は、関数 <code>nwo_Messages</code> で割り当てられたリソース（通常はメモリ）を解放するために、関数 <code>nwo_FreeMessages</code> が呼び出されます。</p> <p>エラーの場合は、関数のリターンコードは "0" である必要があります。</p> <p>設定からユーザー出口を削除するには、以下のコマンドを入力します。</p> |

| コマンド | 説明 |
|------|--|
| | NWOSVCD -CONFIG -USEREXIT1= 設定を変更したら、Web I/O インターフェイスサービスを再起動する必要があります。 |

例：独自の設定を使用した Web I/O インターフェイスサービスの開始

この例では、ポート番号 3344 とユーザー出口 1 C:\Program Files\My Company Name\User Exit1\libuserexit1.dll で新しい設定を作成する方法について説明します。

▶手順 15.1. 独自の設定で Web I/O インターフェイスサービスを開始するには

- 1 Windows サービス管理コンソールを開き、Web I/O インターフェイスサービスを停止します。
- 2 Windows のコマンドプロンプトウィンドウを起動します。
- 3 *nwosvcd.exe* ファイルが保存されている *Natural bin* ディレクトリに移動します。
- 4 次のコマンドを入力して、新しいポート番号と新しいユーザー出口 1 を設定します。

```
NWOSVCD -CONFIG -PORT=3344 -USEREXIT1="C:\Program Files\My Company Name\User Exit1\libuserexit1.dll"
```

- 5 Windows サービス管理コンソールに戻り、Web I/O インターフェイスサービスを開始します。

イベントログ

Web I/O インターフェイスサービスは、情報、警告、およびエラーイベントを Windows イベントビューアに保存します。

イベントの例：

- Natural *n.n* Web IO Service が正常に開始されました。
- Natural *n.n* Web IO Service が正常に停止されました。

エラーの状況については、Windows イベントビューアに保存されている Web I/O インターフェイスサービス関連のイベントを確認してください。

Natural を開始するためのバッチファイル

Natural セッションを開始するために、Web I/O インターフェイスサービスはバッチファイルを実行します。バッチファイルは、Natural セッション用に環境を準備し、最終的に Natural を開始します。

バッチファイルは、Web I/O インターフェイスクライアントから特定のパラメータを受け取ります。パラメータは、バッチファイル自体で評価されるか、Natural に渡されます。Natural セッションを開始するクライアントは、使用するバッチファイルを指定できます。ファイルの場所と指定方法については、「[コンフィグレーションファイルでの Natural セッションの定義](#)」を参照してください。

nwo.bat という名前のサンプルバッチファイルが、*Naturalbin* ディレクトリにあります。次の内容が含まれています。

```
@echo off
set IPAddress=""
set ClientID=""
set Argument3=""
set Argument4=""
set NaturalParameters=""

IF %1 NEQ "null" goto IPAddress
:IPAddress
set IPAddress=%~1

IF %2 NEQ "null" goto ClientID
:ClientID
set ClientID=%~2

IF %3 NEQ "null" goto Argument3
:Argument3
set Argument3=%~3

IF %4 NEQ "null" goto Argument4
:Argument4
set Argument4=%~4

IF %5 NEQ "null" goto NaturalParameters
:NaturalParameters
set NaturalParameters=%~5

rem echo %IPAddress%
rem echo %ClientID%
rem echo %Argument3%
rem echo %Argument4%
rem echo %NaturalParameters%
```

```
natrt.exe %NaturalParameters% webio=on :
```

Naturalセッションを開始する独自のバッチファイルを作成できます。この場合は、上記のサンプルバッチファイルのように、*natrt.exe* で Natural を開始し、パラメータ *WEBIO=ON* を使用する必要があります。上記のサンプルでは、バッチファイルが *Natural bin* ディレクトリに含まれている必要があります。含まれていない場合は、*natrt.exe* への適切な絶対パスまたは相対パスを指定する必要があります。

Web I/O インターフェイスクライアントの特定のバッチファイルを参照するために、絶対パスまたは相対パスを指定することができます。相対パスを指定した場合は、バッチファイルは以下のルールに従って検索されます。

1. *nwosvcd.exe* が保存されているディレクトリ内。これは、*Natural bin* ディレクトリです。
2. システムディレクトリ内。これは、システム関数 *GetSystemDirectory* を提供するディレクトリです。
3. Windows ディレクトリ内。これは、システム関数 *GetWindowsDirectory* を提供するディレクトリです。
4. PATH 環境変数にリストされているディレクトリ内。

引数

バッチファイルは以下の引数を受け取ります。

| 順番 | 引数 | 説明 |
|----|-------------------|---|
| 1 | IPAddress | セッションが開かれるクライアント IP アドレスです。 注意: プロキシがある場合は、これはクライアントワークステーションの IP アドレスではなく、プロキシの IP アドレスです。 |
| 2 | ClientID | ログオンページのユーザー名はクライアント ID として渡されます。 |
| 3 | Argument3 | 予約済み（将来的に使用される予定）です。 |
| 4 | Argument4 | 予約済み（将来的に使用される予定）です。 |
| 5 | NaturalParameters | Natural のいずれのパラメータでも指定できます。パラメータは、セッションのコンフィグレーションファイルで定義されるか、ログオンページで入力されます。次に示すのは、コンフィグレーションファイル内の対応するエントリの一例です。 <pre><natural_parameter>parm=nwoparm\ stack=(logon\ mylib;start-program;fin)<natural_parameter></pre> ログオンページで選択した言語は、最初のエレメントとして "ulang=x" の形式で Natural パラメータに追加されます。 |

引数 1 と 2 は、クライアントの監査、特定の PC からのアプリケーションの実行（IP アドレスの識別）、統計の生成、特別な操作の実行などに使用できます。

環境変数

バッチファイルでは、Natural セッション用に複数の環境変数を設定できます。

NWO_ENABLE_ACK=["YES" | "NO"]

この環境変数は、非同期画面 (SET CONTROL N) で使用されます。

YES 非同期画面をクライアントに送信すると、Natural は ACK パッケージを受け取るまで待機してから次の画面を送信します。

NO 待機せずに非同期画面を送信します。デフォルト値です。

NWO_TIMEOUT=[*number-of-seconds*]

Natural がクライアントからの入力を待機する最長時間 (秒単位) です。この時間が経過するとセッションを終了します。秒数が "0" の場合は、Natural は無限に待機します (タイムアウトなし)。デフォルト値は "0" です。

タイムアウトでエラー NAT5466 が返されます。Natural では、アプリケーションでこのエラーを処理でき、続行するのか終了するのかを決定できます。

16 Web I/O インターフェイスクライアントの設定

Web I/O インターフェイスは、UNIX、メインフレーム、または Windows ホストで実行している Natural アプリケーションからデータを受け取り、ユーザーの Web ブラウザに Web ページを表示します。

このセクションでは、Natural for Mainframes、Natural for UNIX、または Natural for Windows のランタイム環境で実行している Web I/O インターフェイスのサーバー部分で Web I/O インターフェイスクライアントを使用できるようにするために、アプリケーションサーバーまたは Web サーバーをインストールおよび構成する方法について説明します。

次のトピックについて説明します。

- [前提条件](#)
- [Sun Java System Application Server への Web I/O インターフェイスクライアントのインストール](#)
- [JBoss Application Server への Web I/O インターフェイスクライアントのインストール](#)
- [Microsoft Internet Information Services \(IIS\) への Web I/O インターフェイスクライアントのインストール](#)
- [コンフィグレーションファイルでの Natural セッションの定義](#)
- [色属性とフィールド属性の変更 \(J2EE のみ\)](#)
- [テーマの使用 \(IIS のみ\)](#)
- [Web I/O インターフェイスクライアントと端末エミュレーションの違い](#)
- [Web I/O インターフェイスを Natural アプリケーションで使用する場合の制限](#)

17 前提条件

| | |
|----------------------------------|-----|
| ■ アプリケーションサーバーまたは Web サーバー | 138 |
| ■ Apache Ant | 139 |
| ■ Natural for UNIX | 139 |
| ■ Natural for Mainframes | 139 |
| ■ Natural for Windows | 140 |
| ■ クライアントとサーバーの互換性 | 140 |
| ■ ブラウザの前提条件 | 141 |

以下では次のトピックについて説明します。

アプリケーションサーバーまたは Web サーバー

次のアプリケーションサーバー／Webサーバーがサポートされています。Web I/O インターフェイスでは、アプリケーションサーバー／Webサーバーは提供されていません。

■ J2EE サーバー

次のアプリケーションサーバーがサポートされています。

■ Sun Java System Application Server 8.1 および 8.2

これは、<http://www.sun.com/> からダウンロードできます。

■ JBoss Application Server 4.0.5

これは、<http://www.jboss.org/> からダウンロードできます。

上記のアプリケーションサーバーでは、Web I/O インターフェイスクライアントは、J2EE エンタープライズアプリケーション (*natuniapp.ear*) および J2EE リソースアダプタ (*naturalunicode.rar*) で構成されています。いずれのコンポーネントも J2EE サーバーにインストールされています。

J2EE サーバーで Web I/O インターフェイスクライアントを使用するには、Java Runtime 1.4.2 以降がインストールされていることが前提条件となります。

■ Microsoft Internet Information Services (IIS)

この Web サーバーでは、Web I/O インターフェイスクライアントは ASP.NET 2.0 テクノロジーに基づいています。したがって、IIS で Web I/O インターフェイスクライアントを実行するには、Microsoft Internet Information Services がインストールされているサーバーに .NET Framework 2.0 がインストールされていることが前提条件となります。

ASP.NET 2.0 がまだインストールされていない場合は、.NET Framework バージョン 2.0 を以下からダウンロードできます。

<http://www.microsoft.com/downloads/details.aspx?familyid=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en>

このドキュメントの後半で説明するように、いずれかのサーバーに Web I/O インターフェイスクライアントをインストールする必要があります。



Note: サーバーファームはサポートされていません。

Apache Ant

JBoss Application Server で展開するには、Apache Ant 1.6.5 以降が必要です。このツールは、<http://ant.apache.org/> から無料で入手できます。

Natural for UNIX

Natural for UNIX で Web I/O インターフェイスクライアントを使用するには、次のものをインストールする必要があります。

- Natural for UNIX バージョン 6.3.1 以降および
- Web I/O インターフェイスデーモン

詳細については、Natural for UNIX で提供されている『インストール』ドキュメントの次のセクションを参照してください。

- UNIX での *Natural* のインストールと設定
- UNIX での Web I/O インターフェイスデーモンのインストール
- UNIX での Web I/O インターフェイスデーモンのアクティブ化

Natural for Mainframes

Natural for Mainframes で Web I/O インターフェイスクライアントを使用するには、次のものをインストールする必要があります。

- Natural for Mainframes バージョン 4.2.3 以降および
- Natural Web I/O インターフェイスサーバー

詳細については、Natural for Mainframes で提供されている『インストール』ドキュメントおよび『*Natural Web I/O* インターフェイスサーバー』ドキュメントを参照してください。

Natural for Windows

Natural for Windows で Web I/O インターフェイスクライアントを使用するには、次のものをインストールする必要があります。

- Natural for Windows バージョン 6.3.3 以降および
- Web I/O インターフェイスサービス

Web I/O インターフェイスサービスは、Natural for Windows とともにインストールされます。詳細については、Natural for Windows で提供されている『オペレーション』ドキュメントの「[Web I/O インターフェイスサービスの設定](#)」を参照してください。

クライアントとサーバーの互換性

Web I/O インターフェイスがサポートされているプラットフォームに応じて、Web I/O インターフェイスのサーバー部分の実装に異なる手法が使用されています。次の説明にある「サーバー」という一般用語は、デーモン (Natural for UNIX)、サーバー (Natural for Mainframes)、およびサービス (Natural for Windows) を指します。

- サーバーは、バージョン番号が同じか高い任意のクライアントとともに動作することが可能です。

クライアントで使用しているバージョンがサーバーのバージョンより低いことがサーバーで検出されると、クライアントのバージョンが古いため接続がクローズされることがサーバーから通知されます。

- クライアントは、バージョン番号が同じか低い任意のサーバーとともに動作することが可能です。

サーバーで使用しているバージョンがクライアントのバージョンより低いことがクライアントで検出されると、クライアントはサーバーのバージョンに切り替わります。ただし、この場合はクライアントの新機能が常にサポートされるとは限りません。

ブラウザの前提条件

このバージョンでは、次のブラウザがサポートされています。

- Internet Explorer 6.0
- Mozilla Firefox 1.5

 **Important:** Cookie および JavaScript がブラウザで有効になっている必要があります。

18 Sun Java System Application Server への Web I/O インターフェイスクライアントのインストール

| | |
|-------------------|-----|
| ▪ インストール手順 | 144 |
| ▪ インストールの確認 | 151 |

Sun Java System Application Server で Web I/O インターフェイスクライアントを使用するには、このchapterで説明している手順に従う必要があります。次のトピックについて説明します。

インストール手順

Web I/O インターフェイスクライアントは、Sun Java System Application Server の管理コンソールを使用してインストールします。

ここでは次のように想定しています。

- `<host>` は、アプリケーションサーバーがインストールされているマシンの名前です。
- `<port>` は、アプリケーションサーバーがインストールされているポートの名前です。デフォルトのインストールでは、ポート 8080 になります。
- `<adminport>` は、管理コンソールがインストールされているポートの名前です。デフォルトのインストールでは、ポート 4848 になります。
- `<sunas>` は、アプリケーションサーバーがインストールされているディレクトリへのパスです。Windows でのデフォルトのインストールでは、`C:/Sun/AppServer` になります。

以下では次のトピックについて説明します。

- [初めてインストールする場合](#)
- [インストールの更新](#)

初めてインストールする場合

▶手順 18.1. Web I/O インターフェイスクライアントをインストールするには

- 1 **ServLine24** のコンポーネントのダウンロードエリアから Sun Java System Application Server 用の Natural Web I/O インターフェイスクライアントをダウンロードして、ハードディスク上の任意のディレクトリにコンテンツを解凍します。

Or:

Natural for UNIX および Natural for Windows : `nwo/<platform>/j2ee/v<nnnn>/sun-apps` ディレクトリのコンテンツをすべて、インストール媒体からハードディスク上の任意のディレクトリにコピーします。

- UNIX プラットフォーム：次のコマンドを使用して、TAR ファイルを解凍します。

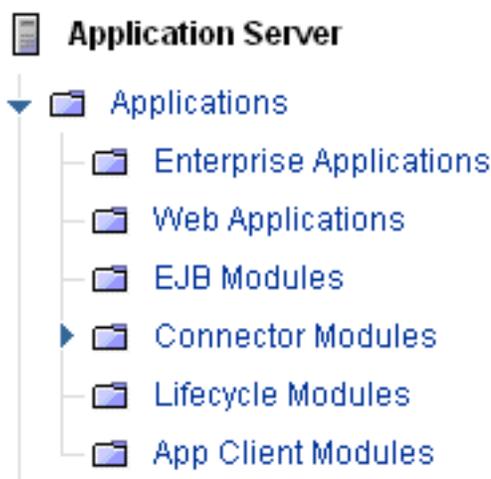
```
tar -xvf nwonnnn.tar
```

- アプリケーションサーバーが動作していることを確認してください。
- Web ブラウザを開き、次の URL を入力します。

```
http://<host>:<adminport>
```

これで管理コンソールが開きます。

- リソースアダプタ *naturalunicode.rar* を展開します。
 - ツリーノード **[Applications]** > **[Connector Modules]** を開きます。



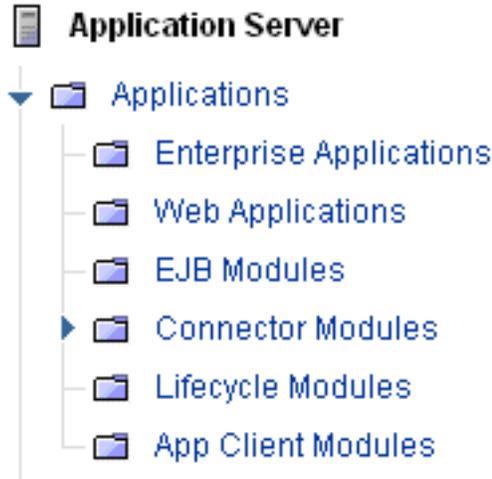
- [Deploy]** を選択します。
 - アプリケーションサーバーにアップロードするパッケージファイルとして *naturalunicode.rar* を選択します。
 - [Next]** を選択します。アプリケーション名として "naturalunicode" が自動的に含まれます。
 - [Finish]** を選択します。
- リソースアダプタの JNDI 名を定義します。

1. ツリーノード **[Resources]** > **[Connectors]** > **[Connector Connection Pools]** を開きます。



2. **[New]** を選択します。
 3. 名前に「NatPool」（名前は任意）と入力します。
 4. リソースアダプタとして **[naturalunicode]** を選択します。
 5. Natural ホストへの各接続で、新しい接続が行われます。各ユーザーには一意のホストセッションが必要であるため、接続のプーリングは使用できません。したがって、ユーザーに対して十分なセッションが存在していることを確認する必要があります。デフォルトの最大値は "32" です。
 6. **[Next]** を選択します。
 7. **[Next]** を選択します。
 8. **[Finish]** を選択します。
 9. ツリーノード **[Resources]** > **[Connectors]** > **[Connector Resources]** を開きます。
 10. **[New]** を選択します。
 11. JNDI 名に「eis/NaturalUnicodeRA」と入力します。
 12. プール名として **[NatPool]**（または前に指定した名前）を選択します。
 13. **[OK]** を選択します。
- 7 エンタープライズアプリケーション *natuniapp.ear* を展開します。

1. ツリーノード [Applications] > [Enterprise Applications] を開きます。



2. [Deploy] を選択します。
 3. アップロードするファイルとして *natuniapp.ear* を選択します。
 4. [Next] を選択します。
 5. [OK] を選択します。展開には数分かかることがあります。
- 8 アプリケーションサーバーを再起動します。

インストールの更新

▶手順 18.2. Web I/O インターフェイスクライアントを更新するには

- 1 `../AppServer/domains/domain1/applications/j2ee-apps/natuniapp/natuniweb_war/WEB-INF` 内にある `sessions.xml` ファイルのバックアップコピーを作成します。他のファイル（スタイルシートなど）に変更を加えた場合は、そのファイルのバックアップコピーも作成します。
- 2 **ServLine24** のコンポーネントのダウンロードエリアから Sun Java System Application Server 用の Natural Web I/O インターフェイスクライアントをダウンロードして、ハードディスク上の任意のディレクトリにコンテンツを解凍します。

Or:

Natural for UNIX および Natural for Windows : `nwo/⟨platform⟩/j2ee/v⟨nnnn⟩/sun-apps` ディレクトリのコンテンツをすべて、インストール媒体からハードディスク上の任意のディレクトリにコピーします。

Sun Java System Application Server への Web I/O インターフェイスクライアントのインストール

- UNIX プラットフォーム：次のコマンドを使用して、TAR ファイルを解凍します。

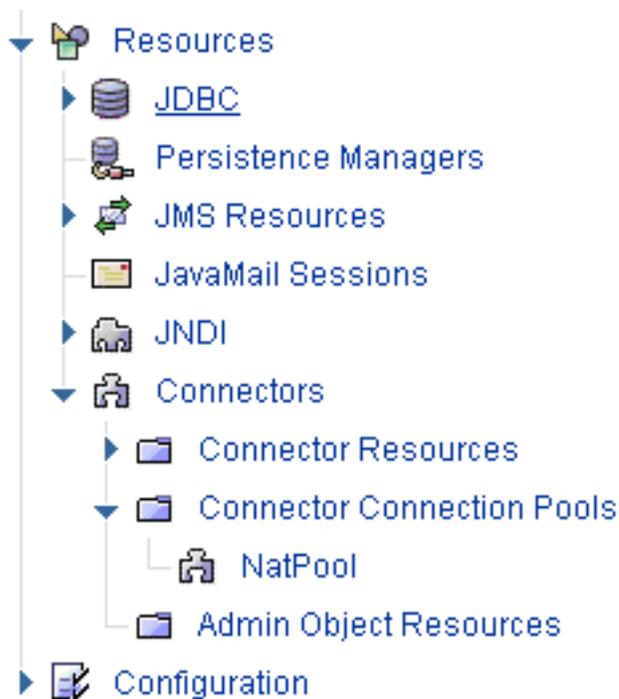
```
tar -xvf nwoonnnn.tar
```

- アプリケーションサーバーが動作していることを確認してください。
- Web ブラウザを開き、次の URL を入力します。

```
http://<host>:<adminport>
```

これで管理コンソールが開きます。

- リソースアダプタ *naturalunicode.rar* を削除します。
 - ツリーノード **[Resources]** > **[Connectors]** > **[Connector Connection Pools]** を開きます。

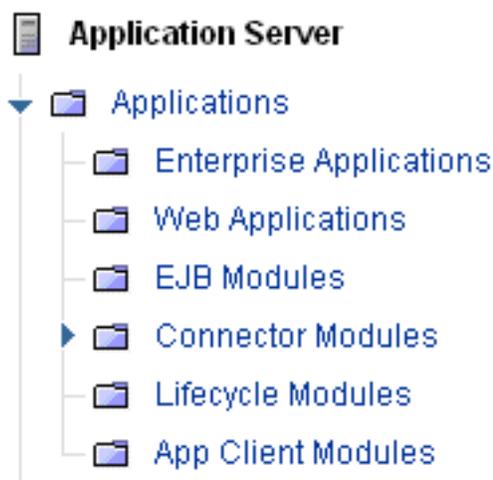


Connector Connection Pools

Deploy the connector module before creating the pool.

| Current Pools (1) | | | |
|-------------------------------------|---------|-------------|------------------|
| <input checked="" type="checkbox"/> | Name | Description | Resource Adapter |
| <input type="checkbox"/> | NatPool | | naturalunicode |

2. チェックボックス [Natpool] (または前に指定した名前のチェックボックス) をオンにします。
3. [Delete] を選択します。
4. ツリーノード [Applications] > [Connector Modules] を開きます。



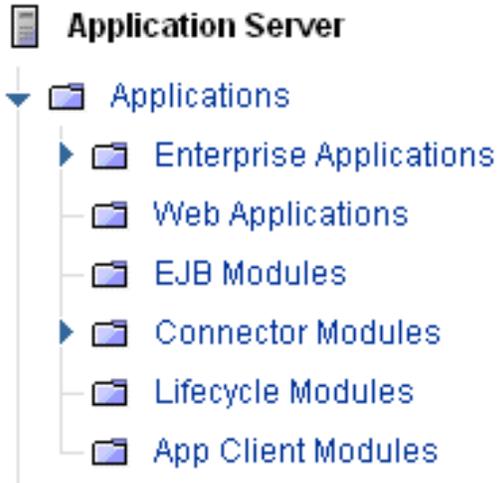
Connector Modules

A connector module is used to connect to an Enterprise Information System (EIS) and is packaged in a RAR (Resource Adapter Archive) file or directory.

| Deployed Connector Modules (1) | | | |
|-------------------------------------|------------------|---------|--|
| Deploy... Undeploy Enable Disable | | | |
| <input checked="" type="checkbox"/> | Application Name | Enabled | Location |
| <input type="checkbox"/> | naturalunicode | true | \${com.sun.aas.instanceRoot}/applications/fj2ee-modules/naturalunicode |

5. チェックボックス [naturalunicode] をオンにします。
6. [Undeploy] を選択します。
7. エンタープライズアプリケーション *natuniapp.ear* を削除します。

1. ツリーノード [Applications] > [Enterprise Applications] を開きます。



Enterprise Applications

An enterprise application is a J2EE application in an EAR (Enterprise Application Archive) file or directory.

| Deployed Enterprise Applications (1) | | | |
|--------------------------------------|------------------|---------|---|
| Deploy... Undeploy Enable Disable | | | |
| <input checked="" type="checkbox"/> | Application Name | Enabled | Location |
| <input type="checkbox"/> | natuniapp | true | \$(com.sun.aas.instanceRoot)/applications/j2ee-apps/natuniapp |

2. チェックボックス [natuniapp] をオンにします。
 3. [Undeploy] を選択します。
-
- 8 初めてインストールする場合と同様に、リソースアダプタ *naturalunicode.rar* を展開します。
 - 9 初めてインストールする場合と同様に、リソースアダプタの JNDI 名を定義します。
 - 10 初めてインストールする場合と同様に、エンタープライズアプリケーション *natuniapp.ear* を展開します。
 - 11 バックアップファイルを必要な場所にコピーして戻します。
 - 12 アプリケーションサーバーを再起動します。

インストールの確認

`http://<host>:<port>` はアプリケーションサーバーの URL とみなします。

▶手順 18.3. インストールを確認するには

- Web ブラウザで以下の URL を入力します。

```
http://<host>:<port>/natuniweb/natural.jsp
```

例：

```
http://myhost:8080/natuniweb/natural.jsp
```

WebI/O インターフェイスクライアントがブラウザで起動します。表示されるログオンページのエント리는、コンフィグレーションファイルの設定によって異なります。詳細については、「[コンフィグレーションファイルでの *Natural* セッションの定義](#)」を参照してください。

19 JBoss Application Server への Web I/O インターフェ イスクライアントのインストール

- インストール手順 154
- インストールの確認 157

JBoss Application Server で Web I/O インターフェイスクライアントを使用するには、このchapterで説明している手順に従う必要があります。次のトピックについて説明します。

インストール手順

同じ JBoss Application Server には、Web I/O インターフェイスクライアントの1つのバージョンのみをインストールできます。

JBoss Application Server には Web I/O インターフェイスクライアントまたは Natural for Ajax のいずれかをインストールできますが、両方を同時にインストールすることはできません。

<jboss> は JBoss Application Server のインストールディレクトリとみなします。

以下では次のトピックについて説明します。

- [初めてインストールする場合](#)
- [インストールの更新](#)

初めてインストールする場合

▶手順 19.1. Web I/O インターフェイスクライアントをインストールするには

- 1 **ServLine24** のコンポーネントのダウンロードエリアから JBoss Application Server 用の Natural Web I/O インターフェイスクライアントをダウンロードして、ハードディスク上の任意のディレクトリにコンテンツを解凍します。

Or:

Natural for UNIX および Natural for Windows : `nwo/<platform>/j2ee/v<nnnn>/jboss` ディレクトリのコンテンツをすべて、インストール媒体からハードディスク上の任意のディレクトリにコピーします。

- 2 UNIX プラットフォーム : 次のコマンドを使用して、TAR ファイルを解凍します。

```
tar -xvf nwonnnn.tar
```

- 3 ファイル `<jboss>/server/default/deploy/jbossjca-service.xml` を編集し、次の設定

```
<!-- Enable connection close debug monitoring -->  
<attribute name="Debug">true</attribute>
```

を次のように変更します。

```
<!-- Enable connection close debug monitoring -->  
<attribute name="Debug">false</attribute>
```

- 4 Apache Ant をインストールします。Web I/O インターフェイスクライアントを JBoss Application Server に展開するには、Apache Ant が必要です。必要なバージョン番号については、上記の「[前提条件](#)」を参照してください。

1. (<http://ant.apache.org/> から) Apache Ant をダウンロードして、任意のインストールディレクトリに解凍します。空白を含むディレクトリ名は使用しないでください。
2. 環境変数 `ANT_HOME` がディレクトリ `<ant>` (`<ant>` は Ant のインストールディレクトリ) をポイントするようにします。
3. `<ant>/bin` を `PATH` 環境変数に追加します。

- 5 Web I/O インターフェイスクライアントを JBoss Application Server に展開します。

1. Web I/O インターフェイスクライアントの配布ファイルをディスクドライブ上のディレクトリにコピーします。
2. Web I/O インターフェイスクライアント配布ファイルを含むディレクトリには、`jbossdeploynwo.xml` という名前の Ant スクリプトがあります。このスクリプトを編集し、次の設定

```
<property name="jbossHome" value="C:/Program Files/Java/jboss-4.0.5"/>
```

を次のように変更します。

```
<property name="jbossHome" value="<jboss>"/>
```

`<jboss>` は、JBoss Application Server のインストールディレクトリです。

 **Important:** ディレクトリパスを指定するときは、(バックスラッシュでない) 通常のスラッシュを使用してください (Windows でも同様)。

3. 次のコマンドを入力して、スクリプト `jbossdeploynwo.xml` を実行します。

```
ant -f jbossdeploynwo.xml
```

メッセージ「BUILD SUCCESSFUL」を待ちます。これは、展開が成功したことを示します。

6 JBoss Application Server を起動します。

インストールの更新

▶手順 19.2. Web I/O インターフェイスクライアントを更新するには

1 **ServLine24** のコンポーネントのダウンロードエリアから JBoss Application Server 用の Natural Web I/O インターフェイスクライアントをダウンロードして、ハードディスク上の任意のディレクトリにコンテンツを解凍します。

Or:

Natural for UNIX および Natural for Windows : `nwo<platform>/j2ee/v<nnnn>/jboss` ディレクトリのコンテンツをすべて、インストール媒体からハードディスク上の任意のディレクトリにコピーします。

2 UNIX プラットフォーム：次のコマンドを使用して、TAR ファイルを解凍します。

```
tar -xvf nwo<nnnn>.tar
```

3 JBoss Application Server をシャットダウンします。

4 Web I/O インターフェイスクライアントを JBoss Application Server に展開します。

1. Web I/O インターフェイスクライアントの配布ファイルをディスクドライブ上のディレクトリにコピーします。

2. Web I/O インターフェイスクライアント配布ファイルを含むディレクトリには、`jbossdeploynwo.xml` という名前の Ant スクリプトがあります。このスクリプトを編集し、次の設定

```
<property name="jbossHome" value="C:/Program Files/Java/jboss-4.0.5"/>
```

を次のように変更します。

```
<property name="jbossHome" value="<jboss>"/>
```

<jboss> は、JBoss Application Server のインストールディレクトリです。

 **Important:** ディレクトリパスを指定するときは、（バックスラッシュでない）通常のスラッシュを使用してください（Windows でも同様）。

3. 次のコマンドを入力して、スクリプト `jbossdeploynwo.xml` を実行します。

```
ant -f jbossdeploynwo.xml redeploy
```

メッセージ「BUILD SUCCESSFUL」を待ちます。これは、展開が成功したことを示します。

- JBoss Application Server を起動します。

インストールの確認

`http://<host>:<port>` はアプリケーションサーバーの URL とみなします。

▶手順 19.3. インストールを確認するには

- Web ブラウザで以下の URL を入力します。

```
http://<host>:<port>/natuniweb/natural.jsp
```

例：

```
http://myhost:8080/natuniweb/natural.jsp
```

WebI/O インターフェイスクライアントがブラウザで起動します。表示されるログオンページのエントリは、コンフィグレーションファイルの設定によって異なります。詳細については、「[コンフィグレーションファイルでの *Natural* セッションの定義](#)」を参照してください。

20 Microsoft Internet Information Services (IIS) への Web I/O インターフェイスクライアントのインストール

- インストール手順 160
- インストールの確認 162

Microsoft Internet Information Services (IIS) で Web I/O インターフェイスクライアントを使用するには、このchapterで説明している手順に従う必要があります。次のトピックについて説明します。

インストール手順

次の説明は、Windows XP を操作している場合に適用されます。異なるバージョンの Windows を操作している場合には、手順が多少異なる可能性があります。

以下では次のトピックについて説明します。

- [初めてインストールする場合](#)
- [インストールの更新](#)

初めてインストールする場合

▶手順 20.1. Web I/O インターフェイスクライアントをインストールするには

- 1 **ServLine24** のコンポーネントのダウンロードエリアから IIS 用の Natural Web I/O インターフェイスクライアントをダウンロードして、ハードディスク上の任意のディレクトリにコンテンツを解凍します。

Or:

Natural for UNIX および Natural for Windows : `nwo/windows/iis/v<nnnn>` ディレクトリのコンテンツをすべて、インストール媒体からハードディスク上の任意のディレクトリにコピーします。

- 2 [コントロールパネル] を開き、[管理ツール] > [インターネット インフォメーションサービス] を選択します。
- 3 表示されるウィンドウで、コンピュータ名の付いたノードをツリー内で展開し、ノード [Web サイト] > [既定の Web サイト] を選択します。
- 4 コンテキストメニューを開き、[新規作成] > [仮想ディレクトリ] を選択します。

Or:

[操作] メニューの [新規作成] > [仮想ディレクトリ] を選択します。

新しい仮想ディレクトリを作成するウィザードが表示されます。

- 5 [次へ] ボタンを選択し、ウィザードの次のページに進みます。
- 6 仮想ディレクトリのエイリアス名（「Natural」など）を入力し、[次へ] ボタンを選択します。

 **Important:** Eclipse 用の Natural Execution and Debugging プラグインを使用する場合、仮想ディレクトリのエイリアス名は "Natural" である必要があります。

- 7 ハードディスクにコピーしたファイルを含むディレクトリへのパスを入力し、[次へ] ボタンを選択します。
- 8 アクセス許可はデフォルト値（[読み取り] および [スクリプトを実行する] が選択済み）のままにしておき、[次へ] ボタンを選択します。
- 9 [完了] ボタンを選択します。

これで、IIS サーバーに仮想ディレクトリが作成されます。

- 10 新しい仮想ディレクトリの名前をツリーから選択します。
- 11 コンテキストメニューを開き、[プロパティ] を選択します。

Or:

[操作] メニューの [新規作成] > [プロパティ] を選択します。

- 12 表示されるダイアログボックスで、[ASP.NET] ページを選択します。ASP.NET のバージョンを選択します。最低でも 2.0.50727 (ASP.NET 2.0 のリリースバージョン) である必要があります。[OK] ボタンを選択して、ダイアログボックスを閉じます。

インストールの更新

▶手順 20.2. Web I/O インターフェイスクライアントを更新するには

- 1 ハードディスクで、インストールする更新バージョン用に新しいディレクトリを作成します。
- 2 **ServLine24** のコンポーネントのダウンロードエリアから IIS 用の Natural Web I/O インターフェイスクライアントをダウンロードして、新しいディレクトリにコンテンツを解凍します。

Or:

Natural for UNIX および Natural for Windows : `nwo/windows/iis/v<nnnn>` ディレクトリのコンテンツをすべて、インストール媒体からハードディスク上の任意のディレクトリにコピーします。

- 3 [コントロールパネル] を開き、[管理ツール] > [インターネット インフォメーション サービス] を選択します。
- 4 既存の仮想ディレクトリ ("Natural" など) のプロパティを開きます。[仮想ディレクトリ] ページで、更新バージョンを含む新しいディレクトリをポイントするようにパスを変更します。
- 5 以前のバージョンを含むディレクトリから新しいディレクトリに `settings.xml` ファイルをコピーします。`settings.xml` ファイルはインストールディレクトリのルート内にあります。他

のファイル（スタイルシートなど）に変更を加えた場合は、そのファイルも新しいディレクトリにコピーします。

インストールの確認

`http://<host>:<port>` は Web サーバーの URL とみなします。

▶手順 20.3. インストールを確認するには

- IIS でエイリアス名 "Natural" を定義した場合は（上記を参照）、次の URL を使用して Web I/O インターフェイスクライアントを起動できます。

Web ブラウザで以下の URL を入力します。

```
http://<host>:<port>/Natural/Default.aspx
```

例：

```
http://myhost:8080/Natural/Default.aspx
```

Web I/O インターフェイスクライアントがブラウザで起動します。表示されるログオンページのエント리는、コンフィグレーションファイルの設定によって異なります。詳細については、「[コンフィグレーションファイルでの *Natural* セッションの定義](#)」を参照してください。

21 コンフィグレーションファイルでの Natural セッ ションの定義

| | |
|------------------------------|-----|
| ▪ ログオンページについて | 164 |
| ▪ コンフィグレーションファイルの名前と場所 | 165 |
| ▪ コンフィグレーションファイルの内容 | 166 |
| ▪ コンフィグレーションファイル要素の概要 | 167 |
| ▪ ログオンページでのパスワードの変更 | 171 |
| ▪ タイムアウトの設定 (J2EE のみ) | 172 |
| ▪ ブラウザの制限 | 172 |

このchapterでは、次のトピックについて説明します。

ログオンページについて

Web I/O インターフェイスクライアントをブラウザで起動すると、ログオンページが表示されます。このログオンページのエントリは、コンフィグレーションファイルの設定によって異なります。

次の例は、セッション用に特殊なエントリが定義されていないデフォルトのコンフィグレーションファイルを使用した場合に表示されるログオンページを示しています。



The screenshot shows the logon page for Natural 2006. The page title is "Enterprise-scale Application Development". The form includes the following elements:

- Session ID:** A dropdown menu with "Nothing configured" selected.
- Logon:** A button.
- Change password:** A link.
- Natural Server:** An input field.
- Natural Port:** An input field.
- User Name:** An input field.
- Password:** An input field.
- Natural Application:** An input field.
- Natural Parameter:** An input field.
- Theme:** A dropdown menu with "3270Theme" selected.

 **Note:** 上記のログオンページには、**[Theme]** ドロップダウンリストボックスが表示されています。テーマは IIS でのみ使用できます。

次の例は、多くの設定（ユーザー ID およびパスワードを含む）が事前定義されているコンフィグレーションファイルを使用した場合に表示されるログオンページを示しています。



Naturalセッションにログオンするには、すべての必須情報をログオンページで指定する必要があります（対応するドロップダウンリストボックスからセッションを選択するなど）。[Logon] ボタンを選択すると、選択したセッションの画面が表示されます。

コンフィグレーションファイルの名前と場所

ログオンページから呼び出すことができるセッションを定義するには、コンフィグレーションファイルが必要です。

■ J2EE サーバー

コンフィグレーションファイルの名前は *sessions.xml* です。このファイルは、次の場所にあります。

```
<application-server-install-dir>/domains/domain1/applications/j2ee-apps/natuniapp/natuniweb_war/WEB-INF
```

■ Microsoft Internet Information Services (IIS)

コンフィグレーションファイルの名前は *settings.xml* です。このファイルは、IIS 内のアプリケーションのルートディレクトリにあります。

コンフィグレーションファイルの内容

デフォルトでは、コンフィグレーションファイルには次のエントリが含まれています。

```
<?xml version="1.0" encoding="utf-8"?>
<settings>
  <global>
    <last_activity_timeout>3600</last_activity_timeout>
  </global>
  <sessions>
    <session id="Nothing configured">
    </session>
  </sessions>
</settings>
```

 **Note:** タイムアウト値が含まれている <global> セクションは、J2EE コンフィグレーションファイルでのみ使用できます。「[タイムアウトの設定](#)」も参照してください。

上記の場合には、セッションのログオンに必要な入力フィールドがすべてログオンページに表示されます。

新しいセッション定義をコンフィグレーションファイルに追加するには、次の形式で新しい <session> 要素を追加する必要があります。

```
<session id="your-session-name" trace="false">
  <natural_server>your-Natural-server</natural_server>
  <natural_port>port-ID-of-Natural-server</natural_port>
  <natural_program>Natural-program-name</natural_program>
</session>
```

上記のエントリは必須です。また、「[コンフィグレーションファイル要素の概要](#)」セクションで説明しているようにオプションのエントリを指定することもできます。

次に、コンフィグレーションファイルの例を示します。

```
<?xml version="1.0" encoding="utf-8"?>
<settings>
  <global>
    <last_activity_timeout>3600</last_activity_timeout>
  </global>
  <sessions>
    <session id="DUMP (Mainframe)" trace="false">
      <natural_server>myserver1</natural_server>
```

```

<natural_port>1234</natural_port>
<natural_program>DUMP</natural_program>
</session>
<session id="Test (UNIX)" trace="false">
  <natural_server>Myserver2</natural_server>
  <natural_port>4321</natural_port>
  <natural_program>test.sh</natural_program>
</session>
</sessions>
</settings>

```

コンフィグレーションファイル要素の概要

次の表では、新規セッションを追加するためにコンフィグレーションファイルで使用できる要素および属性について説明します。

| 要素名 | 属性名 | 説明 |
|---------|-------|---|
| session | id | 任意のセッション名。ログオンページでは、セッション名がドロップダウンリストボックスに表示されます。 |
| session | trace | IIS のみ。 通常は "false"。Software AG サポートでトレースを必要とする場合にのみ、"true" に設定する必要があります。"true" に設定されている場合は、トレースが次のディレクトリに書き込まれます。 <i><drive>:\Documents and Settings\<pc-name>\ASPNET\Application Data\Software AG\Natural WebIO\Traces</i> |
| session | type | オプション。ユーザー ID およびパスワードが認証されるプラットフォーム。設定可能値は次のとおりです。 <ul style="list-style-type: none"> ■ MF Natural for Mainframes。ユーザー ID とパスワードには最大 8 文字を使用できます。 ■ MF-NSC Natural Security が搭載された Natural for Mainframes。ユーザー ID とパスワードには最大 8 文字を使用できます。ユーザー ID は、ライブラリ名に関する Natural の命名規則に準拠している必要があります。 ■ OS Natural for UNIX または Natural for Windows：ユーザー ID およびパスワードには最大 32 文字を使用できます。ドメインが必要な場合は、ユーザー ID とともにドメインを指定する必要があります ("domain\user-ID" の形式)。 デフォルト："OS"。 |

コンフィグレーションファイルでの Natural セッションの定義

| 要素名 | 属性名 | 説明 |
|----------------|----------|---|
| session | protocol | <p>オプション。Web I/O インターフェイスの現在のプロトコルバージョンは 4 です。最新の Natural バージョンでは、適切なプロトコルバージョンが自動的に使用されます。以前の Natural バージョンでは、適切なプロトコルバージョンをコンフィグレーションファイルで定義する必要があります。</p> <p>■ Natural バージョン 6.2.1～6.2.4 (UNIX) には、プロトコルバージョン 2 が必要です。</p> <p>バージョン 6.2.5 の時点で、プロトコルバージョンの定義は不要になりました。</p> <p>■ Natural バージョン 6.3.1 および 6.3.2 (UNIX) では、プロトコルバージョン 3 が必要です。</p> <p>バージョン 6.3.3 の時点で、プロトコルバージョンの定義は不要になりました。</p> <p>■ Natural バージョン 4.2.3.2 以下 (メインフレーム) では、プロトコルバージョン 3 が必要です。</p> <p>バージョン 4.2.3.3 および 4.2.4 の時点で、プロトコルバージョンの定義は不要になりました。</p> <p>設定可能値：2、3 および 4。デフォルト：4。</p> |
| user | id | <p>オプション。現在のマシンに有効なユーザー ID。この属性を指定すると、対応するフィールドはログオンページに表示されません。</p> |
| user | pwd | <p>オプション。上記のユーザー ID に有効なパスワード。この属性を指定すると、対応するフィールドはログオンページに表示されません。</p> |
| user | ucase | <p>オプション。"true" に設定されていると、ユーザー ID の入力フィールドが大文字モードになります。</p> <p>デフォルト："false"。</p> |
| natural_server | | <p>Natural および Web I/O インターフェイスデーモン (Natural for UNIX)、Web I/O インターフェイスサーバー (Natural for Mainframes)、または Web I/O インターフェイスサービス (Natural for Windows) が稼動しているサーバーの名前または TCP/IP アドレス。この要素を指定すると、対応するフィールドはログオンページに表示されません。</p> |
| natural_port | | <p>Web I/O インターフェイスデーモン (Natural for UNIX)、Web I/O インターフェイスサーバー (Natural for Mainframes)、または Web I/O インターフェイスサービス (Natural for Windows) が聴取している TCP/IP ポート番号。この要素を指定すると、対応するフィールドはログオンページに表示されません。</p> |

| 要素名 | 属性名 | 説明 |
|-------------------|---------|---|
| natural_program | | <p>■ Natural for Mainframes NEXTプロンプトで入力するとアプリケーションが起動される Natural プログラムまたはコマンドシーケンスの名前。次に例を示します。</p> <pre>TEST01 data1,data2</pre> <p>■ Natural for UNIX Natural アプリケーションを起動するための UNIX シェルスクリプトの名前 (<i>nwo.sh</i> に似たファイル)。</p> <p>■ Natural for Windows Natural アプリケーションを起動するための Windows コマンドファイル (<i>.bat</i>) の名前。</p> <p>この要素を指定すると、対応するフィールドはログオンページに表示されません。</p> |
| natural_parameter | | <p>オプション。Natural アプリケーションを起動するためのパラメータ。スタックパラメータ、パラメータファイル/モジュール、または他の Natural 固有情報がこのパラメータとなる可能性があります。</p> <p>■ Natural for Mainframes この要素は、Natural アプリケーションの起動に使用されます。次に例を示します。</p> <pre>STACK=(LOGON MYAPPL)</pre> <p>■ Natural for UNIX および Natural for Windows この要素は、上記のシェルスクリプト (UNIX) またはコマンドファイル (Windows) で "natural" の後にパラメータ \$5 を指定する場合に使用されます。次に例を示します。</p> <pre>parm=myparm\ stack=(logon\ mylib;menu;fin)</pre> <p>"parm" は Natural パラメータファイルを呼び出し、"stack" は Natural アプリケーションそのものを呼び出します。</p> <p>重要: 一重引用符は使用しないでください。文字列に空白が含まれている場合は、スラッシュ (/) を代わりに使用します。</p> |
| natural_parameter | visible | <p>IIS のみ。</p> <p>ブールパラメータ。"true" に設定されていると、Natural パラメータを含むフィールドがログオンページに表示されます。設定可能値："true" および "false"。デフォルト："false"。</p> |
| nattimeout | | <p>J2EE のみ。</p> <p>ホストからの応答のタイムアウト。「タイムアウトの設定」を参照してください。</p> |
| theme | | <p>IIS のみ。</p> |

コンフィグレーションファイルでの Natural セッションの定義

| 要素名 | 属性名 | 説明 |
|--------|----------|--|
| | | Web ページで使用しているテーマ (スタイル)。最初のテストでは、"3270Theme" という名前のテーマを使用できます (「 テーマの使用 (IIS のみ) 」を参照)。この要素を指定すると、対応するフィールドはログオンページに表示されません。テーマがログオンページに表示されると、ドロップダウンリストボックスに表示されます。 |
| screen | rows | オプション。出力ウィンドウの行数。設定可能値：最小値24、上限なし。デフォルト：24。 Natural for Mainframes では使用されず、代わりにプロファイルパラメータ TMODEL が使用されます。 |
| screen | columns | オプション。出力ウィンドウの列数。設定可能値：最小値80、上限なし。デフォルト：80。 Natural for Mainframes では使用されず、代わりにプロファイルパラメータ TMODEL が使用されます。 |
| screen | top | IIS のみ。 ブラウザ内の出力ウィンドウの上端。上端には "0" を使用します。この値はピクセルではなく Natural 単位で指定されます。 |
| screen | left | IIS のみ。 ブラウザ内の出力ウィンドウの左端。左端には "0" を使用します。この値はピクセルではなく Natural 単位で指定されます。 |
| screen | pfkeypos | IIS のみ。 PF キーの位置。設定可能値："bottom" および "right"。デフォルト："bottom"。 |
| screen | size | IIS のみ。 出力ウィンドウのサイズ。設定可能値："normal"、"small"、"tiny"、および "extratiny"。デフォルト："normal"。 |

フィールドがログオンページに表示されない場合、上記のように対応する要素または属性を指定できます。これには、属性または要素の値を指定するか、値を省略します。例えば、次のいずれかを指定できます。

```
<natural_program></natural_program>
```

または

```
<natural_program>sysprof</natural_program>
```

いずれの場合も、対応するフィールドはログオンページに表示されません。

要素名または属性名が指定されていない場合にのみ、対応するフィールドがログオンページに表示されます。



Note: 「色属性とフィールド属性の変更 (J2EE のみ)」も参照してください。

ログオンページでのパスワードの変更

現時点では、この機能は Natural on UNIX および Natural for Windows でのみ使用できます。

次の情報は、ユーザー ID およびパスワードの入力フィールドがログオンページに表示される場合に適用されます。このフィールドは、ユーザー ID およびパスワードがコンフィグレーションファイルで定義済みの場合には表示されず、**[Change password]** リンクも表示されません。

ログオンページの **[Change password]** リンクを選択すると、次の入力フィールドが Web フォームに追加されます。

■ New Password

■ Repeat New Password

パスワードを変更するには、ユーザー ID および現在のパスワードを通常どおり入力します。次に、新しいパスワードを2つの新しい入力フィールドに入力し、**[Logon]** ボタンを選択してパスワードを変更します。

パスワードを変更しない場合は、**[Don't change password]** リンクを選択します。2つの新しい入力フィールドが表示されなくなります。

パスワードが期限切れになると、Web フォームから新しいパスワードが自動的に求められます。現在のパスワードでログオンしようとする、エラーメッセージが表示され、パスワード変更用の2つの入力フィールドが表示されます。この場合は、**[Change password]** リンクを選択する必要はありません。

タイムアウトの設定（J2EE のみ）

2つの異なるタイムアウトを設定できます。

■ 最後のアクティビティのタイムアウト

ユーザーの最後のアクティビティ以降のタイムアウトです。コンフィグレーションファイルの <global> セクションで定義し、秒単位で設定します。デフォルトは1時間（3600秒）です。定義された秒数に達すると、セッションがクローズされます。

```
<?xml version="1.0" encoding="utf-8"?>
<settings>
  <global>
    <last_activity_timeout>3600</last_activity_timeout>
  </global>
</sessions>
...
```

■ Natural サーバーのタイムアウト

このタイムアウトでは、ページの更新が Natural に送信された後、Natural からの応答をクライアントが待機する時間を分単位で定義します。セッション定義の <nattimeout> 要素で定義します。デフォルトは10分です。通常、このデフォルト値を変更する必要はありません。このタイムアウトは、セッションごとに個別に設定できます。

```
...
<session id="Test (UNIX)" trace="false">
  <natural_server>Myserver2</natural_server>
  <natural_port>4321</natural_port>
  <natural_program>test.sh</natural_program>
  <nattimeout>600</nattimeout>
</session>
...
```

ブラウザの制限

ブラウザの [戻る] および [進む] ボタンは Web I/O インターフェイスでは動作しないため、使用しないでください。

2つの Natural セッションを同時に実行する場合は、（Windows の [クイック起動] ツールバーで対応するアイコンを選択するなどして）ブラウザの新しいインスタンスを開始する必要があります。ブラウザの [新規ウィンドウ] 機能は使用しないでください。この機能を使用すると、1つのセッションが2つのブラウザで稼動することになりますが、これは許可されていません。

22 色属性とフィールド属性の変更 (J2EE のみ)

| | |
|-------------------------|-----|
| ■ 前景色の設定 | 174 |
| ■ 背景色の設定 | 175 |
| ■ PF キーボタンの適合 | 177 |
| ■ 反転属性の設定 | 177 |
| ■ 下線属性と点滅属性の設定 | 178 |
| ■ 筆記体／イタリック体属性の設定 | 178 |
| ■ 高輝度属性の設定 | 179 |

このchapterの情報は、J2EE サーバーにのみ適用されます。

Web I/O インターフェイスクライアントでは、Natural フィールドの色属性（前景色）、背景色、および Natural フィールド属性のレンダリングは XSLT ファイルによって制御されます。

次の XSLT ファイルは、`<installdir>/WEB-INF` ディレクトリで使用できます。

- `transuni.xml` (Internet Explorer の場合)
- `transuni-ff.xml` (Mozilla Firefox の場合)

このchapterの情報は、いずれの XSLT ファイルにも適用されます。XSLT ファイルは、サーバーの起動時に 1 回だけ読み取られます。したがって、このファイルを変更すると、サーバーを再起動して変更内容を有効にする必要があります。

このchapterでは、次のトピックについて説明します。

前景色の設定

フォントの前景色は、XSLT ファイルの次のセクションにマップされます。

```
<!-- Map foreground colors -->
<xsl:variable name="forecol">
  <xsl:choose>
    <xsl:when test="@forecol='Red'">red</xsl:when>
    <xsl:when test="@forecol='Green'">green</xsl:when>
    <xsl:when test="@forecol='Blue'">blue</xsl:when>
    <xsl:when test="@forecol='White'">white</xsl:when>
    <xsl:when test="@forecol='Pink'">fuchsia</xsl:when>
    <xsl:when test="@forecol='Turquoise'">aqua</xsl:when>
    <xsl:when test="@forecol='Yellow'">yellow</xsl:when>
    <xsl:when test="@forecol='Black'">white</xsl:when>
    <xsl:when test="@forecol='Transparent'">
      <xsl:choose>
        <xsl:when test="@intensified='True'">aqua</xsl:when>
        <xsl:otherwise>white</xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>white</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

```
</xsl:choose>  
</xsl:variable>
```

例えば、Natural 画面の色属性 Red は、CSS カラー "red" にマップされます。

```
<xsl:when test="@forecol='Red'">red</xsl:when>
```

また、色属性を RGB 値にマップすることもできます。次に例を示します。

```
<xsl:when test="@forecol='Red'">#FF0000</xsl:when>
```

色属性 Transparent では、属性 intensified が設定されているかどうかチェックされます。設定されている場合は、"aqua" が使用されます。設定されていない場合は、"white" が使用されます。

CSS カラーの詳細については、http://www.w3schools.com/css/css_colors.asp を参照してください。

背景色の設定

背景色は、XSLT ファイルの先頭で定義されるグローバル変数 defbackcolor で設定されます。

```
<xsl:variable name="defbackcolor" select="'000000'"/>
```

RGB 値 "000000" は、黒を背景色として定義します。

また、次で説明するように従来の Natural 画面に使用される JSP ファイルでも背景色を定義する必要があります。

- Web I/O インターフェイスクライアント用の JSP ファイル
- Natural for Ajax 用の JSP ファイル

Web I/O インターフェイスクライアント用の JSP ファイル

ファイル *naturalsession.jsp* を適合させる必要があります。このファイルは `<installdir>` ディレクトリにあります。背景色は、Internet Explorer および Firefox の両方に対して `body` 要素で設定する必要があります。

```
        if(browserIE6){
%>
            <script src="scripts/natunicscript-ie.js" type="text/javascript"></script>
            <link rel="stylesheet" href="resources/natural.css"
type="text/css"></link>
            </head>

            <body onload="runLoadScript('webio')" onkeydown="checkKeyPress()"
onhelp="cancelHelp()" bgcolor="000000" onbeforeunload="handleDisconnect();">
<%
                }
            else if (browserFirefox){
%>
                <script src="scripts/natunicscript-ff.js" type="text/javascript"></script>
                <link rel="stylesheet" href="resources/natural.css"
type="text/css"></link>
                </head>

<body onload="runLoadScript('webio')" bgcolor="000000" onbeforeunload="handleDisconnect();"
>
<%
                }
            }
        }
```

Natural for Ajax 用の JSP ファイル

Natural for Ajax を操作している場合、ファイル *natlegacy.jsp* を適合させる必要があります。このファイルも `<installdir>` ディレクトリにあります。背景色は、Internet Explorer および Firefox の両方に対して `body` 要素で設定する必要があります。コードは、上記の Web I/O インターフェイスクライアント用のものと同じです。唯一の違いは、`onload` 命令で ("webio" の代わりに) 値 "rich" が使用されることです。

```
        if(browserIE6){
%>
            <script src="scripts/natunicscript-ie.js" type="text/javascript"></script>
            <link rel="stylesheet" href="resources/natural.css"
type="text/css"></link>
            </head>

            <body onload="runLoadScript('rich')" onkeydown="checkKeyPress()"
onhelp="cancelHelp()" bgcolor="000000" onbeforeunload="handleDisconnect();">
<%
                }
            else if (browserFirefox){
%>
                <script src="scripts/natunicscript-ff.js" type="text/javascript"></script>
                <link rel="stylesheet" href="resources/natural.css"
type="text/css"></link>
                </head>

<body onload="runLoadScript('rich')" bgcolor="000000" onbeforeunload="handleDisconnect();"
>
<%
                }
            }
        }
```

```

onhelp="cancelHelp()" bgcolor="000000" onbeforeunload="handleDisconnect();">
<%
    }
    else if (browserFirefox){
%>
        <script src="scripts/natunicscript-ff.js" type="text/javascript"></script>
        <link rel="stylesheet" href="resources/natural.css"
type="text/css"></link>
    </head>
    <body onload="runLoadScript('rich')" bgcolor="000000"
onbeforeunload="handleDisconnect();" >
<%
    }

```

PF キーボタンの適合

Natural for UNIX では、各 PF キーを PF キー行にボタンとして表示できます。ボタンの外観（スタイル）は、ディレクトリ `<installdir>/resources` にある CSS ファイル `natural.css` を使用して適合できます。

反転属性の設定

反転表示のフィールド属性は、XSLT ファイルの次のセクションで設定されます。このセクションは、前景色がマップされるセクションの直後にあります。

```

<!-- START SET FOREGROUND AND BACKGROUND COLORS FOR REVERSE ATTRIBUTE -->
<!-- Set foreground colors -->
<xsl:variable name="ForeCol">
  <xsl:choose>
    <xsl:when test="@reverse='True'">color:<xsl:value-of
select="$defbackcolor"/></xsl:when>
    <xsl:otherwise>color:<xsl:value-of select="$forecol"/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<!-- Set background colors -->
<xsl:variable name="BackCol">
  <xsl:choose>
    <xsl:when test="@reverse='True'">background-color:<xsl:value-of
select="$forecol"/></xsl:when>
    <xsl:otherwise>background-color:<xsl:value-of
select="$defbackcolor"/></xsl:otherwise>
  </xsl:choose>

```

```
</xsl:variable>  
<!-- END SET FOREGROUND AND BACKGROUND COLORS FOR REVERSE ATTRIBUTE -->
```

reverse 属性が true に設定されている場合、前景色は \$defbackcolor の値に設定され、背景色は前景色の値になります。

reverse 属性が true に設定されていない場合、前景色はマップされている前景色の値になり、背景色は \$defbackcolor の値になります。

下線属性と点滅属性の設定

下線付きのテキストおよび点滅テキストのフィールド属性は、XSLT ファイルの次のセクションで設定されます。

```
<!-- Set underline and blinking attributes -->  
<xsl:variable name="TextDecoStyle">  
  <xsl:choose>  
    <xsl:when test="@underline='True'">text-decoration:underline;</xsl:when>  
    <xsl:when test="@blinking='True'">text-decoration:blink;</xsl:when>  
    <xsl:otherwise>text-decoration:normal;</xsl:otherwise>  
  </xsl:choose>  
</xsl:variable>
```

点滅テキストのフィールド属性は、Internet Explorer ではサポートされていません。

筆記体／イタリック体属性の設定

筆記体／イタリック体テキストのフィールド属性は、XSLT ファイルの次のセクションで設定されます。

```
<!-- Set cursive attribute -->  
<xsl:variable name="FontStyle">  
  <xsl:choose>  
    <xsl:when test="@italic='True'">font-style:italic;</xsl:when>  
    <xsl:otherwise>font-style:normal;</xsl:otherwise>  
  </xsl:choose>  
</xsl:variable>
```

```
</xsl:choose>  
</xsl:variable>
```

高輝度属性の設定

高輝度テキストのフィールド属性は、XSLT ファイルの次のセクションで設定されます。

```
<!-- Set intensified attribute - draw text as bold -->  
<xsl:variable name="FontWeight">  
  <xsl:choose>  
    <xsl:when test="@intensified='True'">font-weight:bolder</xsl:when>  
    <xsl:otherwise>font-weight:normal;</xsl:otherwise>  
  </xsl:choose>  
</xsl:variable>
```

23 テーマの使用 (IIS のみ)

| | |
|-----------------|-----|
| ■ スキンファイル | 182 |
| ■ スタイルシート | 184 |

このchapterの情報は、Microsoft Internet Information Services (IIS) に対してのみ適用されません。

テーマは、Natural アプリケーションの出力スタイルを定義するために使用されます。一部のテーマは、インストールパッケージにすでに含まれています。

テーマは ASP.NET 2.0 の標準機能です。テーマは *App_Themes* フォルダに含まれている必要があります。テーマごとに、*App_Themes* フォルダにサブフォルダが含まれています。例えば、"3270Thema" という名前のテーマは、フォルダ *3270Theme* に入っています。

テーマ用に提供される各サブフォルダには、次のファイルが含まれている必要があります。

- *SkinFile.skin* (以下を参照)
- *StyleSheet.css* (以下を参照)

このchapterでは、次のトピックについて説明します。

スキンファイル

スキンファイル (*SkinFile.skin*) には、Natural 要素 (フィールドなど) のエントリが次の形式で含まれています。

```
<asp:Label SkinId="Red" runat="server" forecolor="red"/>
<asp:Label SkinId="Green" runat="server" forecolor="green"/>
<asp:Label SkinId="Blue" runat="server" forecolor="blue" />
...
```

次の表に、Web I/O インターフェイスで Natural 画面をマップするために必要なスキンファイルの最小設定を示します。この表には、次の列があります。

- **Natural 属性**
Natural で設定できる属性。
- **出力フィールド**
テキスト用の Natural 出力フィールド。
- **入力フィールド**
データを入力できる Natural フィールド。
- **読み取り専用入力フィールド**
Natural 画面にウィンドウが表示される場合、メイン画面の入力フィールドは読み取り専用を設定されます。一番手前のウィンドウの後ろにあるウィンドウの入力フィールドも、読み取り専用として表示されます。

| Natural 属性 | 出力フィールド | 入力フィールド | 読み取り専用入力フィールド |
|--------------|--|--|---|
| Red | <asp:Label runat="server" SkinId="Red"/> | <asp:TextBox runat="server" SkinId="Red"/> | <asp:TextBox runat="server" SkinId="RedReadOnly"/> |
| Green | <asp:Label runat="server" SkinId="Green"/> | <asp:TextBox runat="server" SkinId="Green"/> | <asp:TextBox runat="server" SkinId="GreenReadOnly"/> |
| Blue | <asp:Label runat="server" SkinId="Blue"/> | <asp:TextBox runat="server" SkinId="Blue"/> | <asp:TextBox runat="server" SkinId="BlueReadOnly"/> |
| Pink | <asp:Label runat="server" SkinId="Pink"/> | <asp:TextBox runat="server" SkinId="Pink"/> | <asp:TextBox runat="server" SkinId="PinkReadOnly"/> |
| Yellow | <asp:Label runat="server" SkinId="Yellow"/> | <asp:TextBox runat="server" SkinId="Yellow"/> | <asp:TextBox runat="server" SkinId="YellowReadOnly"/> |
| Turquoise | <asp:Label runat="server" SkinId="Turquoise"/> | <asp:TextBox runat="server" SkinId="Turquoise"/> | <asp:TextBox runat="server" SkinId="TurquoiseReadOnly"/> |
| White | <asp:Label runat="server" SkinId="White"/> | <asp:TextBox runat="server" SkinId="White"/> | <asp:TextBox runat="server" SkinId="WhiteReadOnly"/> |
| Black | <asp:Label runat="server" SkinId="Black"/> | <asp:TextBox runat="server" SkinId="Black"/> | <asp:TextBox runat="server" SkinId="BlackReadOnly"/> |
| Intensified | <asp:Label runat="server" SkinId="Intensified"/> | <asp:TextBox runat="server" SkinId="Intensified"/> | <asp:TextBox runat="server" SkinId="IntensifiedReadOnly"/> |
| No attribute | <asp:Label runat="server" SkinId="Normal"/> | <asp:TextBox runat="server" SkinId="Normal"/> | <asp:TextBox runat="server" SkinId="NormalReadOnly"/> |

上記の表のセルに示される情報は、ASP.NET Web サーバーコントロールの設定を示しています。ASP.NETにより、多くのWebサーバーコントロールが定義されます。属性 runat="server" は、Web コントロールがサーバーサイトで解釈されることを示します。

テーマの使用 (IIS のみ)

デフォルトのスキンファイルでは、次のコントロールが使用されます。

| コントロール | 説明 |
|-------------|--------------------------|
| asp:Label | テキスト出力フィールド |
| asp:TextBox | テキスト入力フィールド |
| asp:Table | フィールド要素が PF キーボタンであるテーブル |
| asp:Button | PF キーボタン |
| asp:Image | イメージリンク |
| asp:Panel | 他の Web コントロールのコンテナ |

上記の色の定義以外に、SkinId は次の定義にも使用されます。

| Natural 要素 | 説明 | ASP.NET Skin ファイル |
|-------------|--------------------------|---|
| メッセージ行 | Natural 画面内の情報テキスト行。 | <code><asp:Label runat="server" SkinId="messageline"/></code> |
| 子ウィンドウ | Natural アプリケーションの子ウィンドウ。 | <code><asp:Panel runat="server" SkinId="WindowPanel"/></code> |
| 子ウィンドウのタイトル | Natural 子ウィンドウのタイトル。 | <code><asp:TextBox runat="server" SkinId="WindowTitle"/></code> |
| 子ウィンドウのシャドウ | 子ウィンドウはシャドウとともに表示されます。 | <code><asp:Panel runat="server" SkinId="WinShadow"/></code> |
| PF キーテーブル | PF キーボタンの行はテーブルで定義されます。 | <code><asp:Table runat="server" SkinId="PFKeys"/></code> |
| PF キーボタン | PF キーの定義。 | <code><asp:Button runat="server" SkinId="Pfkey"/></code> |
| 先頭画像 | ページの一番上に表示される画像。 | <code><asp:Image runat="server" SkinId="headpic"/></code> |

スタイルシート

スタイルシート (*StyleSheet.css*) では、CSS 標準に基づいてスタイルを定義できます。
<http://www.w3.org/Style/CSS/> を参照してください。

次の例は、Web ページの背景色とフォントを定義するスタイルシートを示しています。

```
body
{
background-color: transparent;
font-family: Courier New, Monospace;
}
```



Note: CSS ファイルでフォントサイズを定義しても、画面サイズには影響を与えません。フォントサイズを変更すると、それよりも小さいテキストのみが表示されます。

24 Web I/O インターフェイスクライアントと端末エミュレーションの違い

Web I/O インターフェイスクライアントは、ブラウザコントロール内で HTML 端末エミュレータとして動作します。Web I/O インターフェイスクライアントの表示の外観は通常の端末（エミュレーション）に非常に似ていますが、ブラウザ機能のため多少の違いがあります。

- 任意のフィールドでマウスポインタを使用してダブルクリックすると、Enter キーがシミュレートされます。
- マウスポインタを使用して出力専用フィールドにカーソルを置くことはできますが、返されるカーソル位置（システム変数 *CURSOR）は常にフィールドの最初の文字の位置になります。
- 入力フィールドおよび出力フィールドの範囲外にカーソルを置くことはできません。
- 1つの入力フィールド内でのみ、左矢印キーおよび右矢印キーを使用してカーソルを移動できます。他の矢印キーを使用した他のカーソル移動（出力フィールド内での移動、次の入力フィールドへの移動、上下移動など）はできません。
- Insert キーを使用して、挿入モードのオン／オフを切り替えることができます。
- Unicode 文字セット（タイプ U：中国語など）の場合、1文字が通常の英数字に比べてスペースを多く必要とする可能性があります。これは、Unicode 文字表示がプロポーショナルであるためです。Natural は固定幅の文字を基にしているため、アプリケーションの設計ではこの点を考慮する必要があります。入力フィールドの場合はフィールド内でのスクロールが可能です。出力フィールドの場合には Unicode 文字を表示するためのスペースが不十分な可能性もあります。フィールドの表示長は、セッションパラメータ DL で制御できます。
- 先行入力モードはサポートされていません。
- 上書きモードでの貼り付けはサポートされていません。
- キー体系は固定されているため、右側の Ctrl キーや数字パッドの Enter キーなどのキーを定義できなくなりました。
- 更新のみでなく完全な画面が送信されるため、画面更新速度は低下します。

- 点滅属性はサポートされていません。反転属性は、現時点では Microsoft の Internet Information Services (IIS) の実装でのみサポートされています。
- キー PF1～PF12 は、F1～F12 のキーの組み合わせによってシミュレートされます。
- キー PF13～PF24 は、Shift+F1～Shift+F12 のキーの組み合わせによってシミュレートされます。

IBM メインフレームのみ

- 端末画面のサイズは、Natural プロファイルパラメータ TMODEL によって制御されます。デフォルト設定の TMODEL=0 は、43 行で 132 列を意味しています。
- プログラムアテンションキー (PA1、PA2、および PA3) は、Ctrl+F1、Ctrl+F2、Ctrl+F3 のキーの組み合わせによってシミュレートされます。
- クリアキーは Ctrl+F4 によってシミュレートされます。
- ATTN (アテンション割り込み) キー、RESET キー、EEOF (エンドオブファイルの消去) キーはありません。

VT のみ

- I/O はブロックモードで行われます。したがって、Natural プログラムはファンクションキーが押された場合にのみ反応します。
- キー PF25～PF36 は、Ctrl+F1～Ctrl+F12 のキーの組み合わせによってシミュレートされます。
- キー PF37～PF48 は、Alt+F1～Alt+F12 のキーの組み合わせによってシミュレートされます。

25 Web I/O インターフェイスを Natural アプリケーションで使用する場合の制限

UNIX、メインフレーム、または Windows の各ホストで Web I/O インターフェイスを Natural アプリケーションで使用する場合、いくつかの制限があります。



Note: 「アプリケーション」という用語は、アプリケーションソフトウェアを指します。システムソフトウェアまたは開発用のソフトウェアを指すわけではありません。

次の制限が適用されます。

■ GUI コントロール

ダイアログ、ボタン、ラジオボタン、リストボックス、リストビュー、チェックボックスなどの GUI コントロールはサポートされていません。Web I/O インターフェイスでは、GUI コントロールなしで開発された Natural アプリケーションのみがサポートされます。

■ ファイル転送

ファイル転送（DOWNLOAD ステートメントを使用するものなど）は、Web I/O インターフェイスではサポートされていません。

■ ランタイムエラー

Natural アプリケーションでのランタイムエラーは、Web I/O インターフェイスでは処理されません。これにより、セッションが失われます。回避策：Natural システム変数 *ERROR-TA を使用してエラーを処理します。Natural エラートランザクションのサンプルは次のとおりです。

```
DEFINE DATA  
LOCAL  
1 ERR_INFO  
  2 ERR_NR(N5)  
  2 ERR_LINE(N4)  
  2 ERR_STAT(A1)  
  2 ERR_PNAM(A8)  
  2 ERR_LEVEL(N2)
```

```
END-DEFINE
INPUT ERR_INFO
DISPLAY ERR_INFO
STACK TOP COMMAND 'E'
END
```

 **Note:** 次の一部の制限はメインフレームでは動作する場合がありますが、実際のアプリケーションでは無意味です。

■ **Natural** メイン画面への復帰

Natural メイン画面に戻る Natural アプリケーションは使用できません。これを使用すると、常に間違った画面が表示され、セッションが失われます。

■ **Natural** エディタおよび **Natural** ユーティリティ

SYSMAIN や SYSDDM などの Natural ユーティリティおよびプログラムエディタなどのエディタは使用できません。これを使用すると、常に間違った画面が表示され、セッションが失われます。

■ **Natural** システムコマンド

CATALL、FIND、GLOBALS、HELP、KEY、LIST、SCAN XREF などの Natural システムコマンドは使用できません。これを使用すると、常に間違った画面が表示され、セッションが失われます。

■ **Natural** コマンド SETUP および RETURN

Natural コマンド SETUP および RETURN によりセッションが失われる可能性があるため、これらのコマンドは使用しないでください。

■ 端末コマンド

端末コマンドはサポートされていません。Web I/O インターフェイスクライアントで端末コマンドを入力しても、動作しません。

■ **Natural** システム変数 *INIT-ID

UNIX、メインフレーム、または Windows の各ホストで Web I/O インターフェイスを Natural アプリケーションで使用する場合、Natural システム変数 *INIT-ID には端末タイプの値は挿入されません。UNIX および Windows では、値 "notty" が埋められます。メインフレームでは、サーバー上で一意のセッション ID が埋められます。

26 NaturalX アプリケーションの管理

Windows プラットフォームでは、NaturalX クラスで構成されるアプリケーションを、DCOM を使用して複数のプロセスおよびマシンに配布することができます。

このセクションでは、次のトピックについて説明します。

- **NaturalX** サーバー
- アクティベーションポリシー
- 登録
- タイプ情報
- コンフィグレーション概要
- **NaturalX** でのセキュリティ
- **Windows** 上での **DCOM** コンフィグレーション
- **NaturalX** システムレジストリエントリ
- **NaturalX** サーバー環境でのステートメントおよびコマンドの使用

Windows プラットフォームでは、ライブラリ SYEXNXX にサンプルアプリケーションが用意されています。このアプリケーションを実行する方法については、ライブラリ SYEXNXX のテキスト A-README を参照してください。

『プログラミングガイド』の「NaturalX」も参照してください。

27 NaturalX サーバー

| | |
|---|-----|
| ■ COM クラスとサーバー | 194 |
| ■ NaturalX クラスとサーバー | 194 |
| ■ Windows での NaturalX サーバーと Natural セッション | 194 |
| ■ サーバー ID の役割 | 195 |
| ■ サーバー ID の編成 | 196 |

このchapterでは、次のトピックについて説明します。

COM クラスとサーバー

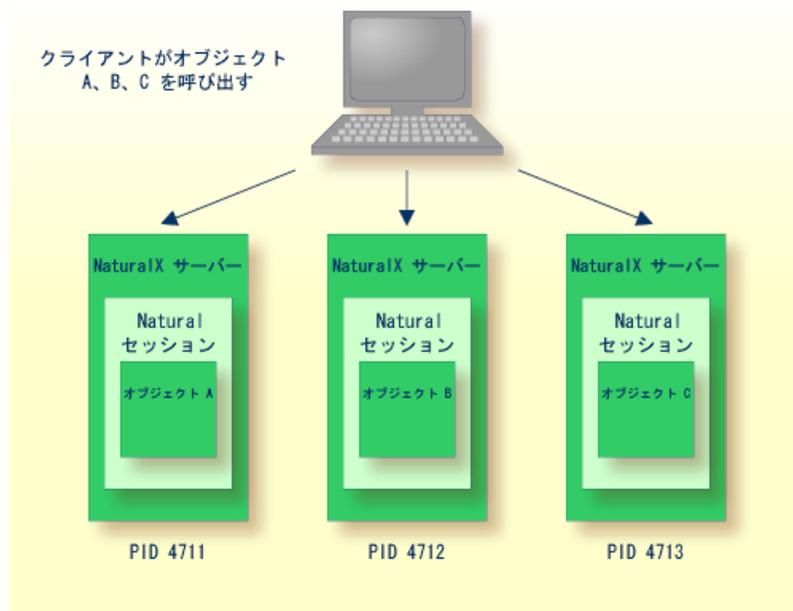
各 COM クラスは、サーバープロセスによってホストされる必要があります。サーバープロセスは、管理および技術的な役割を数多く果たします。例えば、クラスおよびそのインターフェイスを DCOM で使用可能にしたり、作成されたオブジェクトによって占有されているメモリを管理したりします。クライアントが特定クラスの新しいオブジェクトを要求するたびに、DCOM は対応するサーバープロセスがすでに実行されているかどうかをチェックします。実行されていない場合は、DCOM はサーバープロセスを開始し、要求をサーバーに渡します。サーバーが起動すると、そのクラスが DCOM で使用可能になります。稼働中のサーバーは、クライアント要求を実行して、オブジェクトを作成または削除したり、メソッドを実行したりします。サーバーで管理していた最後のオブジェクトが削除されると、サーバーは自動的にシャットダウンします。DCOM クラスとサーバーの詳細については、Microsoft DCOM 仕様を参照してください。

NaturalX クラスとサーバー

Natural で実装されたクラスは、DCOM クラスとしてアクセスできます。ただし、Natural では、クラスをホストするために DCOM サーバーを実装する必要はありません。代わりに、NaturalX 自体が DCOM サーバーのタスクを実行します。NaturalX は、Natural で記述されたすべてのクラス用の汎用 DCOM サーバーとして動作します。Natural クラス開発者が行うタスクは、クラスを実装し、それらを NaturalX サーバーに割り当てるだけです。

Windows での NaturalX サーバーと Natural セッション

Windows では、各 Natural セッションはそれ自体の排他的な NaturalX サーバープロセスで実行されます。



サーバー ID の役割

DCOMサーバーのタスクの1つは、起動時にそのクラスをDCOMで使用可能にすることです。NaturalXは汎用DCOMサーバーとして動作するので、NaturalXには提供するクラスの情報は組み込まれていません。代わりに、システムレジストリでサーバーIDのキーの下にあるクラスリストを確認します。サーバーIDはシステムレジストリ内にNaturalが所有するキーであり、特定のNaturalXサーバーに属するすべてのクラスがまとめられています。これは32文字の任意の英数字文字列で、空白を含まず、大文字と小文字は区別されません。

NaturalXサーバーは、実行に使用しているサーバーIDをどのように認識するのでしょうか。サーバーIDは、NaturalパラメータCOMSERVERIDで定義されます。このパラメータは、[ダイナミック](#)パラメータとしてNaturalXサーバーに渡されるか、Naturaパラメータファイルに定義されます。

クラスはどのようにサーバーIDに割り当てられるのでしょうか。あるサーバーIDでNaturalが起動されたとします。このNaturalセッション中にユーザーが登録するクラスはすべて、このサーバーIDでシステムレジストリに入力されます。

サーバーIDは、Naturalで作成されたクラスを分類して各NaturalXサーバープロセスに割り当てるのに使用されます。ただし、サーバーIDの使用は必須ではありません。サーバーIDなしでNaturalを起動した場合は、すべてのNaturalクラスは事前定義されたサーバーID "Default" で登録されます。

例

サンプルの `Employees` アプリケーションは、クラス `DepartmentList`、`EmployeesList`、および `Employee` で構成されています（このアプリケーションは、サンプルライブラリ `SYSEXCOM` にあります）。これらの3つのクラスは、`Employees` という名前の NaturalX サーバーによってホストされます。

1. 任意のサーバー ID を使用して Natural を起動します。
2. ライブラリ `SYSEXCOM` にログオンします。

```
LOGON SYSEXCOM
```

3. Natural コマンド行に `REGISTER` コマンドを入力してクラスを登録します。

```
REGISTER *
```

これで、3つのクラスはサーバー ID "Employees" で登録されました。

これらのクラスのオブジェクトが要求されるたびに、DCOM はサーバー ID "Employees" で NaturalX サーバープロセスを開始し、クラスを提供します。

サーバー ID の編成

サーバー ID は、対応する NaturalX サーバーの開始時に DCOM で使用可能になるすべてのクラスの集合を表しています。業務の観点から1つのアプリケーションを形成するクラスや、論理的に同属のクラスは、同じサーバー ID でグループ化することをお勧めします。同様に、絶対に同じコンテキストで使用されないクラスは、異なるサーバー ID で登録することをお勧めします。サーバー ID へのクラスの割り当てでは、セキュリティも基準にします（「[NaturalX でのセキュリティ](#)」を参照）。この点から、共通の権限が定義されるクラスを同じサーバー ID でグループ化することは意味があります。

28 アクティベーションポリシー

- Windows 環境のアクティベーションポリシー 198
- アクティベーションポリシーの設定 199
- アクティベーションポリシーを使用するタイミング 199

このchapterでは、次のトピックについて説明します。

Windows 環境のアクティベーションポリシー

あるクラスのオブジェクトの作成をクライアントが要求した場合は、DCOMが、クラスを提供するサーバープロセスを開始し、このプロセスに要求を渡します。Natural クラスを担当するサーバープロセスは NaturalX サーバーです。DCOM は、新しいサーバープロセスの開始時や実行中のサーバープロセスでのオブジェクトの作成時を制御するオプションを認識します。詳細については、「[登録](#)」を参照してください。Natural クラスを REGISTER コマンドで登録する場合は、DCOM がこのクラスに使用するアクティベーションオプションを制御できます。NaturalX では、DCOM でサポートされているさまざまなオプションを、次の3つのアクティベーションポリシーの形式で組み合わせます。

■ ExternalMultiple

Natural クラスがアクティベーションポリシー "ExternalMultiple" で登録され、クライアントがそのクラスのオブジェクトを要求すると、DCOM は要求されたオブジェクトを最初に現在のプロセスで作成しようとします。クライアント自身が NaturalX サーバーでもあり、クラス自体を提供できる場合があることに注意してください。現在のプロセスがクラス用のサーバーではない場合は、DCOM は新しい NaturalX サーバープロセスを開始し、そのプロセスでオブジェクトを作成します。後で同じクラスの2番目のオブジェクトが作成されると、このオブジェクトはそのサーバープロセスでも作成されます。つまり、1つのサーバープロセスはクラスの複数のオブジェクトを含むことができます。

■ ExternalSingle

Natural クラスがアクティベーションポリシー "ExternalSingle" で登録されると、このクラスのオブジェクトが作成されるたびに、DCOM は新しい NaturalX サーバープロセスを開始します。1つのサーバープロセスはクラスのオブジェクトを1つだけ含むことができます。

■ InternalMultiple

Natural クラスがアクティベーションポリシー "InternalMultiple" で登録されると、DCOM は常に現在のプロセスでこのクラスのオブジェクトを作成します。1つのサーバープロセスはクラスの複数のオブジェクトを含むことができます。

デフォルトのアクティベーションポリシーは "ExternalMultiple" です。このデフォルトは、Natura パラメータ ACTPOLICY で定義され、コンフィグレーションユーティリティで変更できません。

アクティベーションポリシーの設定

クラスのアクティベーションポリシーは、次の順に3通りの方法で設定できます。

- REGISTER コマンドの一部として明示的に
- DEFINE CLASS ステートメントで
- プロファイルパラメータ ACTPOLICY を使用して

アクティベーションポリシーを使用するタイミング

単純ではないDCOMアプリケーションは、ほとんどの場合、「パーシスタント」オブジェクト（データベースに保存されたオブジェクト）を処理します。このようなアプリケーションのために、データベースアクセス、トランザクション処理、およびユーザー分離についていくつか考慮する必要があります。次のシナリオを考えてみます。クライアント A と B の両方が、ある NaturalX サーバプロセスによって提供されるクラスのオブジェクトを作成します。そのオブジェクトをロードし保存するために、NaturalX サーバがデータベースを使用します。両方のクライアントが同じサーバプロセスによって処理される場合は、それらはデータベースに単一ユーザーとして表示されます。これは、クライアント A からのメソッド呼び出しで開始されたトランザクションを、クライアント B からのメソッド呼び出しでコミットまたはバックアウトできることを意味します。このような衝突は、明らかに回避する必要があります。

この衝突を回避するためのアプローチは2つあります。クライアントでパーシスタントオブジェクトを使用しないようにするか、クライアントがそれぞれ独自の NaturalX サーバプロセスによって処理されるようにします。どちらのアプローチにも状況によってそれぞれ利点があります。データベースまたは他の共有リソースにアクセスしないクラスまたはアプリケーションの場合は、単一サーバプロセスで複数のクライアントを処理するのは便利です。データベースまたは他の共有リソースにアクセスするクラスの場合は、異なるサーバプロセスで異なるクライアントを分離する必要があります。したがって、両方のアプローチが有効です。アクティベーションポリシーは、登録時にクラスごとに起動動作を制御する方法を管理者に提供します。

例

この例では、さまざまなアクティベーションポリシーをどのように使用できるかについて説明します。架空の旅行代理店アプリケーションの一部を例に取ります。アプリケーションには、ビジネスクラス Trip、Skipper、および RoutePlanner があります。Trip クラスは船旅の計画を表し、Skipper クラスは旅行を導くことができる船長を表します。RoutePlanner クラスは、旅行の最適なルートを決定します。Trip クラスと Skipper クラスは、オブジェクトの読み取りと保存にデータベースを使用します。RoutePlanner クラスは、特定の Trip オブジェクトに関する計算を実行するだけで、データベースは使用しません。

ビジネスクラスのいくつかがデータベースへのトランザクションアクセスを使用し、トランザクションが複数のメソッド呼び出しにわたる可能性があるため、アクティブな各クライアントは独自のNaturalXサーバープロセスで処理される必要があります。これは、追加のクラスSagToursを定義することによって実行できます。このクラスは、アプリケーションセッションを表します。このクラスはセッションステータスの全般的な情報の保持などに使用できますが、メインタスクはクライアントの代わりにビジネスオブジェクトを作成することです。

クラス SagTours

```
* Represents a SagTours application session.
*
define class SagTours
  local using tour-ids
  id clsid-sagtours
*
  interface Create /* Used to create application objects. */
    id iid-sagtours-create
*
    method newTrip /* Creates a new Trip object. */
      is trip-n
      parameter
      1 trip handle of object by value result
    end-method

    method newSkipper /* Creates a new Skipper object. */
      is skip-n
      parameter
      1 skipper handle of object by value result
    end-method
*
end-interface
*
end-class
end
```

このクラスは "ExternalSingle" として登録されます。これは、SagTours オブジェクトが作成されるたびに、オブジェクトを要求したクライアントに対して NaturalX サーバープロセスが開始されることを意味します。クライアントは SagTours オブジェクトを1回だけ作成し、後でそのメソッドを使用して必要なビジネスオブジェクトを作成します。Trip オブジェクトを作成するために、クライアントは次のように実装されるメソッド newTrip を呼び出します。

メソッド newTrip

```
* This method creates a new Trip object.
*
define data parameter
  1 trip handle of object by value result
end-define
*
create object trip of class "Trip"
*
end
```

Trip クラス自体は "**InternalMultiple**" として登録されます。これは、メソッド newTrip によって作成された Trip オブジェクトが、このクライアント用に開始された NaturalX サーバープロセスで作成されることを意味します。

クラス RoutePlanner を見てみましょう。

クラス RoutePlanner

```
* Plans optimal routes for sailing trips.
*
define class RoutePlanner
  local using tour-ids
  id clsid-planner
*
  interface routing
    id iid-planner-routing
*
    method plan /* Plans a sailing trip. */
      is plan-n
      parameter
        1 trip handle of object by value
      end-method
*
  end-interface
*
```

```
end-class
end
```

メソッド plan

```
* This method plans a sailing trip.
*
define data parameter
1 trip handle of object by value
end-define
*
* Perform some operations on the given Trip object.
*
end
```

このクラスは "ExternalMultiple" として登録できます。この場合は、異なるクライアントによって作成されたすべての RoutePlanner オブジェクトが、同じ NaturalX サーバードキュメントで作成されます。このクラスのメソッドがデータベースにアクセスしない場合、または各データベーストランザクションが1つのメソッド内に完全に含まれている場合（つまり、各メソッドサブプログラムが BACKOUT TRANSACTION ステートメントまたは END TRANSACTION ステートメントで終わる場合）は、他に影響しません。

サンプルクライアントプログラムを見てみましょう。

サンプルクライアントプログラム

```
define data local
  sagTours handle of object
  trip handle of object
  planner handle of object
end-define
*
* Start the application session.
create object sagTours of "SagTours"
*
* Create a Trip object.
send "newTrip" to sagTours return trip
* Create a RoutePlanner object.
create object planner of "RoutePlanner"
* Plan the trip.
send "plan" to planner with trip
*
end
```

クライアントは、まず SagTours オブジェクトを作成します。これは、このクライアント用に新しい NaturalX サーバードキュメントを排他的に開始します。クライアントは、次に SagTours オブ

ジェクトを使用して、このアプリケーションセッションのコンテキスト内に Trip オブジェクトを作成します。クライアントが RoutePlanner オブジェクトを直接作成することに注意してください。これは、クラスが "ExternalMultiple" として登録されるので可能ですが、必要というわけではありません。SagTours クラスは RoutePlanner オブジェクトを作成するメソッドも提供できます。その後、ビジネスオブジェクトが各自のジョブを実行します。オブジェクトはプログラム終了時に自動的に解放されます。SagTours オブジェクトが削除されると、NaturalX サーバーがシャットダウンします。



Note: この例では、アクティベーションポリシーの使用方法を説明するために必要な NaturalX の手法のみを示しています。実際のアプリケーションでは、これより多くを必要とします。クラスは、オブジェクトデータエリアを使用し、必ずグローバルユニーク ID が割り当てられています。また、インラインパラメータ宣言の代わりにパラメータデータエリアが使用されます。

29 登録

| | |
|-------------------------|-----|
| ■ Natural での登録 | 206 |
| ■ 自動登録 | 206 |
| ■ 手動登録 | 207 |
| ■ 登録ファイルとタイプライブラリ | 210 |
| ■ クライアントの登録 | 210 |
| ■ 登録のヒント | 211 |

クラスをDCOMクライアントからアクセスできるようにするには、クラスのいくつかの情報をシステムレジストリに追加する必要があります。DCOMクライアントは、ほとんどの場合、次の例のようにプログラム識別子 (ProgID) と呼ばれる意味のある名前です。

```
CREATE OBJECT #01 OF CLASS "Employee"
```

Natural クラスについては、DEFINE CLASS ステートメントに定義されたクラス名が、ProgID としてレジストリに書き込まれます。

システムレジストリエントリは、この ProgID をクラスのグローバルユニーク ID (GUID) にマップします。これにより、DCOMはクラスのすべての情報を一意に見つけることができます。レジストリに保存されるその他の情報には、担当DCOMサーバーのパスと名前、タイプライブラリのパスと名前、およびインターフェイス情報などがあります。

このchapterでは、次のトピックについて説明します。

Natural での登録

Natural クラスの登録 (または登録解除) は、システムコマンド REGISTER (または UNREGISTER) を使用して手動で行うか、クラスが格納 (または削除) された後に自動的に行うか、クラスが登録されるたびに生成される .reg ファイルを実行することによって行います。

クラスを登録するには、システムレジストリを修正できる権限が必要です。また、システム環境で COM を使用できる必要があります。

システムレジストリの Natural エントリをレジストリエディタで直接変更することは、レジストリエントリに矛盾が発生するおそれがあるため、通常はお勧めしません。

クラスは常に、Natural を起動したサーバー ID に登録されます。

自動登録

プロファイルパラメータ AUTOREGISTER を "ON" に設定すると、Natural クラスは格納 (カタログ) されるときに自動的に登録され、削除されるときに自動的に登録解除されます。これにより、ユーザーはクラスを格納した後にそのクラスを直接テストできます。

自動登録では、クラスの DEFINE CLASS ステートメントの WITH ACTIVATION POLICY 節に定義されたアクティベーションポリシー設定が使用されます。この節が指定されていない場合は、プロファイルパラメータ ACTPOLICY の設定が使用されます。

自動登録が設定されていて、クラスが格納 (カタログ) される場合は、古いレジストリエントリがすべて削除されるように、クラスは格納前に登録解除され、格納後に登録されます。

手動登録

以下では次のトピックについて説明します。

- REGISTER コマンド
- UNREGISTER コマンド

REGISTER コマンド

システムコマンド REGISTER を使用して、Natural クラスを登録します。それらは、Natural を起動したサーバー ID に登録されます。

```
REGISTER { class-module-name } [ [ { library-name } [ [ { ES } ] ] ] ]
          *
```

class-module-name

適切な Natural オブジェクトモジュール名を指定することによって登録されるクラス（複数可）を定義します。

library-name

クラスの検索先ライブラリ（複数可）を定義します。

ES、IM、EM

クラスに登録されるアクティベーションポリシーを定義します。

以下のいずれかのパラメータを設定できます。

| パラメータ | 説明 |
|-------|--|
| ES | アクティベーションポリシー "ExternalSingle" を設定します。 |
| IM | アクティベーションポリシー "InternalMultiple" を設定します。 |
| EM | アクティベーションポリシー "ExternalMultiple" を設定します。 |

登録

以下の表に、クラスが登録されるすべての有効なクラス/ライブラリの組み合わせを示します。

| クラスモジュール名指定 | ライブラリ名指定 | | |
|--------------------------|---|---|--|
| <i>library-name</i> | * | - | |
| <i>class-module-name</i> | ライブラリ <i>library-name</i> のクラスモジュール名 <i>class-module-name</i> のクラス | 現在のステップライブラリにあるクラスモジュール名 <i>class-module-name</i> のすべてのクラス | クラスモジュール名 <i>class-module-name</i> のクラス |
| * | ライブラリ <i>library-name</i> にあるすべてのクラスを登録 | 現在のステップライブラリにあるすべてのクラスを登録 | 現在のログオンライブラリのすべてのクラスを登録 |

このパラメータが REGISTER コマンドまたは DEFINE CLASS ステートメントに指定されていない場合は、パラメータファイルに定義されているデフォルトのアクティベーションポリシーが使用されます。

UNREGISTER コマンド

システムコマンド UNREGISTER を使用して、Natural クラスの登録を解除します。

```
UNREGISTER { class-module-name } [ [ { library-name } [server-id] ] ]  
*
```

class-module-name

適切な Natural オブジェクトモジュール名を指定することによって登録解除されるクラス（複数可）を定義します。

library-name

クラスの検索先ライブラリ（複数可）を定義します。

server-ID

クラスのサーバー ID を定義します。

以下の表に、クラスが登録解除されるすべての有効なクラス/ライブラリ/サーバー ID の組み合わせを示します。

| クラス名指定 | ライブラリ名/サーバー ID の組み合わせ | | | | |
|--------------------------|--|--|---|--|---|
| | -- | <i>library-name-</i> | <i>library-name</i> <i>server-ID</i> | *- | |
| <i>class-module-name</i> | 現在のサーバー ID に登録されている、現在のログオンライブラリにある <i>class-module-name</i> のクラス | 現在のサーバー ID に登録されている、ライブラリ <i>library-name</i> にある <i>class-module-name</i> のクラス | サーバー <i>server-ID</i> に登録されている、ライブラリ <i>library-name</i> にある <i>class-module-name</i> のクラス | 現在のサーバー ID に登録されている、現在のステップライブラリにある <i>class-module-name</i> のすべてのクラス | サーバー ID に登録されている、現在のステップライブラリにある <i>class-module-name</i> のすべてのクラス |
| * | 現在のサーバー ID に登録されている、現在のログオンライブラリにあるすべてのクラス | 現在のサーバー ID に登録されている、ライブラリ <i>library-name</i> にあるすべてのクラス | サーバー <i>server-ID</i> に登録されている、ライブラリ <i>library-name</i> にあるすべてのクラス | 現在のサーバー ID に登録されている、現在のステップライブラリにあるすべてのクラス | サーバー ID に登録されている、現在のステップライブラリにあるすべてのクラス |

class-module-name または *class-module-name* および *library-name* を指定した場合は、クラスまたはライブラリが見つからないと、REGISTER または UNREGISTER システムコマンドはエラーメッセージを返します。REGISTER または UNREGISTER システムコマンドにアスタリスク (*) のみを指定した場合は、登録または登録解除されるクラスがなくてもエラーメッセージは返されません。

クラス GUID またはインターフェイス GUID のないクラスを REGISTER システムコマンドに指定した場合は、エラーメッセージが返されます。このようなクラスはローカル Natural セッションでのみ使用できます。

 **Note:** Natural Security 環境では、このコマンドは 1 つのライブラリにのみ呼び出すことができます。これは、ライブラリ名を省略するか、特定のライブラリを使用する必要があることを意味します。アスタリスク (*) は使用できません。

登録ファイルとタイプライブラリ

登録ファイル（".reg" ファイル）を実行すると、情報がシステムレジストリに入力されます。

Naturalでは、クラスの登録時にサーバー用とクライアント側用の登録ファイルが自動的に作成されます。

サーバー用の".reg" ファイルには、システムレジストリに入力されたのと同じ情報が含まれます。クライアント用の".reg" ファイルには、クライアント側用に生成されたすべての情報が含まれます。クラスを登録解除すると、.reg ファイルは削除されます。登録解除時に".reg" ファイルが削除されないようにするには、クラスを登録解除する前に、ファイル名を変更します。Natural ではデフォルトの名前の".reg" ファイルのみが削除されます。

".reg" ファイルの名前は、サーバー用は `classmodule_name_S.reg`、クライアント用は `classmodule_name_C.reg`、異なるバージョンをアクティブにする場合は `classmodule_name_V.reg` です。

タイプライブラリは、クラスの登録時に自動的に作成され、クラスの登録解除時に削除されます。タイプライブラリへの参照はレジストリにも入力されます。

デフォルトのタイプライブラリ名は `classmodule_name.tlb` です。この名前のタイプライブラリがすでに存在する場合は、別の名前が生成されます。

登録ファイルとタイプライブラリは、次のように Natural *etc* ディレクトリに保存されます。

```
$NATDIR/$NATVERS/etc/serverid/classname/v<version-number>
```

例

サーバー ID "SERVER01" に登録されたクラス MY.TEST.CLASS のバージョン1のファイルは、次のように配置されます。

```
$NATDIR/$NATVERS/etc/SERVER01/MY.TEST.CLASS/v1
```

クライアントの登録

Naturalでは、クライアントの登録情報はシステムレジストリに自動的に入力されませんが、クライアント用の登録ファイルが作成されます。クライアント用の登録ファイルには、DCOM サーバークラスを持つマシンを DCOM に通知するエントリ (RemoteServerName) が含まれます。このエントリは Natural からは入力できません。次のいずれかの方法で入力できます。

1. 登録ファイルで RemoteServerName を入力できます。以下の行を見つけます。

```
"RemoteServerName"=
```

以下のように変更します。

```
"RemoteServerName"="server_machine_name"
```

変更したら、登録ファイルをクライアントマシンで実行します。

2. まず登録ファイルを実行し、次に DCOMCNFG ツールまたはレジストリエディタを使用して RemoteServerName を変更します（「[Windows 上での DCOM コンフィグレーション](#)」を参照）。

登録のヒント

クラスを登録および登録解除するときは、以下の点を考慮してください。

- 既存のクラスのクラス GUID は変更しないでください。すでにレジストリにあるクラスを別の GUID で再度登録すると、Natural にエラーメッセージが表示されます。この場合は、古いクラスを先に登録解除する必要があります。
- 1つのクラスを複数のサーバー ID に登録しないでください。サーバー ID と AppID は 1 対 1 の関係になっており、クラスには AppID を 1 つだけ定義できます。2 つ目のサーバー ID に登録すると、AppID が上書きされます。また、クラスをサーバー ID から登録解除すると、クラスのすべてのエントリは、2 つ目のサーバーに登録されたものかどうかチェックされずに削除されます。
- クライアント登録を除き、常に Natural システムコマンド REGISTER および UNREGISTER を使用して、クラスのレジストリエントリを変更してください。これらのコマンドを使用すると、重複するレジストリエントリを削除できます。

例えば、クライアントクラスが "server1" に登録されている場合に、"server2" の同じクラスが登録されたサーバー登録ファイルを実行すると、クラスの AppID キーが変更され、古い AppID キーへのすべての参照が失われます。この場合は、この古い AppID キーを削除できません。クラスをシステムコマンド REGISTER で登録すると、AppID が変更されたかどうかチェックされ、他のクラスで必要とされていない古い AppID が削除されます。

- クライアントマシンで Natural を使用できない場合に、Natural クラスのレジストリエントリをシステムレジストリから削除するには、レジストリエディタを使用してください。クライアントマシンで Natural を使用できる場合は、最初に Natural システムコマンド REGISTER でクラスを登録し、後でシステムコマンド UNREGISTER でそれを登録解除する方が簡単です。
- クラスの登録情報はカタログされたクラスオブジェクトから取得されるので、ソースフォーマットのみで使用可能なクラスを登録したり登録解除したりすることはできません。

- Natural セッション中にクラスを登録する場合は、以下に示すように、セッションをパラメータ PARM および COMSERVERID で開始する必要があります。これは、これらの2つのパラメータのみがレジストリキー "LocalServer32" に保存されるためです。他のパラメータ設定でテストしたクラスは、後で DCOM クライアントから開始したときに実行される保証はありません。

```
NATURAL.EXE PARM=COMPARM COMSERVERID=SERVER1
```

- 通常は、管理者権限を持つユーザーのみがシステムレジストリを変更できます。クラスを登録しようとするエラーになる場合は、レジストリの変更に必要な権限を所有しているかどうかを確認してください。
- Natural クラスの登録時に、Natural のみに必要 (DCOM には不要) な追加情報がレジストリに入力されます。追加レジストリキーに保存される情報は、クラスのサーバー ID (「[NaturalX サーバー](#)」を参照)、アクティベーションポリシー (「[アクティベーションポリシー](#)」を参照)、および場所 (クラスの Natural クラスモジュール名とライブラリ) です。この情報は、例えば、指定されたサーバー ID のすべてのクラスを登録解除したり、Natural の起動時に提供されたクラスを使用可能にしたりする場合に必要です。
- クラスのサーバー ID と AppID (HKEY_CLASSES_ROOT/AppID) は 1 対 1 の関係です。クラスを新しいサーバー ID に登録すると、新しい GUID (AppID) が生成されてこのサーバー ID に割り当てられます。AppID は、DCOM クラスをグループ化するために DCOM によって使用されます。セキュリティ設定および (クライアント登録のための) リモートマシン名は、AppID に定義されます。つまり、1 つの AppID に属しているすべてのクラスは、同じセキュリティ設定になります (「[コンフィグレーション概要](#)」および「[NaturalX でのセキュリティ](#)」を参照)。

30 タイプ情報

- 概要 214
- NaturalX とタイプ情報 214
- タイプ情報の使用 214

このchapterでは、次のトピックについて説明します。

概要

タイプ情報は、そのインターフェイスのすべてとともに（メソッドの名前とタイプにいたるまで）クラスを完全に記述するための方法です。それはクラスおよびそれらのインターフェイス、例えば、インターフェイスが存在するクラス、それらのインターフェイスに存在するメンバ関数、およびそれらの関数が必要とする引数についての必要な情報を含みます。

この情報は、例えば、エンドユーザーに有効なオブジェクト、インターフェイス、メソッドおよびプロパティを表示するためのタイプ情報ブラウザで、クラスとそのメソッドについての詳細を見つけ出すためにクライアントによって使用されます。

他にタイプ情報の使用で重要な点は、NaturalXも使用する、広く使われているOLEオートメーションテクニックです。

タイプ情報を保存するには、いくつかの方法があります。共通の方法は、タイプライブラリ（.TLB）ファイルにタイプ情報を生成することです。

NaturalX とタイプ情報

タイプ情報の作成

Natural クラスの登録時に、クラスごとにタイプライブラリファイルが作成されます。

タイプライブラリは、`$NATDIR/$NATVERS/etc/<serverid>/<classname>/<version>` ディレクトリに生成され、レジストリのエントリ経由でクラスに接続されます。

タイプライブラリファイル名には、既存の名前と競合しない限り、クラスモジュール名が使用され、拡張子 ".tlb" が付きます。クラスモジュール名には番号が付加されます。

タイプ情報の使用

Natural クラスに定義された各インターフェイスは、クライアントではダイナミックインターフェイスとして参照されます（「ディスパッチインターフェイス」とも呼ばれます）。インターフェイスの各メソッドは、クライアントでは METHOD ステートメントに定義された名前でも参照されます。

Natural クラスの最初のインターフェイスは、デフォルトのディスパッチインターフェイスとしてマークされます。

タイプ情報のサポートにより、複数のインターフェイスを同じメソッド名またはプロパティ名で定義することもできます。Natural クライアントは、以下の例のように、インターフェイス名（Natural クラスに定義）をメソッド名の接頭語として使用して、対応するメソッドをアドレスします。

```
CREATE OBJECT #03 OF CLASS "DepartmentList"
SEND "Iterate.PositionTo" TO #03 WITH "C" RETURN #DEPT
```

Natural クライアントは、タイプ情報を使用して、メソッドまたはプロパティが属しているインターフェイスを検出します。

 **Note:** Natural クライアントは、構文チェックの実行にカタログ時のタイプ情報を使用しません。

データタイプ変換

以下では次のトピックについて説明します。

- [Natural データフォーマットから OLE タイプ](#)
- [OLE タイプから Natural データフォーマット](#)

Natural データフォーマットから OLE タイプ

クライアントからデータを受け取るため、または他のプログラミング言語で記述されたクラスにデータを渡すために、Natural データフォーマットはいわゆる OLE オートメーション互換タイプに変換されます。この表は、COM クライアントが Natural クラスのメソッドパラメータまたはプロパティをどのように参照するかを示します。例えば、Natural クラスにフォーマット A のメソッドパラメータまたはプロパティがある場合は、これは COM クライアントによって VT_BSTR として参照されます。

| Natural データフォーマット | オートメーション互換タイプ |
|-------------------|--------------------|
| A | VT_BSTR |
| B1 | VT_UI1 |
| B2 | VT_UI2 |
| B4 | VT_UI4 |
| Bn (n != 1, 2, 4) | VT_UI1 の SAFEARRAY |
| C | サポートされません |
| D | VT_DATE |
| F4 | VT_R4 |
| F8 | VT_R8 |
| I1 | VT_I2 |
| I2 | VT_I2 |

| Natural データフォーマット | オートメーション互換タイプ |
|------------------------------------|---------------|
| I4 | VT_I4 |
| GUI のハンドル | サポートされません |
| HANDLE OF OBJECT | VT_DISPATCH |
| L | VT_BOOL |
| N15.4 | VT_CY |
| N <i>n.m</i> (<i>n.m</i> != 15.4) | VT_R8 |
| P15.4 | VT_CY |
| P <i>n.m</i> (<i>n.m</i> != 15.4) | VT_R8 |
| T | VT_DATE |
| U | VT_BSTR |

指定された Natural データフォーマットの配列は、対応する "VT" タイプの SAFEARRAY にマップされます。

ただし、次のような特別な場合があります。

- 固定長でフォーマット B*n* (*n*は、1、2、4 以外、またはそのような変数の配列ではない) の変数は、VT_UI1 の 1 次元 SAFEARRAY にマップされます。これは、ラージ変数およびダイナミック変数がまだサポートされていない、以前のバージョンの Natural との互換性のためです。したがって、ラージバイナリ変数は固定長でタイプ B の変数の配列によってシミュレートする必要があります。
- フォーマット B のダイナミック変数は、VT_UI1 の 1 次元 SAFEARRAY にマップされます。
- フォーマット B のダイナミック変数の配列は、それぞれが VT_UI1 の 1 次元 SAFEARRAY を含むバリエーションの SAFEARRAY にマップされます。
- 属性制御変数はマップされません。それらは Natural 以外では意味がありません。フォーマット HANDLE OF GUI の変数もマップされません。対応するオートメーション互換タイプはありません。したがって、属性制御変数または HANDLE OF GUI フォーマットのプロパティは、COM/DCOM を介してクライアントにアクセスされません。これらのタイプのメソッドパラメータは、COM/DCOM を介してメソッドを呼び出すときにクライアントがパラメータを省略できるように、パラメータデータエリアでオプションとしてマークする必要があります。

OLE タイプから Natural データフォーマット

この表は、外部クラスのパラメータまたはプロパティを Natural でどのようにアドレスできるかを示します。例えば、外部クラスにタイプ VT_R4 のメソッドパラメータまたはプロパティがある場合は、このパラメータまたはプロパティは、F4 または F4 への MOVE 互換性のあるフォーマットとして Natural でアドレスできます。

| オートメーション互換タイプ | Natural データフォーマット |
|---------------|-------------------|
| VT_BOOL | L |
| VT_BSTR | A または U |
| VT_CY | P15.4 |
| VT_DATE | T |
| VT_DISPATCH | HANDLE OF OBJECT |
| VT_UNKNOWN | HANDLE OF OBJECT |
| VT_I1 | I1 |
| VT_I2 | I2 |
| VT_I4 | I4 |
| VT_INT | I4 |
| VT_R4 | F4 |
| VT_R8 | F8 |
| VT_U1 | B1 |
| VT_U2 | B2 |
| VT_U4 | B4 |
| VT_UINT | B4 |

最大 3 次元の SAFEARRAY は、同じ次元数および対応するフォーマットの Natural 配列に変換されます。3 次元を超える SAFEARRAY は、Natural 内からは使用できません。

ただし、次のような特別な場合があります。

- VT_BSTR は、フォーマット A の Natural 変数、または固定長でフォーマット A の Natural 変数の 1 次元配列にマップされます。253 文字を超える文字列を保存するために、追加の次元が使用されます。これは、ラージ変数およびダイナミック変数がまだサポートされていない、以前のバージョンの Natural との互換性のためです。このマッピングは、これ以降は使用できません。代わりに、フォーマット A のダイナミック変数を使用する必要があります。
- VT_BSTR の SAFEARRAY は、同じ次元数でフォーマット A の Natural 変数の配列、または 1 次元上で固定長でフォーマット A の Natural 変数の配列にマップします。253 文字を超える文字列を保存するために、追加の次元が使用されます。これは、ラージ変数およびダイナミック変数がまだサポートされていない、以前のバージョンの Natural との互換性のためです。このマッピングは、これ以降は使用できません。代わりに、フォーマット A のダイナミック変数の配列を使用する必要があります。

- VT_UI1 の SAFEARRAY は、全体のサイズが一致している固定長でフォーマット B の Natural 変数の配列にマップできます。これは、レンジ変数およびダイナミック変数がまだサポートされていない、以前のバージョンの Natural との互換性のためです。このマッピングは、これ以降は使用できません。代わりに、フォーマット B のダイナミック変数を使用する必要があります。

31 コンフィグレーション概要

- サーバーのコンフィグレーション - 一般的な設定 220
- サーバーのコンフィグレーション - アプリケーション固有の設定 221
- クライアントのコンフィグレーション - 一般的な設定 222
- クライアントのコンフィグレーション - アプリケーション固有の設定 222

アプリケーションのすべてのクラスがクライアントおよびサーバーマシンに登録されたら、アプリケーションの動作の特定の面をシステムレジストリ設定で制御および構成できます。このセクションでは、NaturalX アプリケーションに関連するレジストリエントリとその意味について簡単に説明します。レジストリキーおよびその管理についての詳細な予備知識については、該当するプラットフォームの特定の DCOM レジストリに関するドキュメントを参照してください。

このコンテキストに関連するレジストリキーは、DCOMCFG やレジストリエディタ (REGEDIT) のような一般に使用されるツールで管理します。ツールによってレジストリキーの表示方法が異なります。ここでは、レジストリキーの名前のみを挙げます。レジストリキーの設定法については、「[Windows 上での DCOM コンフィグレーション](#)」を参照にしてください。

 **Note:** "HKLM" は、`HKEY_LOCAL_MACHINE` の一般的な短縮形です。"HKCR" は、`HKEY_CLASSES_ROOT` の短縮形です。

このchapterでは、次のトピックについて説明します。

サーバーのコンフィグレーション - 一般的な設定

このセクションでは、一般的なサーバーコンフィグレーションの設定について説明します。

- レジストリエントリ `HKLM\Software\Microsoft\OLE\EnableDCOM` は、DCOM を経由してサーバーマシンにアクセスできるようにするために、"Y" に設定する必要があります。
- ゲスト (サーバーマシンに自分のアカウントを持っていないユーザー) がサーバーマシン上のアプリケーションにアクセスできるようにするには、事前定義したアカウント "Guest" をユーザーマネージャで有効にする必要があります (Windows 2000 のみ)。
- レジストリエントリ `HKLM\Software\Microsoft\OLE\DefaultLaunchPermissions` および `HKLM\Software\Microsoft\OLE\DefaultAccessPermissions` は、DCOM アプリケーションの起動およびそのクラスへのアクセスを許可されるまたは許可されないユーザーやグループを定義します。ここで定義された権限は、アプリケーション固有の設定が定義されていないすべてのアプリケーションに適用されます。
- レジストリエントリ `HKLM\Software\Microsoft\OLE\LegacyAuthenticationLevel` は、このマシン上の DCOM アプリケーションにアクセスするクライアントに対して行われる認証のレベルを制御します。NaturalX サーバーがクライアントのユーザー ID を Natural Security に渡すことができる場合は、設定は最低でも "Connect" にする必要があります。認証を行わない場合は、"None" を選択します。この場合は、NaturalX サーバーはクライアントのユーザー ID を取得しません。代わりに、サーバーの起動に使用されたユーザー ID で各要求を実行します。このエントリがクライアント側とサーバー側で異なって定義されている場合、より厳密な設定が適用されます。
- レジストリエントリ `HKLM\Software\Microsoft\OLE\LegacyImpersonationLevel` は、クライアントについてサーバーが取得する情報量を制御し、また、他のサーバーに対してクライアントの役割で動作するためにこの情報を使用するかどうかを制御します。NaturalX サーバーがクラ

クライアントのユーザー ID を Natural Security に渡すことができる場合は、設定は最低でも "Identify" にする必要があります。設定 "Impersonate" または "Delegate" は、NaturalX サーバーに対して同じ効果があります。サーバーがクライアントのユーザー ID を取得できない場合は、"Anonymous" を選択します。この場合は、サーバーはサーバーの起動に使用されたユーザー ID で各要求を実行します。このエントリがクライアント側とサーバー側で異なって定義されている場合、より厳密な設定が適用されます。

サーバーのコンフィグレーション・アプリケーション固有の設定

NaturalX アプリケーションごとに、アプリケーション固有の設定を別々に設定できます。ただし、これらの設定を適用する場所について考慮する必要があります。1つの NaturalX サーバー ID で登録されたすべてのクラスは、DCOM の意義で 1つのアプリケーションを形成するので、レジストリで 1つの AppID キーに割り当てられます。アプリケーション固有の設定が AppID キーで適用されるのは、このためです。

- レジストリエントリ `HKCR\AppID\<APPID>\LaunchPermission` および `HKCR\AppID\<APPID>\AccessPermission` は、指定した AppID で DCOM アプリケーションを起動してそのクラスにアクセスすることを許可されるまたは許可されないユーザーやグループを定義します。
- レジストリエントリ `HKCR\AppID\<APPID>\RunAs` は、DCOM によって NaturalX サーバーが起動されるときにそのサーバーが実行するユーザーアカウントを定義します。3つのオプションがあります。
 - **Interactive user** :
NaturalX サーバーは、サーバーマシンに対話形式でログインするユーザーアカウントで起動されます。これは通常は望ましくはありませんが、テスト目的に役立ちます。
 - **Launching user** :
NaturalX サーバーは、このサーバーに最初のオブジェクトを作成するクライアントのアカウントで起動されます（オブジェクトに対する最初の要求が DCOM にサーバーの起動を強制します）。各クライアントが独自のサーバープロセスで処理される場合は、この設定を使用します。当然、クライアントはサーバーを起動する権限を持っている必要があります。
 - **This user** :
サーバーは、指定したユーザーのアカウントで起動されます。すべてのクライアントが同じサーバープロセスで処理される場合は、この設定を使用します。ここに入力されたユーザーはサーバーを起動する権限を持っている必要があります。

クライアントのコンフィグレーション - 一般的な設定

このセクションでは、一般的なクライアントコンフィグレーションの設定について説明します。

- レジストリキー `HKLM\Software\Microsoft\OLE\LegacyAuthenticationLevel` は、このマシンで実行されているクライアントが DCOM アプリケーションにアクセスするときに行われる認証のレベルを制御します。NaturalX サーバーにアクセスするクライアントには、「[サーバーのコンフィグレーション - 一般的な設定](#)」と同様の考慮が適用されます。最低でも "Connect" が指定された場合にのみ、NaturalX サーバーはそのユーザー ID を Natural Security に対して使用できます。このエントリがクライアント側とサーバー側で異なって定義されている場合、より厳密な設定が適用されます。
- レジストリキー `HKLM\Software\Microsoft\OLE\LegacyImpersonationLevel` は、クライアントについてサーバーが取得する情報量を制御し、また、他のサーバーに対してクライアントの役割で動作するためにこの情報を使用するかどうかを制御します。NaturalX サーバーにアクセスするクライアントには、「[サーバーのコンフィグレーション - 一般的な設定](#)」と同様の考慮が適用されます。最低でも "Identify" が指定された場合にのみ、NaturalX サーバーはそのユーザー ID を取得し Natural Security に対して使用できます。このエントリがクライアント側とサーバー側で異なって定義されている場合、より厳密な設定が適用されます。

クライアントのコンフィグレーション - アプリケーション固有の設定

NaturalX アプリケーションごとに、アプリケーション固有の設定を別々に設定できます。ただし、これらの設定を適用する場所について考慮する必要があります。1つの NaturalX サーバー ID で登録されたすべてのクラスは、DCOM の意義で1つのアプリケーションを形成するので、レジストリで1つの AppID キーに割り当てられます。アプリケーション固有の設定が AppID キーで適用されるのは、このためです。

- レジストリキー `HKCR\AppID\<APPID>\RemoteServerName` は、このサーバーによってホストされるクラスが要求されたときに DCOM がサーバーを起動するリモートマシンを定義します。サーバーがローカルで開始される場合は、"Run on this computer" を指定します。RemoteServerName は指定しないでください。

32 NaturalX でのセキュリティ

| | |
|-------------------------|-----|
| ■ 概要 | 224 |
| ■ アクティベーションセキュリティ | 224 |
| ■ コールセキュリティ | 225 |

このchapterでは、次のトピックについて説明します。

NaturalX の設定方法については、「[Windows 上での DCOM コンフィグレーション](#)」を参照してください。

概要

分散環境では、セキュリティは特に重要なトピックです。サーバーでは、サーバーが提供するサービスを未認可クライアントに使用されないことを確実にする必要があります。クライアントでは、想定サーバーに接続され、そのサーバーがクライアントの権限を誤用しないことを確実にする必要があります。

DCOM のコンテキストでは、セキュリティは2つのレベルに区別できます。

- アクティベーションセキュリティは、クラスを提供するサーバープロセスの開始とアクセスを許可されるユーザーを制御します。
- コールセキュリティは、クラスが提供する各メソッドの使用の許可されるユーザーを制御します。

多くの場合は、権限の定義にはアクティベーションセキュリティで十分です。このセキュリティレベルは、Windows セキュリティを基準に DCOM 自体によってサポートされます。必要な権限はシステムレジストリで管理されます。詳細については、「[アクティベーションセキュリティ](#)」を参照してください。

その他の場合は、各メソッドのレベルで権限をより詳細に制御する必要があります。このセキュリティレベルは、レジストリ定義では管理できません。これは、Natural Security の支援で NaturalX によって提供されます。詳細については、「[コールセキュリティ](#)」を参照してください。

アクティベーションセキュリティ

このセクションでは、次のトピックについて説明します。

- [アプリケーション](#)

■ レジストリ使用の権限

アプリケーション

アクティベーションセキュリティは、サーバープロセスの開始とアクセスを許可されるユーザーを制御します。これは、基本的に、各クラスに対して権限を定義することによって行われます。ただし、実用的な理由と、管理の労力を削減するために、権限は通常はアプリケーションレベルで管理されます。システムレジストリでは、各アプリケーションは AppID によって定義されます。AppID は、アプリケーションの権限が管理されるキーです。これらの権限を管理するために、各 DCOM 対応プラットフォームでは DCOMCNFG ツールを使用できます。このツールは、他の DCOM アプリケーションと同様に、NaturalX アプリケーションでも使用できます。

NaturalX での AppID の意味を理解するために、NaturalX がクラスをどのようにアプリケーションに編成するかを思い出してください（「[サーバー ID の編成](#)」を参照）。Natura パラメータ COMSERVERID を使用して、特定の NaturalX サーバーに名前を付けることができます。COMSERVERID で指定した値で Natural が起動された場合は、この Natural セッション中に登録されるすべての Natural クラスはこのサーバー ID で登録されます。同時に、それらはすべてシステムレジストリで同じ AppID キーで登録されます。これは、*server-ID* の複数の値がそれぞれ、システムレジストリで異なる AppID キーに対応していることを意味します。

例えば、Natural がサーバー ID "Employees" で実行されているとします。この Natural セッション中に登録されたすべてのクラスは、"Employees" アプリケーションを形成します。REGISTER コマンドは、それらのクラスをすべて 1 つの AppID キー ("Employees" アプリケーションに対応) で登録します。

レジストリ使用の権限

アクティベーションセキュリティの設定には、レジストリキー *LaunchPermissions*、*AccessPermissions*、*DefaultLaunchPermissions*、および *DefaultAccessPermissions* が関連します。キー *DefaultLaunchPermissions* および *DefaultAccessPermissions* は、レジストリにそれぞれ 1 つだけ存在し、個々の権限が定義されていないすべてのアプリケーションの権限を定義します。キー *LaunchPermissions* および *AccessPermissions* は、各アプリケーション（各 AppID）に対して存在し、各アプリケーションの権限を定義します。

コールセキュリティ

この section では、次のトピックについて説明します。

■ Natural Security 使用の権限

■ セキュリティのヒントと提案

Natural Security 使用の権限

コールセキュリティは、クラスが提供する各メソッドの使用の許可されるユーザーを制御します。このレベルの権限は、レジストリ定義では管理できません。コールセキュリティは、Natural Security の支援で NaturalX によって提供されます。

コールセキュリティがどのように Natural Security で達成されるかを理解するには、NaturalX のクラスがどのように実装されるかを考えてください。各クラスはクラスタイプの Natural モジュールであり、各メソッドはサブプログラムタイプの Natural モジュールです。すべての Natural モジュールについて、Natural Security で定義された権限で実行を制御できます。これを行う方法については、『Natural Security』ドキュメントを参照してください。

クラスモジュールを使用してオブジェクトが作成される時、およびメソッドサブプログラムがメソッド呼び出しに回答して実行される時は、クラスモジュールとメソッドサブプログラムに対して定義された権限が評価されます。次の規則が適用されます。クラスモジュールの実行を許可されているユーザーが、そのクラスのオブジェクトの作成を許可されます。メソッドサブプログラムの実行を許可されているユーザーが、対応するメソッドの使用を許可されます。

必要な権限チェックを実行するために、NaturalX サーバーはクライアントのユーザー ID を認識している必要があります。ユーザー ID を確実に信頼できる必要もあります。したがって、コールセキュリティを使用するには、以下の要件を満たしている必要があります。

- クライアントは、ローカルマシンまたは Windows ドメインサーバーのログオンでクライアント自体を識別する必要があります。
- *Authentication level* は、最低でも "Connect" に設定する必要があります（クライアントまたはサーバーマシン上）。
- *Impersonation level* は、最低でも "Identify" に設定する必要があります（クライアントまたはサーバーマシン上）。

上記の要件が満たされている場合は、要求を処理する NaturalX サーバーがクライアントのユーザー ID を取得し、それを Natural システム変数 *USER に挿入します。その後、要求は、必要なすべての Natural Security 権限チェックを含めて、このユーザー ID で実行されます。要求が処理されると、Natural システム変数 *USER は NaturalX サーバーの起動時の値に戻ります。

要件のいずれかが満たされていない場合は、*USER は要求の実行中には変更されません。その後、要求は、NaturalX サーバーを起動したユーザー ID で実行されます。

*USER に加えて、システム変数 *NET-USER も要求の実行中に取得されます。Windows ドメインに属しているクライアントのドメイン名で修飾されたユーザー ID が含まれており、追加のアプリケーション固有セキュリティチェックに使用できます。

セキュリティのヒントと提案

Natural Security で NaturalX を使用するときは、以下の点を考慮してください。

- Natural Security 環境では、NaturalX サーバーは Natural パラメータ `AUTO=ON` で起動する必要があります。これは、クライアント側ですでに認証が行われているからです。設定は、Natural パラメータファイルに入力する必要があります。
- Natural Security 環境では、NaturalX サーバーを常に特定のユーザー ID で起動することをお勧めします。このユーザー ID は、認証されていないユーザーのすべての要求に対して自動的に使用されます。このユーザー ID の最小権限は、Natural Security 管理者が定義します。
- Natural と Natural Security では、8 文字を超えるユーザー ID や空白を含むユーザー ID は処理できません。

33 Windows 上での DCOM コンフィグレーション

| | |
|----------------------------|-----|
| ■ NaturalX サーバーの構成 | 230 |
| ■ NaturalX クライアントの構成 | 240 |

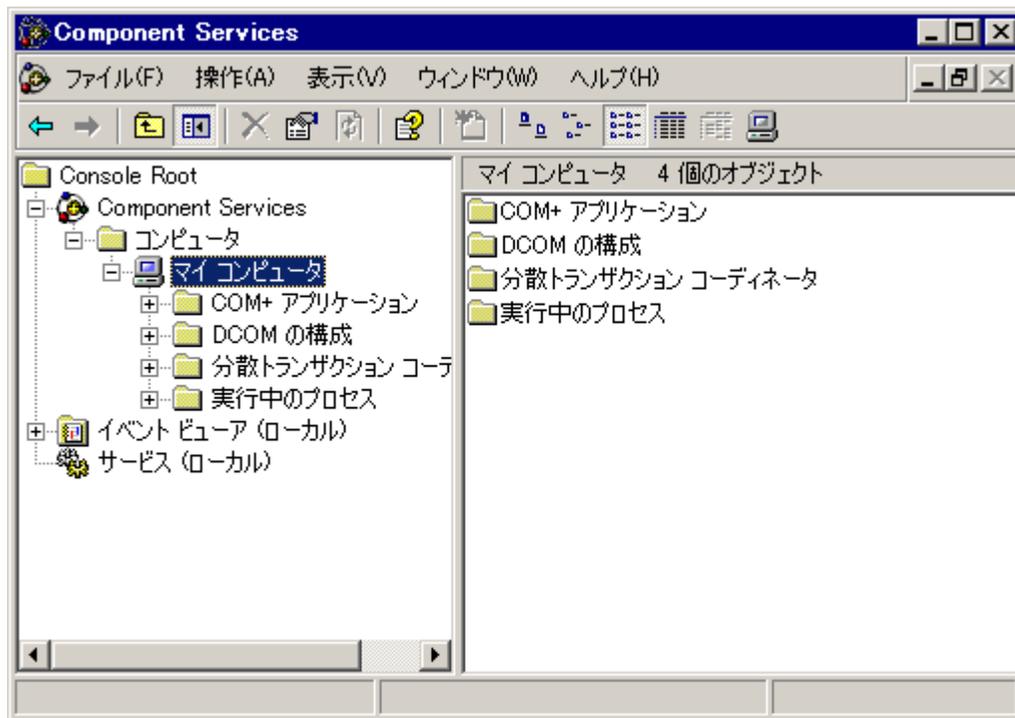
このchapterでは、Windows 2000/XP で NaturalX アプリケーションを設定する方法について説明します。すべての設定は、ツール *DCOMCNFG.EXE* (コンポーネントサービス) を使用して適用します。以降のセクションで表示されるダイアログボックスの例は、Windows XP のコンポーネントサービスです。

このchapterでは、次のトピックについて説明します。

NaturalX サーバーの構成

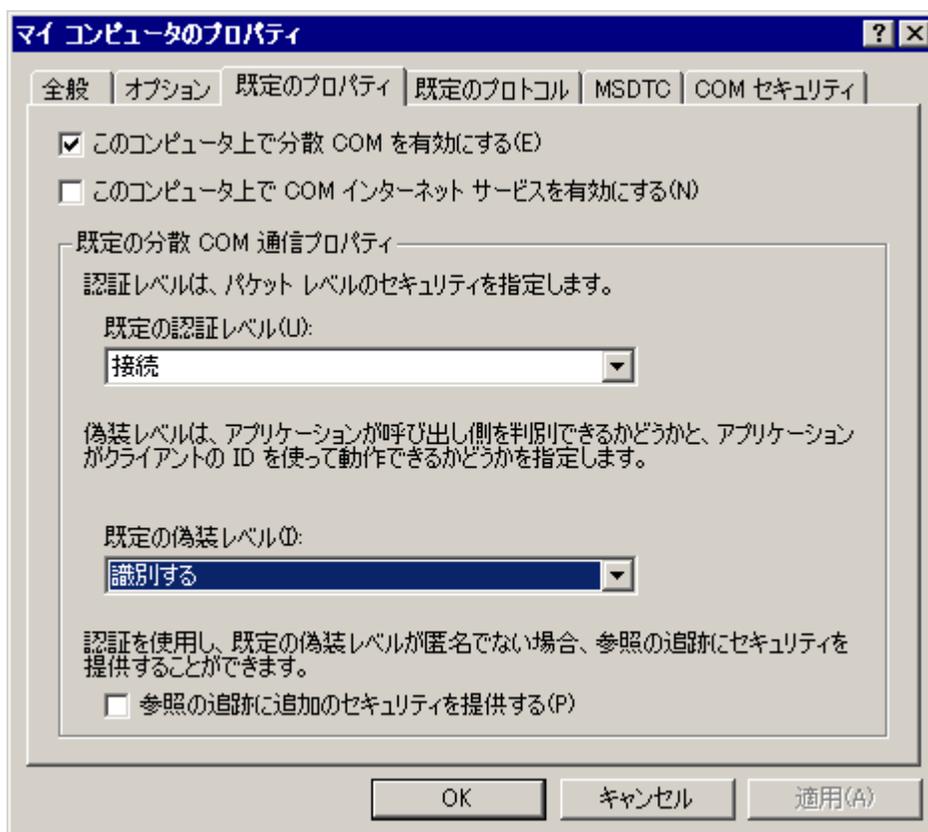
▶手順 33.1. NaturalX サーバーを構成するには

- 1 [コンポーネントサービス] を起動します。



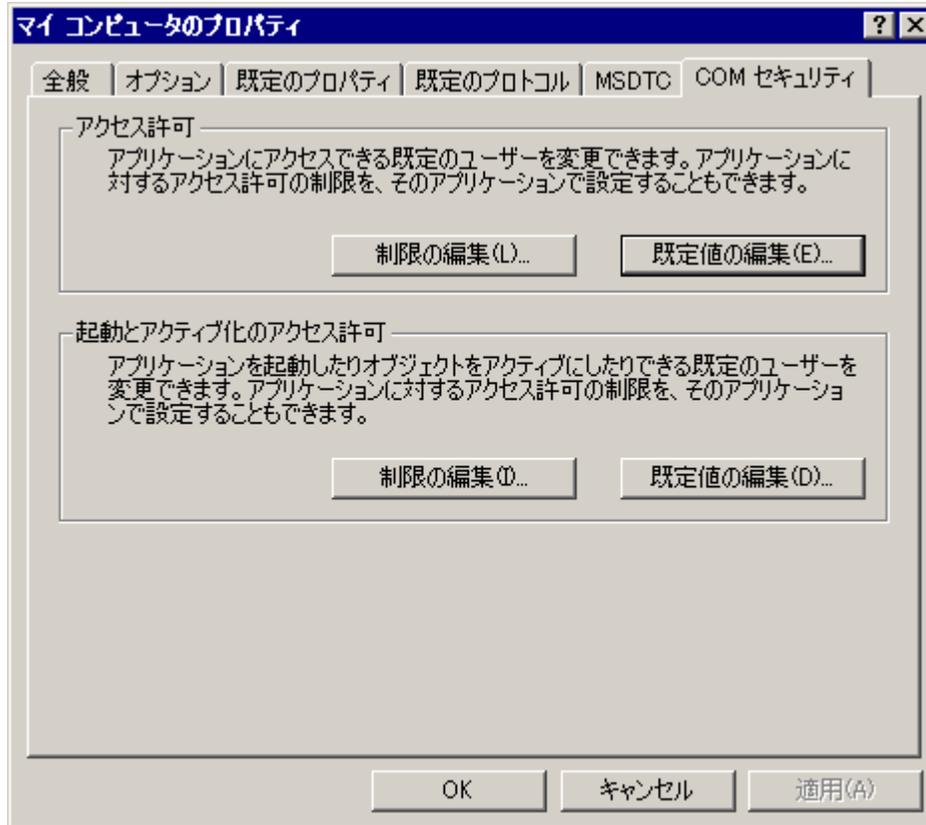
- 2 [マイコンピュータ] の [マイコンピュータのプロパティ] ダイアログボックスを表示し、[既定のプロパティ] タブを選択し、[このコンピュータ上で分散 COM を有効にする] チェックボックスをオンにします。

- 3 「既定の認証レベル」を「既定」に設定し、「既定の偽装レベル」を「識別する」に設定します。



これにより、NaturalX サーバーでクライアントのユーザー ID を取得できるようになります。要求を実行する前に、サーバーは、Natural Security チェックがこのユーザー ID に対して実行されるように、クライアントのユーザー ID を Natural システム変数 *USER に移動します。

- 4 次に、デフォルトのセキュリティ構成を設定します。



[既定の **COM** セキュリティ] タブで、[アクセス許可] グループボックスの [既定値の編集] をクリックします。

[アクセス許可] ダイアログボックスが表示されます。

- 5 [追加] ボタンをクリックし、NaturalX サーバーにアクセスできるユーザーとグループを定義します。



Note: 最低でもアカウント "SYSTEM" にアクセスを許可する必要があります。

ほとんどの場合は、アクセスを許可するすべてのユーザーのグループを定義し、そのグループをここに入力します。この例では、組み込みのグループ " [Everyone] " が選択されています。この設定は、サーバーマシンに定義されているすべてのユーザーにアクセスを許可します。組み込みアカウント "Guest" がユーザーマネージャで有効にされると、この設定はサーバーマシンに定義されていないユーザー (ゲスト) にもアクセスを許可します。



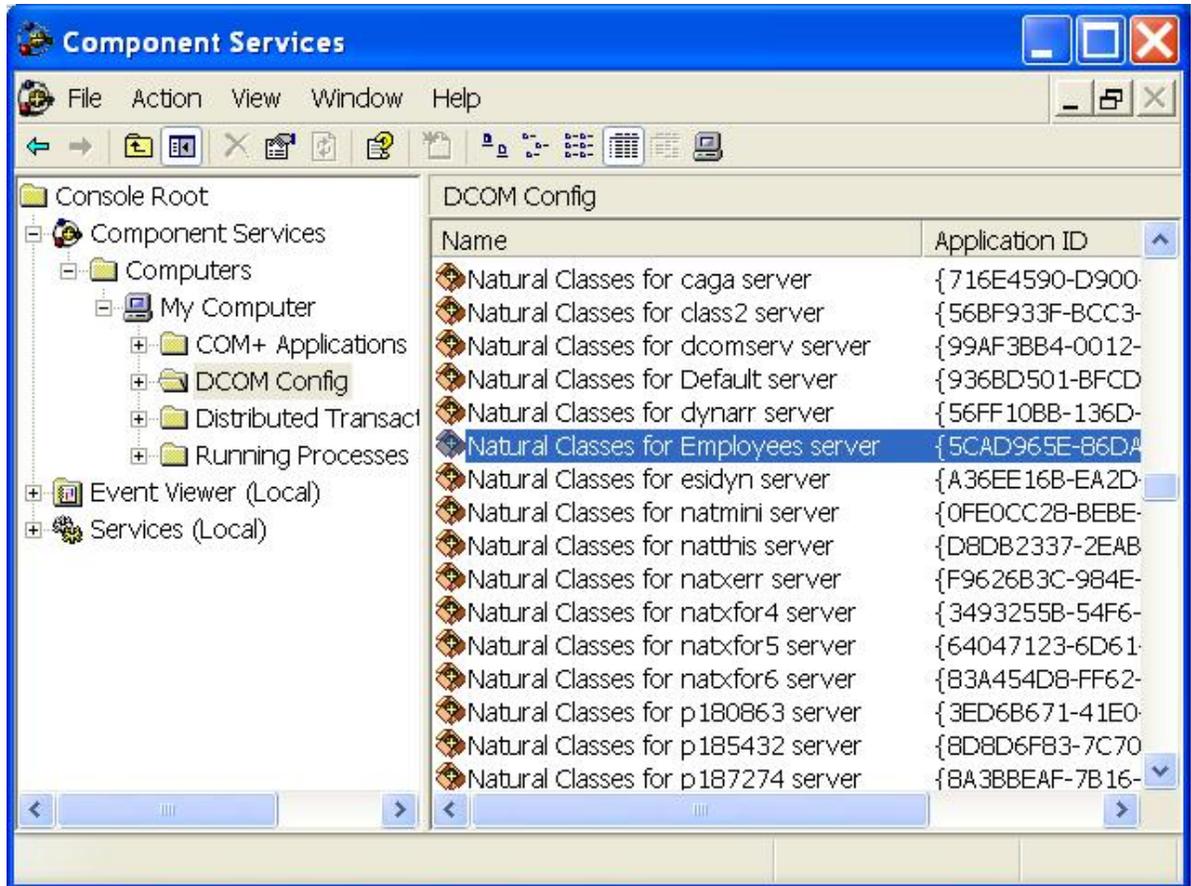
- 6 [既定のCOMセキュリティ] タブで、[起動許可] グループボックスの [既定値の編集] をクリックします。

[起動許可] ダイアログボックスが表示されます。

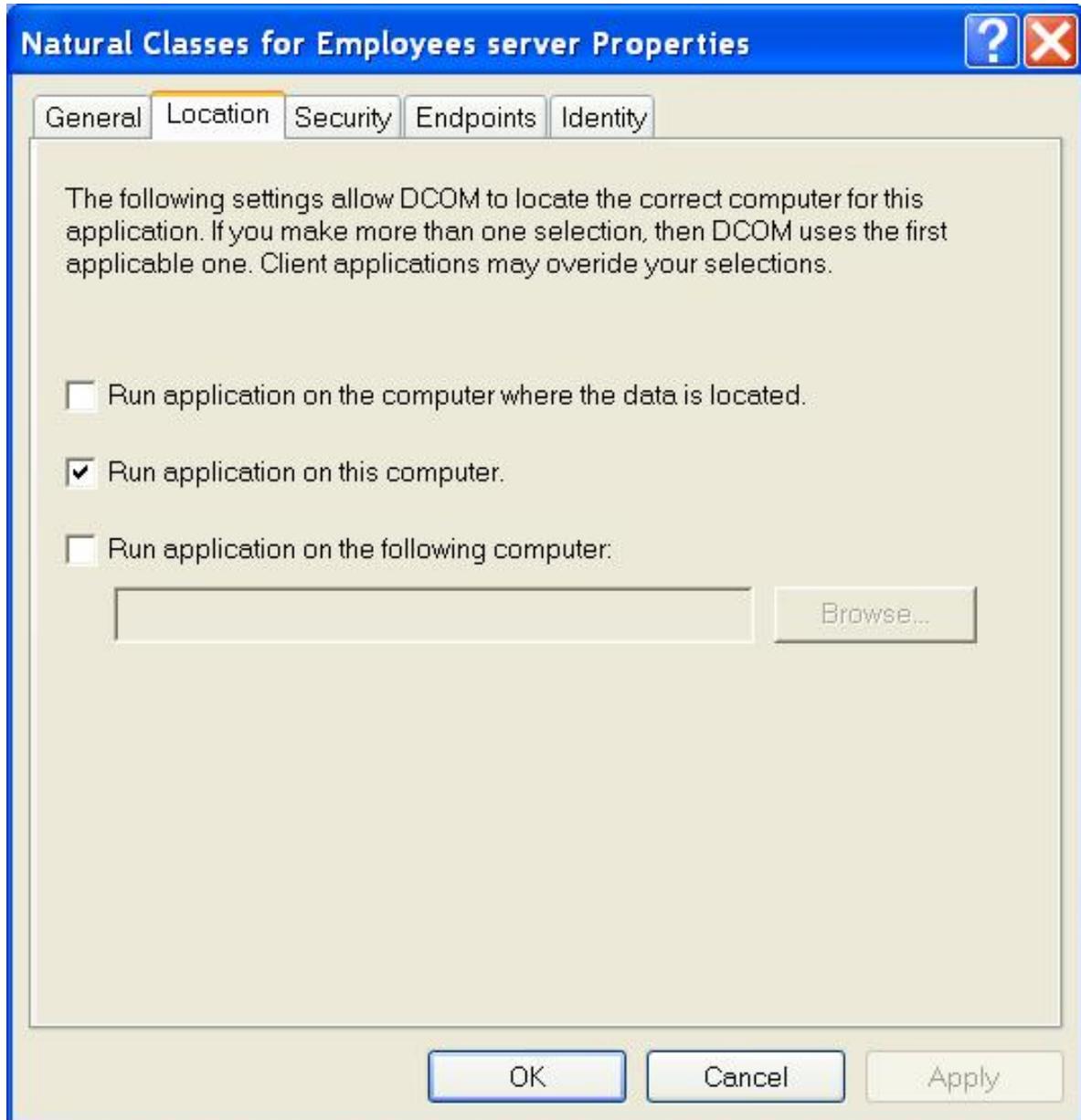


 **Note:** 最低でもアカウント "SYSTEM" と "INTERACTIVE" およびグループ "Administrators" に起動を許可する必要があります。

- 次に、特定の NaturalX サーバーの構成を設定します。[DCOM の構成] ノードを選択し、[DCOM の構成] リストボックスで NaturalX サーバーを見つけます（この例では "Natural classes for Employees server"）。
- サーバーを選択し、[プロパティ] を選択します。

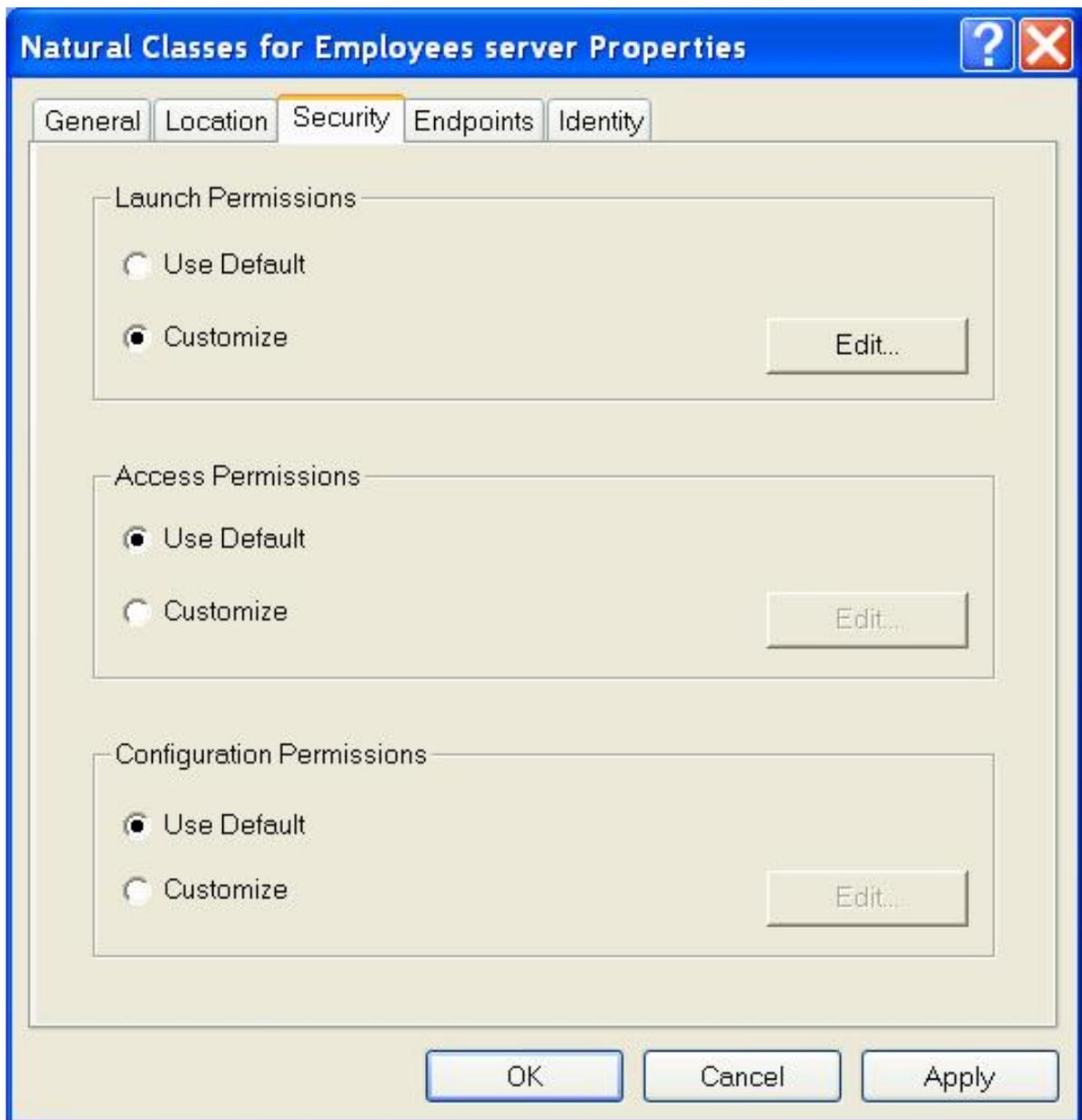


- 9 [場所] タブで、[このコンピュータでアプリケーション実行する] チェックボックスをオンにします。



- 10 [セキュリティ] タブで、[アクセス許可] が [既定値を使用] に設定され、[起動許可] が [カスタマイズ] に設定されることを確認します。

- 11 「起動許可」グループボックスの「編集」をクリックし、アプリケーション固有の起動許可を変更します。



「起動許可」のリストには、最低でもアカウント "SYSTEM" と "INTERACTIVE" およびグループ "Administrators" が含まれます。

- 12 NaturalX サーバーの起動を許可するユーザーとグループを追加します。ほとんどの場合は、起動を許可するすべてのユーザーのグループを定義し、そのグループをここに入力します。この例では、組み込みのグループ「Everyone」が選択されています。この設定は、サーバーマシンに定義されているすべてのユーザーに起動を許可します。組み込みアカウント「Guest」がユーザーマネージャで有効にされると、この設定はサーバーマシンに定義されていないユーザー（ゲスト）にも起動を許可します。



- 13 [ID] タブで、NaturalX サーバーの起動に使用するアカウントを定義します。
- [起動したユーザー] を選択すると、サーバープロセスはクライアントごとに開始されます。サーバープロセスは、クライアントユーザーのアカウントで開始されます。
 - [対話ユーザー] を選択すると、1つのサーバープロセスのみがすべてのクライアントに対して開始されます。

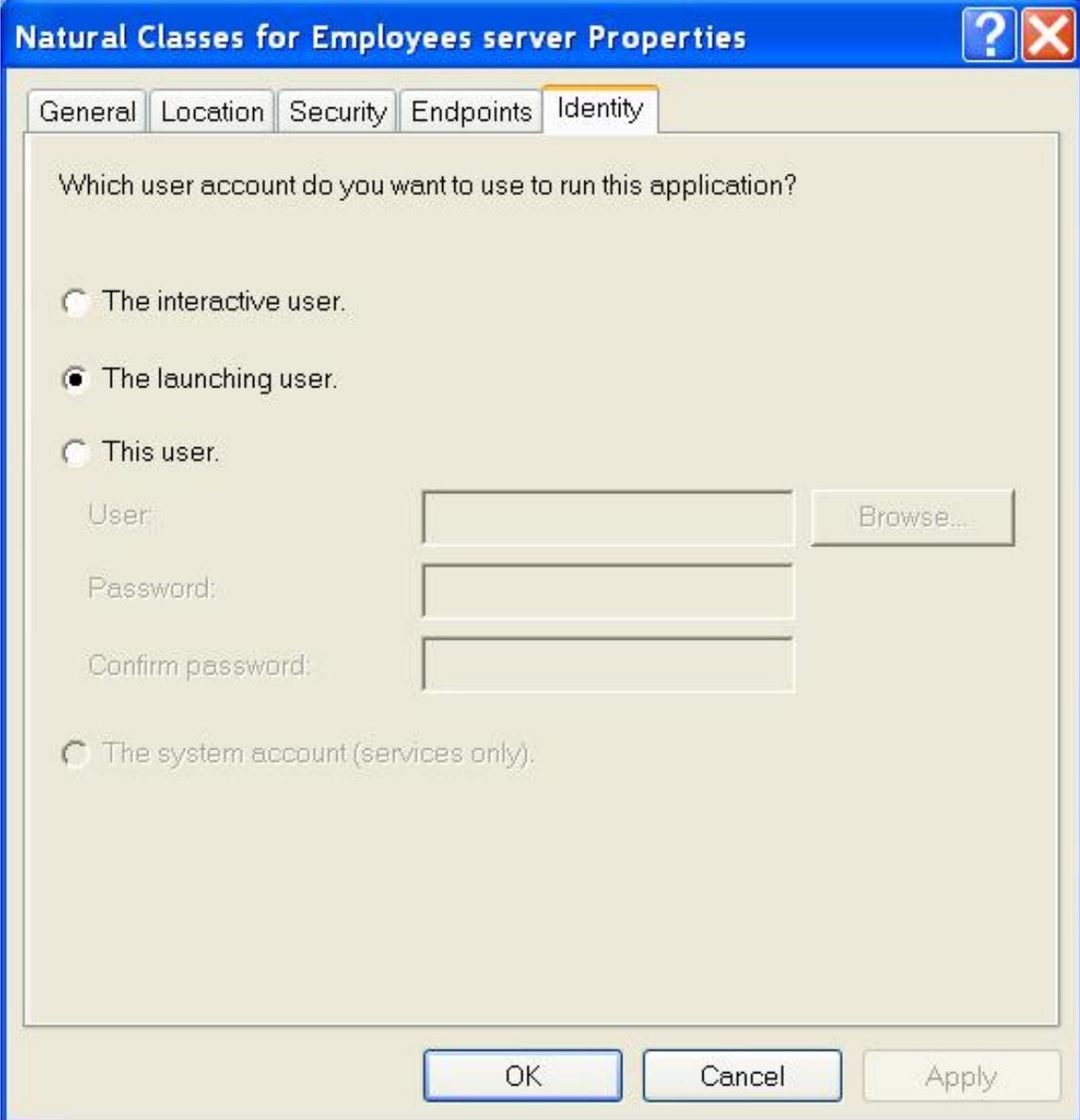
 **Note:** これは、Natural で "ExternalMultiple" として登録済みのクラスにのみ有効です。クラスが "ExternalSingle" として登録されている場合は、このクラスの作成されるオブジェクトごとにサーバープロセスが作成されます。

サーバープロセスは、サーバーマシンに対話形式でログインするユーザーのアカウントで開始されます。サーバーマシンにログインしているユーザーが現在いない場合は、この設定は [起動したユーザー] と同様に機能します。

- [このユーザー] を選択し、特定のユーザーアカウントを指定すると、1つのサーバープロセスのみがすべてのクライアントに対して開始されます。

 **Note:** これは、Natural で "ExternalMultiple" として登録済みのクラスにのみ有効です。クラスが "ExternalSingle" として登録されている場合は、このクラスの作成されるオブジェクトごとにサーバープロセスが作成されます。

サーバープロセスは、指定したユーザーアカウントで開始されます。

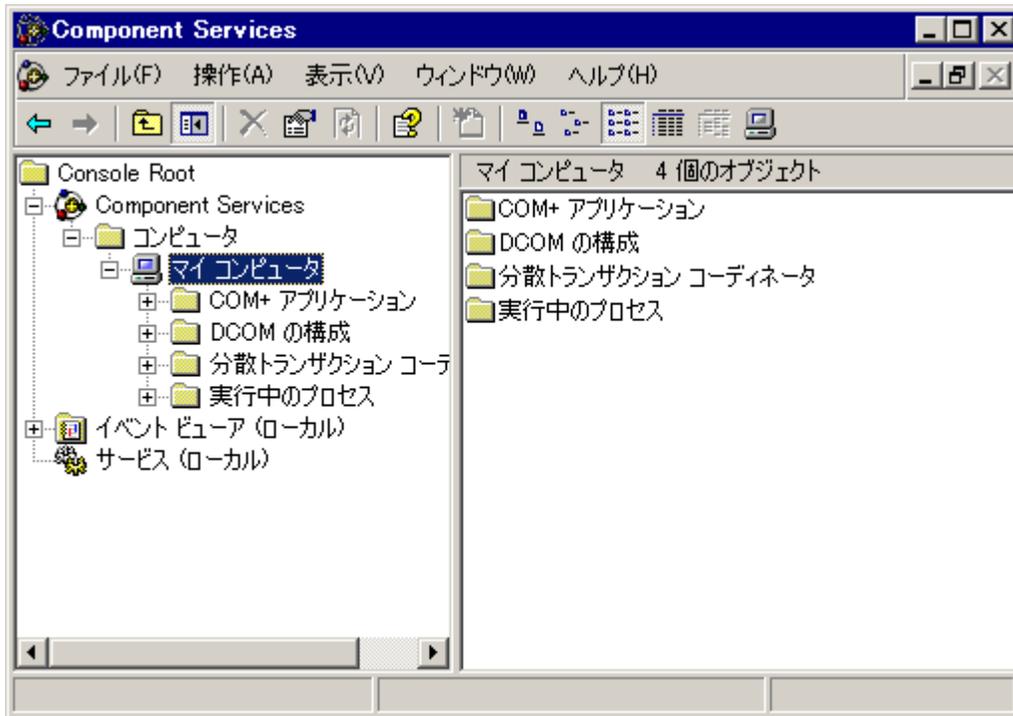


The screenshot shows a Windows dialog box titled "Natural Classes for Employees server Properties". The "Identity" tab is selected. The dialog asks, "Which user account do you want to use to run this application?". There are four radio button options: "The interactive user.", "The launching user." (which is selected), "This user.", and "The system account (services only)". Below the "This user." option, there are three text input fields labeled "User:", "Password:", and "Confirm password:". To the right of the "User:" field is a "Browse..." button. At the bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

NaturalX クライアントの構成

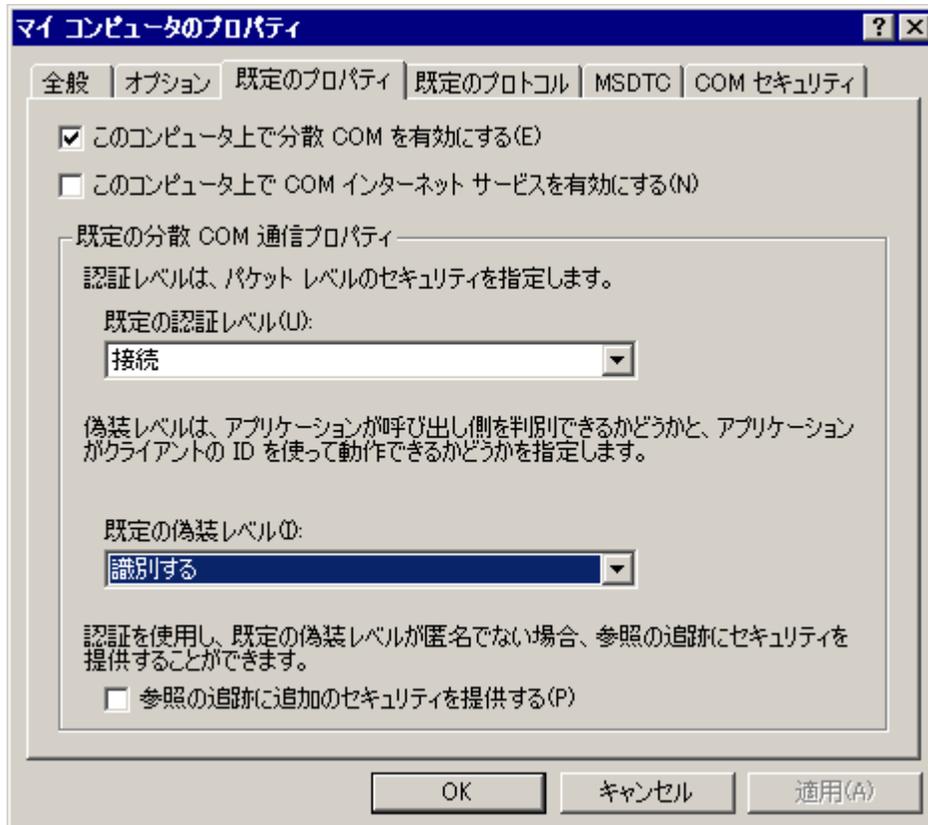
▶手順 33.2. NaturalX クライアントを構成するには

- 1 [コンポーネントサービス] を起動します。



- 2 [マイコンピュータ] の [マイコンピュータのプロパティ] ダイアログボックスを表示し、[既定のプロパティ] タブを選択し、[このコンピュータ上で分散 COM を有効にする] チェックボックスをオンにします。

[既定の認証レベル] を [既定] に設定し、[既定の偽装レベル] を [識別する] に設定します。

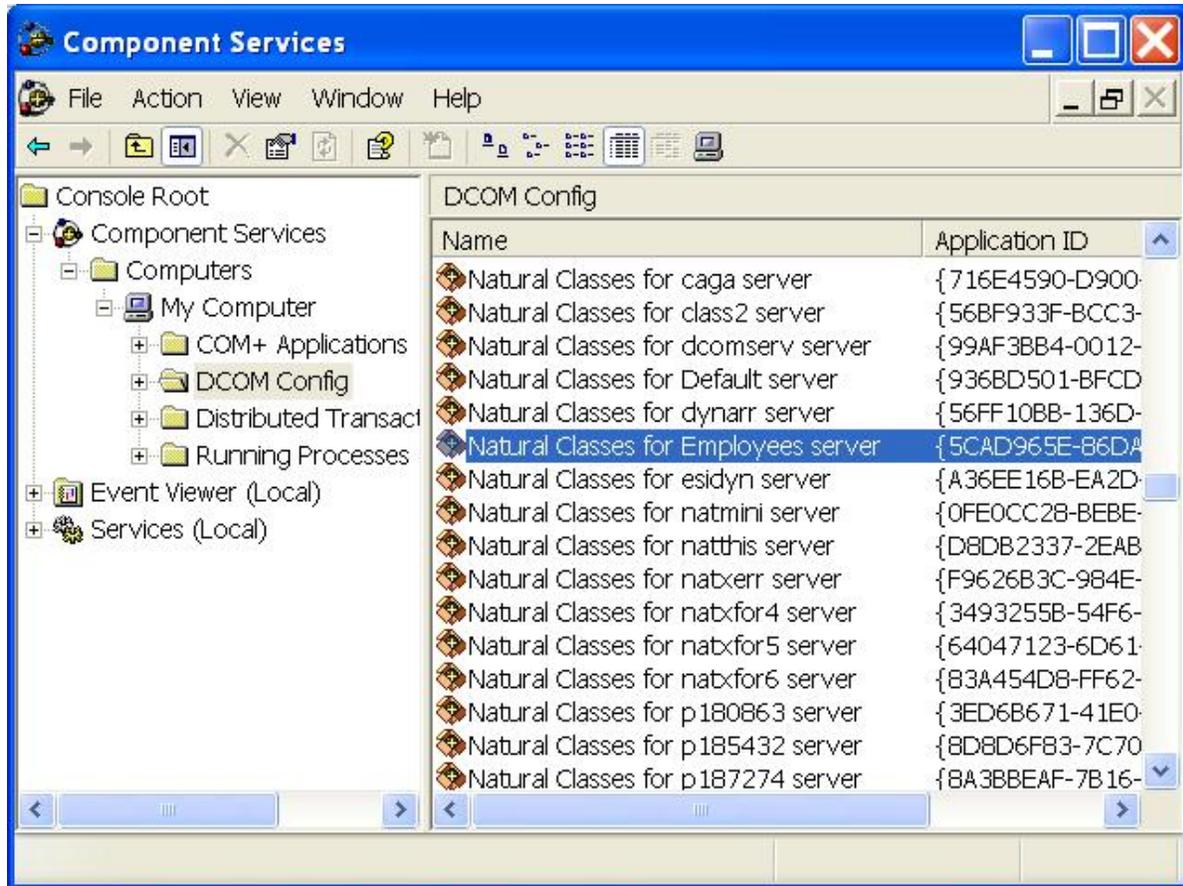


これにより、NaturalX サーバーでクライアントのユーザー ID を取得できるようになります。要求を実行する前に、サーバーは、Natural Security チェックがこのユーザー ID に対して実行されるように、クライアントのユーザー ID を Natural システム変数 *USER に移動します。

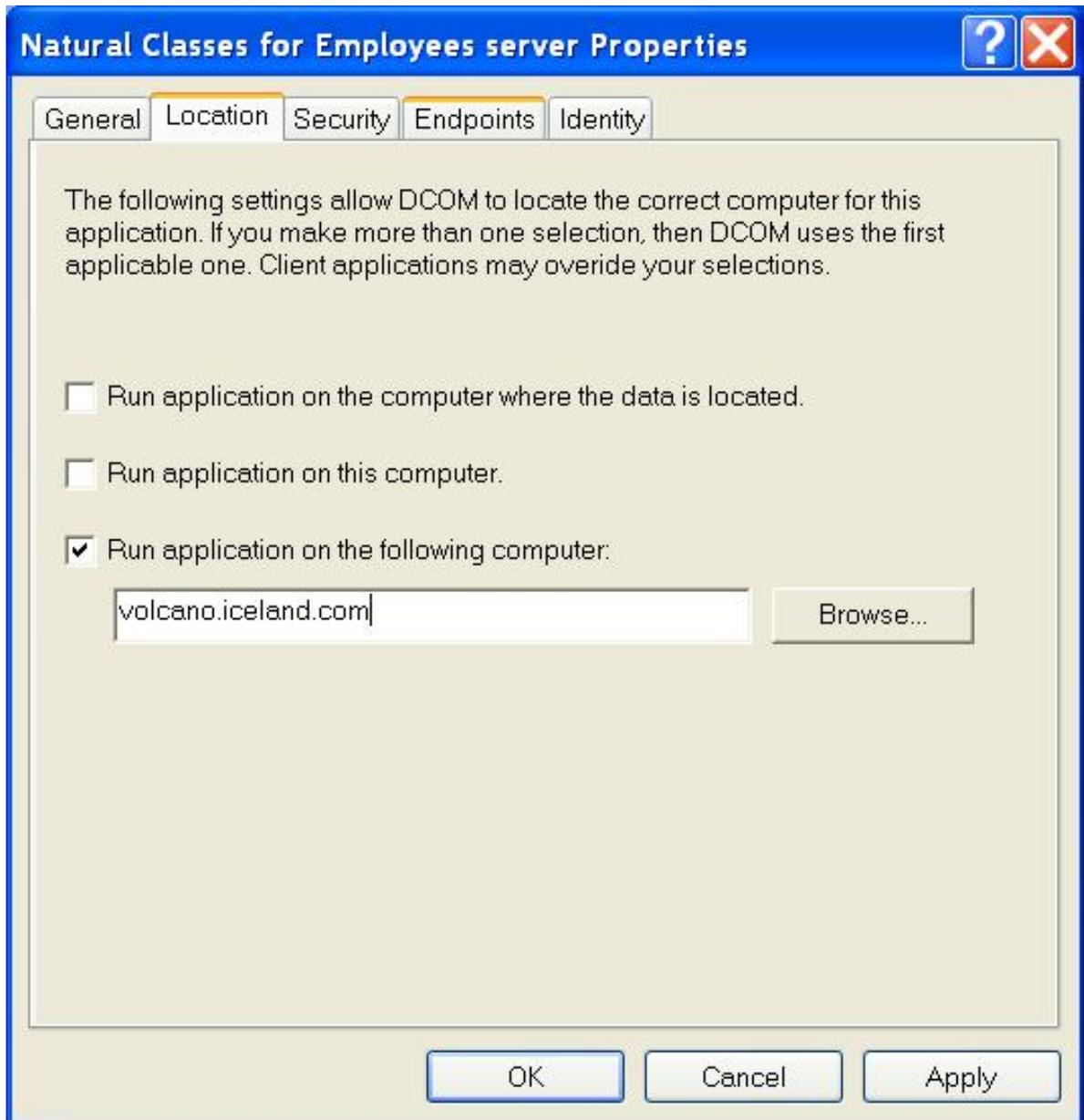
- 次に、特定の NaturalX サーバーにアクセスするための構成を設定します。

[DCOM の構成] ノードを選択し、[DCOM の構成] リストボックスで NaturalX サーバーを見つけます（この例では **Natural classes for Employees server**）。

サーバーを選択し、[プロパティ] を選択します。



- 4 [場所] タブで、[次のコンピュータ上でアプリケーション実行する] チェックボックスをオンにします。NaturalX サーバーがインストールされているリモートマシンの名前を入力します。



34 NaturalX システムレジストリエントリ

- サーバーのためのレジストリエントリ 246
- クライアントのためのレジストリエントリ 247

このchapterでは、次のトピックについて説明します。

サーバーのためのレジストリエントリ

以下の表に、新しいクラスが登録されるときにサーバーのシステムレジストリに追加されるキーと値の概要を示します。

「親キー」列は、その下に新しいキーが作成されるキーを示します。追加されるキーは「サブキー」列にリストされ、「値の名前」列と「値」列は新しいエントリの値を示します。

 **Note:** <class_name> はクラスの名前、<class_ID> はクラスのクラス GUID です。これらは、クラスモジュールの DEFINE CLASS ステートメントに定義されています。

以下では次のトピックについて説明します。

- [DCOM に必要なキー](#)
- [Natural に必要なキー](#)

DCOM に必要なキー

| 親キー (HKEY_CLASSES_ROOT...) | サブキー | 値名 | 値 |
|---|----------------------------------|-------|---|
| ... | <ProgID> (<class_name>.1) | - | <class_name> "1.0" |
| ... \<ProgID> | CLSID | - | <class_GUID> |
| ... | <VersIdProgID> (<class_name>) | - | <class name> "1.0" |
| ... \<VersIdProgID> | CLSID | - | <クラス GUID> |
| ... \AppId | <APPID> | - | "Natural classes for" <server_ID> "server" |
| ... \CLSID | <CLSID> | - | <class_name> "1.0" |
| ... \CLSID | <CLSID> | AppId | <サーバーの GUID> |
| ... \CLSID \<CLSID> | LocalServer32 | - | <Natural パス> |
| ... \CLSID \<CLSID> | ProgID | - | <ProgID> |
| ... \CLSID \<CLSID> | TypeLib | - | <タイプライブラリの GUID> |
| ... \CLSID \<CLSID> | Version | - | "1.0" |
| ... \CLSID \<CLSID> | VersionIndependentProgID | - | <VersIDProgID> |
| ... \CLSID \<CLSID> (バージョン 4.1.2 以降のすべてのリリース) | Programmable | - | - |
| ... \TypeLib | <TLID> | - | - |
| ... \TypeLib \<TLID> | 1.0 <version> | - | "Natural" <class_name> "class" |

| 親キー (HKEY_CLASSES_ROOT...) | サブキー | 値名 | 値 |
|----------------------------|------------------|----|------------------------------|
| ... \TypeLib\<TLID>\1.0 | 0 (langcode) | - | - |
| ... \TypeLib\<TLID>\1.0\0 | win32 (platform) | | <タイプライブラリパス> |
| すべてのインターフェイス: | | | |
| ... \Interface | <IID> | - | <インターフェイス名> |
| ... \Interface\<IID> | ProxyStubClsid32 | - | <IDispatch のプロキシ dll の GUID> |
| ... \Interface\<IID> | BaseInterface | - | <IDispatch の GUID> |

Natural に必要なキー

| 親キー (HKEY_LOCAL_MACHINE\SOFTWARE\SoftwareAG\Natural\Servers...) | サブキー | 値名 | 値 |
|---|-------------------------|------------|---|
| ... | <server_ID> | AppId | <サーバーの GUID> |
| ... \<server_ID>\ | CLSID | - | - |
| ... \<server_ID>\CLSID | <CLSID> (<class_ID>) | NatMember | <Natural クラスモジュール名> |
| ... \<server_ID>\CLSID | <CLSID> | NatLibrary | <クラスモジュールの Natural ライブラリ> |
| ... \<server_ID>\CLSID | <CLSID> | NatContext | "ExternalSingle"、"InternalMultiple"、または "ExternalMultiple" (「アクティベーションポリシー」を参照) |

クライアントのためのレジストリエントリ

以下の表に、クライアント登録ファイルが実行されるときにクライアントシステムレジストリに追加されるキーを示します。

| 親キー (HKEY_CLASSES_ROOT...) | サブキー | 値名 | 値 |
|----------------------------|----------------------------------|----|--------------------|
| ... | <ProgID> (<class_name>.1) | - | <class_name> "1.0" |
| ... \<ProgID> | CLSID | - | <クラス GUID> |
| ... | <VersIdProgID> (<class_name>) | - | <class_name> "1.0" |
| ... \<VersIdProgID> | CLSID | - | <クラス GUID> |

| 親キー (HKEY_CLASSES_ROOT...) | サブキー | 値名 | 値 |
|---|------------------------------|------------------|--|
| ... \<VersIdProgID> | CurVer | - | <ProgID> |
| ... \AppId | <APPID> | - | "Natural classes for server" <server_ID> "server" |
| ... \AppId | <APPID> | RemoteServerName | ユーザーが入力する必要があります。 |
| ... \CLSID | <CLSID> | - | <class_name> "1.0" |
| ... \CLSID | <CLSID> | AppId | <サーバーの GUID> |
| ... \CLSID \<CLSID> | ProgID | - | <ProgID> |
| ... \CLSID \<CLSID> | Version | - | "1.0" |
| ... \CLSID \<CLSID> | VersionIndependent ProgID | - | <VersProgID> |
| ... \CLSID \<CLSID> (バージョン 4.1.2 以降のすべてのリリース) | Programmable | - | - |
| すべてのインターフェイス: | | | |
| ... \Interface | <IID> | - | <インターフェイス名> |
| ... \Interface \<IID> | ProxyStubClsid32 | - | <IDispatch のプロキシ dll の GUID> |
| ... \Interface \<IID> | BaseInterface | - | <IDispatch の GUID> |

35 NaturalX サーバー環境でのステートメントおよび コマンドの使用

- Natural ステートメント 250
- Natural システムコマンド 251

一部の Natural ステートメントおよび Natural システムコマンドの動作はサーバー環境では変化します。

このchapterでは、次のトピックについて説明します。

Natural ステートメント

このsectionでは、次のステートメントについて説明します。

- DISPLAY、INPUT、PRINT、REINPUT、および WRITE ステートメント
- WRITE WORK FILE および READ WORK FILE ステートメント
- STOP および TERMINATE ステートメント

DISPLAY、INPUT、PRINT、REINPUT、および WRITE ステートメント

- 画面への出力（レポート0への出力）は、サーバー環境では適切ではなく、場合によっては実行できません。したがって、サーバー環境での相互 I/O の場合は、クライアントにエラー NAT0723 が返されます。MAINPR パラメータを使用して I/O をリダイレクトすることは、当然可能であり、完全にサポートされます。
- 出力がメソッドによってレポートに書き込まれる場合は、レポートはメソッドの開始時に開かれ、終了時に閉じられます。クライアント間の衝突を避けるために、メソッド呼び出し間ではレポートは開いた状態を保持されません。

WRITE WORK FILE および READ WORK FILE ステートメント

メソッド内でワークファイルにアクセスする場合は、ファイルはメソッドの開始時に開かれ、終了時に閉じられます。クライアント間の衝突を避けるために、メソッド呼び出し間ではファイルは開いた状態を保持されません。

STOP および TERMINATE ステートメント

- TERMINATE ステートメントの動作は、STOP ステートメントの動作と一致しています。戻り値の処理はサポートされません。
- STOP および TERMINATE ステートメントは、メソッド実行中に ESCAPE ROUTINE ステートメントと同じように動作します。メソッド実行はどのような戻り値も生成することなく即座に終了します。

Natural システムコマンド

サーバー環境では、以下の Natural コマンドのみを使用できます。

- CATALOG
- CLEAR
- EXECUTE
- LOGOFF
- LOGON
- READ
- RETURN
- RUN
- SAVE
- SETUP
- STOW

上記のコマンドのうち CATALOG、CLEAR、READ、RUN、SAVE、および STOW は、サーバーが開発 Natural (*natural.exe*) で実行されている場合にのみ許可されます。

許可されないすべてのコマンドは、エラー NAT0082 で拒否されます。

索引

シンボル

\$ ファイル名, 28
*.ico
出力ウィンドウ, 94

A

ASCII
ワークファイルフォーマット, 30

C

CASE-TRANSLATION
NATCONV.INI セクション, 115
CHARACTER-CLASSIFICATION
NATCONV.INI セクション, 115
CHARACTERSET-DEFINITION
NATCONV.INI セクション, 115
CLEAR
NATBPMON コマンド, 73
CMOBJIN
バッチモード, 83
CMPRINT
バッチモード, 83
CMPRTnn
バッチモード, 84
CMSYNIN
バッチモード, 83
CMWRKnn
バッチモード, 84
COM クラス
NaturalX, 194
CONFIG
NWOSVCD コマンド, 129
CONSOLE
NWOSVCD コマンド, 129
CORPSES
NATBPMON コマンド, 73
CPOBJIN
バッチモード, 84
CPPRINT
バッチモード, 84
CPSYNIN
バッチモード, 84
CREATE
NATBPSVC コマンド, 41

NATRTSVC コマンド, 110
CSV
ワークファイルフォーマット, 31

D

DCOM
NaturalX クラス, 191
DCOM コンフィグレーション
NaturalX, 229
DDM
システムファイル FDDM への移行, 19
DELETE
NATBPMON コマンド, 73
NATBPSVC コマンド, 42
NATRTSVC コマンド, 110
DIR
NATBPMON コマンド, 73
DUMP
NATBPMON コマンド, 73

E

Entire Connection
ワークファイルフォーマット, 30
Entire System Server インターフェイス
設定, 123
ERR
サブディレクトリ, 16
EXIT
NATBPMON コマンド, 73

F

FDDM
システムファイル, 17
FDDM モード
アクティブ化, 18
FILEDIR.SAG, 21
FIN
NATBPMON コマンド, 73
FNAT
システムファイル, 15
FUSER
システムファイル, 15

G

GP

サブディレクトリ, 16

H

HELP

NATBPMON コマンド, 73

I

IDENTIFIER-VALIDATION

NATCONV.INI セクション, 115

INSTALL

NATBPSVC コマンド, 41
NATRTSVC コマンド, 109
NWOSVCD コマンド, 129

L

LIBDIR.SAG, 16

N

NATBP

デフォルトバッファプール, 40

NATBPMON

バッファプールモニタ, 71

NATBPSRV

バッファプールを作成するユーティリティ, 44

NATBPSVC

バッファプールサービスコマンドの実行, 41

NATCONV.INI

異なる文字セットのサポート, 113

NATLOG

バッチモード, 84

NATRTSVC

Natural ランタイム起動サービスコマンドの実行, 109

Natural ランタイム, 97

Natural ランタイム起動サービス, 108

NaturalX

アプリケーション, 191

NCD

ワークファイル拡張子, 33

NWOSVCD

Web I/O インターフェイスサービスコマンドの実行, 129

P

PARAM

NATBPMON コマンド, 73

PF キーボタン (参照 Web I/O インターフェイス)

Portable

ワークファイルフォーマット, 31

Q

QUIT

NATBPMON コマンド, 73

R

REGISTER

システムコマンド, 207

REMOVE

NATBPSVC コマンド, 42

NATRTSVC コマンド, 109

NWOSVCD コマンド, 129

RES

サブディレクトリ, 17

S

SAG

ワークファイルタイプ, 30

SET

NATBPSVC コマンド, 41
NATRTSVC コマンド, 110

SHOW

NATBPSVC コマンド, 41
NATRTSVC コマンド, 110

SHUTDOWN

NATBPMON コマンド, 73

SRC

サブディレクトリ, 16

START

NATBPSVC コマンド, 41
NATRTSVC コマンド, 109

STATUS

NATBPMON コマンド, 73
NATBPSVC コマンド, 42
NATRTSVC コマンド, 110

STOP

NATBPSVC コマンド, 41
NATRTSVC コマンド, 109

SWAP

NATBPMON コマンド, 74

T

Transfer

ワークファイルタイプ, 30

U

Unformatted

ワークファイルフォーマット, 31

UNREGISTER

システムコマンド, 208

W

Web I/O インターフェイス (参照 イベントログ) (参照 クライアントの設定) (参照 コンフィギュレーションファイル) (参照 サービスの設定) (参照 サービスコマンド) (参照 テーマ) (参照 バッチファイル) (参照 ログオンページ) (参照 独自の設定で開始) (参照 色属性とフィールド属性)

制限, 189

違い, 187

WHO

NATBPMON コマンド, 74

WRITE

NATBPMON コマンド, 74

Z

ZERO

NATBPMON コマンド, 74

あ

- アイコン
 - 出力ウィンドウ, 94
- アクティベーションポリシー
 - NaturalX, 197
- アプリケーション
 - NaturalX, 191
 - ランタイムワークステーションへの移植, 99

い

- 移行
 - 古い FILEDIR.SAG, 24
- 移植
 - アプリケーションをランタイムワークステーションに, 99
- 移植可能なシステムファイル, 22
- イタリック体属性 (参照 Web I/O インターフェイス)
- 一般情報
 - バッファプールモニタのノード, 56
- イベントログ (参照 Web I/O インターフェイス)
- 色属性 (参照 Web I/O インターフェイス)

え

- エラー
 - Natural スタートアップ, 119

お

- オブジェクト
 - バッファプールへの保存, 37
- オプション
 - 出力ウィンドウ, 93
- 折れ線グラフ
 - バッファプールモニタのノード, 62

か

- 開始
 - GUI バージョンのバッファプールモニタ, 48
 - NATBPMON ユーティリティ, 72
 - Natural でのダイナミックパラメータ値の使用, 9
 - バッチモードの Natural, 82
 - バッファプールサーバー, 54
 - 別のバッファプールを使用した Natural の開始, 42
- 下線属性 (参照 Web I/O インターフェイス)

く

- クラス
 - NaturalX, 194
 - 登録, 205
- グラフィックアナライザ
 - バッファプールモニタのノード, 61

こ

- 高輝度属性 (参照 Web I/O インターフェイス)
- 構成
 - NaturalX, 219
- 高度分析
 - バッファプールモニタのノード, 70

コマンド

- NATBPMON, 73
- Natural ランタイム起動サービス, 109
- Web I/O インターフェイスサービス, 129
 - バッファプールサービス, 41
- コントロールブロック
 - バッファプール, 36
- コンフィグレーションファイル (参照 Web I/O インターフェイス)
- コードページ
 - バッチモード, 84

さ

- 削除オブジェクト
 - バッファプールモニタのノード, 60
- 作成
 - バッファプール, 44
- サーバー
 - NaturalX, 193
- サーバー ID
 - NaturalX, 195

し

- システムファイル
 - 全般的な情報, 13
- シャットダウン
 - バッファプールサーバー, 53
- 終了
 - GUI バージョンのバッファプールモニタ, 48
 - バッチモードの Natural, 83
- 出力ウィンドウ
 - ローカル環境, 91
- 出力チャネル
 - バッチモード, 83
- シンプルレポート
 - バッファプールモニタのノード, 67

す

- スタートアップエラー, 119
- ステータスバー
 - バッファプールモニタ, 52

せ

- 制限
 - Web I/O インターフェイス, 189
- 制限事項
 - バッチモード, 89
 - バッファプール, 40
- セキュリティ
 - NaturalX, 223
- 接続
 - バッファプール, 52
- 切断
 - バッファプール, 52
- 前景色 (参照 Web I/O インターフェイス)

た

- タイプ情報

NaturalX, 213
ダイアログ
バッチモードでの使用, 89

ち

違い
Web I/O インターフェイス, 187
チャンネル
バッチモード, 83

つ

ツールバー
バッファプールモニタ, 50

て

定義
ワークファイル, 26
点滅属性 (参照 Web I/O インターフェイス)
テーマ (参照 Web I/O インターフェイス)
ディレクトリエントリ
バッファプール, 36
バッファプールモニタのノード, 58
出口コード, 119
デリミタ
環境変数, 27

と

統計
バッファプールモニタのノード, 56
登録
NaturalX, 205
同期
複数のバッファプール, 37
ドル記号
ファイル名, 28

に

入力チャンネル
バッチモード, 83

は

背景色 (参照 Web I/O インターフェイス)
反転属性 (参照 Web I/O インターフェイス)
バイナリ
ワークファイルフォーマット, 30
バッチファイル (参照 Web I/O インターフェイス)
バッチモード, 81
検出, 88
バッファプール
作成, 44
設定, 40
接続, 52
切断, 52
全般的な情報, 35
同期, 37
バッファプールモニタのノード, 51
モニタ, 44

バッファプールコンテンツ
バッファプールモニタのノード, 58
バッファプールサーバー
開始, 54
シャットダウン, 53
バッファプールサービス, 40
バッファプールモニタ
GUI バージョン, 47
NATBPMON, 71
プロパティ, 55
パラメータ
バッファプールモニタのノード, 58
パラメータ階層, 8
パラメータ値のスタティックな割り当て, 9
パラメータ値のダイナミックな割り当て, 9
パラメータ値のランタイム割り当て, 10

ひ

筆記体属性 (参照 Web I/O インターフェイス)

ふ

ファイアウォール
Natural を実行するための設定, 3
フィルタ
バッファプールモニタのディレクトリエントリ用の定義, 59
フィールド属性 (参照 Web I/O インターフェイス)
複数のバッファプール, 37
プロパティ
バッファプールモニタ, 55
プロファイルパラメータ
使用方法, 7

ほ

棒グラフ
バッファプールモニタのノード, 65

ま

マップ
バッチモードでの使用, 89

め

メニューバー
バッファプールモニタ, 50
メモリ使用グラフ
バッファプールモニタのノード, 66

も

文字セット
NATCONV.INI での異なる定義, 113
モニタ
バッファプール, 44

ゆ

ユーザー
バッファプールモニタのノード, 61

よ

要素

バッファプールモニタウィンドウ, 49

ら

ランタイム, 97

ランタイム起動サービス, 108

り

リソース処理

バッファプール, 37

れ

レジストリエントリ

NaturalX, 245

レポート

バッファプールモニタのノード, 66

ろ

ログ

バッファプールモニタのノード, 68

ログオンページ (参照 Web I/O インターフェイス)

ローカル環境

出力ウィンドウ, 91

わ

ワークファイル

全般的な情報, 25

定義, 26

フォーマット, 29

