

Natural for Mainframes

Web テクノロジ

バージョン 4.2.5

October 2009

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

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

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

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

目次

1 Web テクノロジ	1
2 Natural Web インターフェイス	3
3 Natural Web インターフェイスについて	5
Natural Web インターフェイスとは	6
アーキテクチャ	6
Natural Web インターフェイスモジュール	8
機能	8
機能	9
セキュリティ	11
4 Natural Web インターフェイスのコンフィグレーション	13
5 Natural Web インターフェイスの構成	15
サポートされる HTTP サーバー	16
RPC および RPC サーバーの構成	16
Web インターフェイスの構成	18
HTTP サーバーの構成	19
Natural Security との通信	19
6 Web インターフェイスのトラブルシューティング	21
7 Natural Web インターフェイスの要点	25
8 Natural Web インターフェイスでの作業	27
環境の設定	28
Natural でのサブプログラムの構築	29
9 Natural Web サーバーエクステンション	45
10 Natural Web サーバーエクステンションの概要 - SYSWEB を使用する場合	47
一般的な情報	48
インストール - RPC	48
変換	49
変数	49
エラーログとメッセージ	49
プログラムの呼び出し	50
11 Natural Web サーバーエクステンション - 初期化ファイル	51
一般的な情報	52
RPC パラメータ	52
DCOM パラメータ	53
Natural Web サーバーエクステンション設定	53
HTTP サーバー変数	56
追加の変数	56
エラーテンプレート	56
12 Natural Web サーバーエクステンション - エラーメッセージ	61
13 プログラミングのヒント	67
小文字での編集	68
引用符と アポストロフィ	68
値で定義された変数	69
リソースへのアクセス	69

定数値	69
新しいページの作成	70
DCOM/RPC	70
14 Web インターフェイスの管理	71
戻りページトランスポートバッファサイズの設定	72
ユーザー定義エラーページの作成	73
ユーザー定義エラーページ XML スタイルの作成	73
英数字から HTML への変換	74
英数字から URL への変換	74
15 デモアプリケーション - JavaScript なし	75
ビジネス要件	76
設計に関する決定	77
ライブラリ、モジュール、および命名規則	77
デモアプリケーションの開始	78
Natural Web インターフェイスオンラインマニュアルの開始	78
要件	78
16 デモアプリケーション - JavaScript 使用	79
ビジネス要件	80
設計に関する決定	81
デモアプリケーションの開始	81
要件	81
17 Natural Web インターフェイスのエラーメッセージ	83
エラーメッセージ	84
18 Natural Web オンラインドキュメント SYSWEB	85
全般的な情報	86
基本モジュール	86
出力の後処理	88
HTML エクステンション	88
ユーティリティ	90
デモアプリケーション	90
19 出力エリアのクリア	93
20 ドキュメントのコンテンツタイプの設定	95
21 出力エリアのサイズのカウンタ	97
22 エラーページの生成	99
23 ドキュメントへの書き込みおよび有効な HTML への変換	101
24 ドキュメントへの HTTP 設定の書き込み	103
25 内部値に関する情報	105
26 ドキュメントの終了および初期化	107
27 すべての環境変数のリスト	109
28 ドキュメントの場所の設定	111
29 環境変数の読み取り	113
30 環境変数グループの読み取り	115
31 環境テキストエリア変数の読み取り	117
32 ドキュメントへのテキストの書き込み	119
33 出力エリアへの改行の書き込み	121



34 テキストから HTML	123
35 テキストから XML	125
36 テキストから URL	127
37 返されるドキュメント内での置換	129
38 出力ページの読み取り	131
39 アンカー	133
40 ボタン	135
41 チェックボックスグループ	137
42 コメント行	139
43 レベル n のヘッダー	141
44 イメージ	143
45 入力	145
46 改行	147
47 フォーム	149
48 HTML ドキュメント	151
49 List	153
50 パラグラフ	157
51 ラジオボタングループ	159
52 横罫線	161
53 スクロールリスト	163
54 テーブル	165
55 汎用タグ	169
56 テキストエリア	171
57 テキストから URL デコード形式	173
58 時刻/日付文字列	175
59 すべての Natural ライブラリのリスト	177
60 オンライン Natural Web インターフェイス サブプログラムの実行	179
61 Natural Web インターフェイスで使用する Natural サブプログラムの生成	181
62 Natural ライブラリのディレクトリのリスト	183
63 呼び出された Natural サブプログラムに HTTP サーバーから渡されたすべてのパラメータのリスト	185
64 Natural ソースオブジェクトとして保存された HTML ページを返す	187
65 Natural Web インターフェイスの現在の設定をリストします。	189
66 Natural オブジェクトのソースのリスト	191
67 オンラインドキュメント	193
68 XML ツールキット	195
69 はじめに	197
XML ツールキットの特徴	198
XML ツールキットの説明	198
今後の方針	202
考慮事項と制限事項	202
70 XML ツールキットの使用	203
前提条件	204
ワークファイルの処理	204
出力ファイル	207

アプリケーションの起動	207
PF キーの割り当て	208
71 特定の生成オプションの設定	209
生成オプション設定画面の起動	210
最初の画面	210
2 番目の画面	212
オプションの永続的な保存	214
72 Natural データソースの使用	215
Natural データエリアの選択	216
ルートグループの選択	217
DTD 定義でのファイルの生成	218
XML ドキュメントのパースの生成	219
生成レポートの表示	220
73 外部データソースの使用	223
文書型定義からの生成	224
ルート要素の選択	225
Natural データエリアの生成	226
XML ドキュメントへのシリアル化	227
コピーコードの生成	229
生成結果の表示	230
74 Natural のシンプルな XML パーサー	233
パーサーの説明と例	234
パーサーの制限事項	241
75 例	243
コピーコードのシリアル化	244
生成された Natural データエリア	246
Natural DTD パーサー	248
生成されたタイプ定義	249
パーサー CALLBACK コピーコード	250
76 解析エラーメッセージ	257
索引	259

1 Web テクノロジ

このドキュメントでは、Natural Web テクノロジおよびその機能の概要について説明します。

次のトピックについて説明します。

 Natural Web インターフェイス	Natural Web インターフェイスは、Web サーバー (HTTP サーバー) とユーザーの Natural 環境をリンクするものです。
 XML ツールキット	XML ツールキットを使用すると、開発者は Natural 内で XML ドキュメントを処理できます。

2 Natural Web インターフェイス

Natural Web インターフェイスは、Web サーバー（正確には HTTP サーバー）とユーザーの Natural 環境とをリンクするものです。

Natural Web インターフェイスのドキュメントは、以下のドキュメントで構成されます。

- [Natural Web インターフェイスについて](#)
- [Natural Web インターフェイスのコンフィグレーション](#)
- [Natural Web インターフェイスの要点](#)
- [Natural Web オンラインドキュメント SYSWEB](#)

3 Natural Web インターフェイスについて

- Natural Web インターフェイスとは 6
- アーキテクチャ 6
- Natural Web インターフェイスモジュール 8
- 機能 8
- 機能 9
- セキュリティ 11

情報やサービスをインターネット経由で提供する必要のある企業が、ますます増加しています。以前、Web ページに日々アクセスする訪問者に対しては、スタティックな HTML ページを提供することで十分でした。今日、Web ではより洗練された HTML ページが提供されるようになり、インターネットを経由してビジネスロジックに完全にアクセスできるようにする必要性が急激に高まっています。ビジネスに不可欠な情報が含まれているデータベース管理システムは、多くの場合、メインフレームのような高性能サーバーを基盤としています。

このセクションでは、次のトピックについて説明します。

Natural Web インターフェイスとは

Natural Web インターフェイスは、Web サーバー（正確には HTTP サーバー）とユーザーの Natural 環境とをリンクするものです。これは、別のサーバーマシン（メインフレームなど）または HTTP サーバーと同じマシン（Apache や Microsoft IIS など）でも使用可能です。

Web ページの内容は、容易に Natural プログラムによって動的に作成できます。これは、実際の対話式アプリケーションを Web に実装する基盤となります。

対話式アプリケーションでは、ユーザーは情報を入力でき、アプリケーションはその入力に応じた出力を発行することによって応答します。Web ベースのアプリケーションの例として、注文入力システム、旅行予約サービス、集配システムなどを挙げることができます。これにより、Natural アプリケーションの適用範囲が大きく拡張されます。今日、企業内ユーザーだけでなく、世界中の潜在的な顧客が、同じアプリケーションを使用できるようになりました。

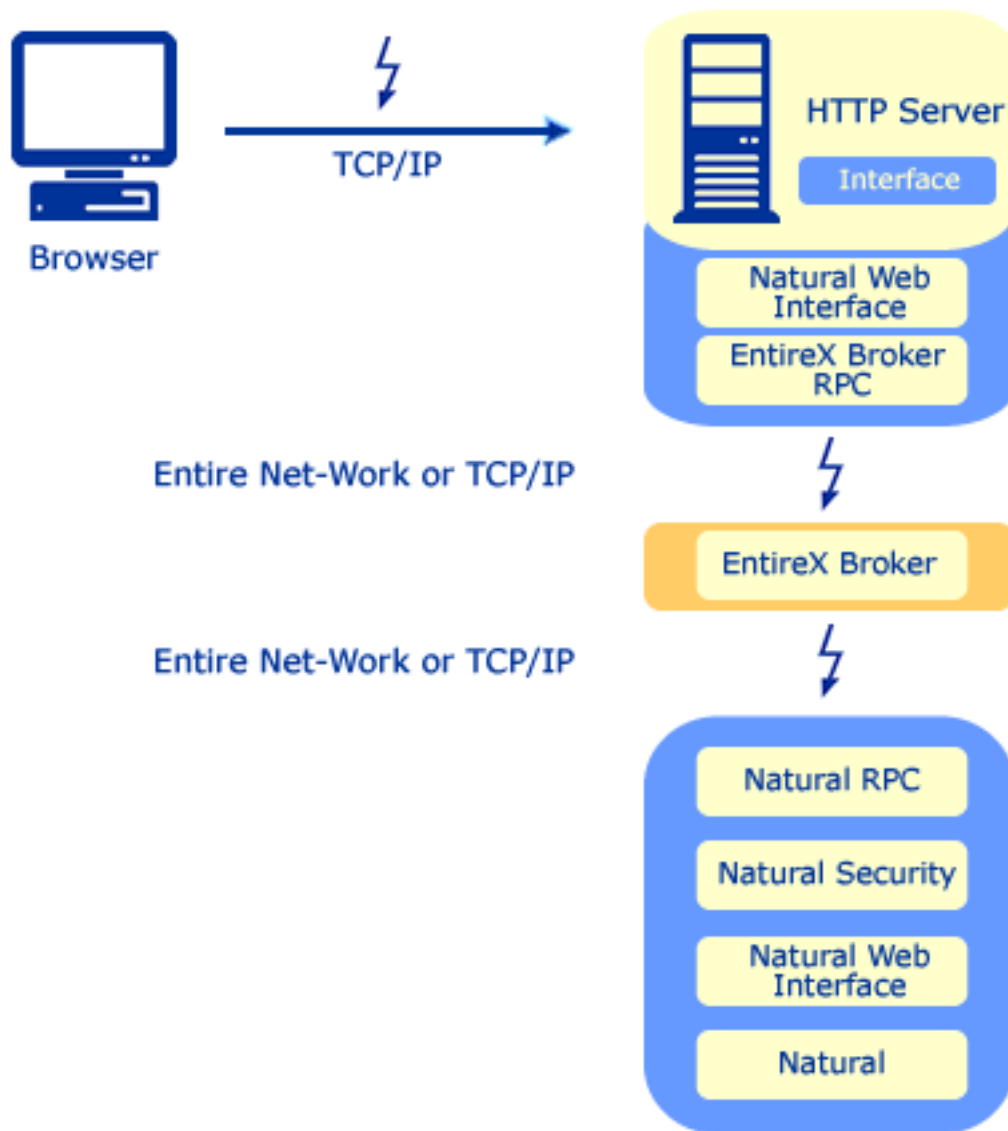
Natural ユーザーは、このようなアプリケーションを実装するために、新しいプログラミング言語を習得する必要がありません。ナビゲーションとユーザー入出力は、Natural に埋め込み HTML ステートメントを追加でいくつか使用することによって、完全に実装できます。

アーキテクチャ

次のトピックについて説明します。

- Natural RPC 技法を使用した通信

Natural RPC 技法を使用した通信



Natural Web インターフェイスモジュール

Natural Web インターフェイスは、以下の3つの内部モジュールで構成されています。

1. **Natural Web** インターフェイス
Natural の HTML API と HTTP API
2. **Natural Web** サーバーエクステンション
同一マシン上の Web サーバーにインターフェイスを提供する部分
3. 必要なミドルウェア
RPC テクノロジを使用する EntireX Broker を含む EntireX Communicator

機能

Web ページからの Natural サブプログラムの呼び出し

Natural Web インターフェイスの主要な機能の1つは、Web ページから Natural サブプログラムを呼び出し可能であることです。このことは、入力フィールドとボタンがある Web ページのフォームを使用して行うことができます。ユーザーは、データを入力し、ボタンの1つをクリックしてこれらのデータを送信することができます。これにより、ユーザーデータをパラメータとして渡す Natural サブプログラムが実行されます。

これにより、アプリケーション機能（サブプログラム）に簡単にアクセスできます。大部分の Web サーバーでは、SQL（および ODBC ドライバ）を使用してデータを取得する簡単なデータベースアクセスが提供されていますが、この機能は、対話式アプリケーションを実装するには十分ではありません。また、データ整合性およびユーザーデータの処理を保証するためのビジネスロジックも必要となります。

通常、整合性チェックや妥当性チェックなどのビジネスロジックは、実稼働アプリケーション向けに過去に実装されて、すでに存在しています。これらのロジックが別の Natural モジュール（サブプログラム、プログラム、サブルーチンなど）として実装されている場合は、容易に再利用できるため、異なる環境または異なる言語で再実装する必要がありません。

したがって、Web サーバーをビジネス機能に接続するために特別なインターフェイスプログラムを作成する必要はありません。Natural Web インターフェイスは、その目的のために標準化されたインターフェイスです。

プログラミング言語を学習する必要がないため、既存のスキルを活用できます（出力ページをフォーマットする HTML ステートメントを除く）。

フォーマットされた Web ページでのユーザーへのフィードバック

Web の対話式アプリケーションで 2 番目に重要となる機能は、フォーマットされた Web ページでユーザーにフィードバックすることです。Natural Web インターフェイスを使用すると、これらの Web ページをアプリケーションのニーズに従ってダイナミックにフォーマットできます。

このことには、レイアウトの制御およびこれらのページの内容が、外部の別のディレクトリにではなく、完全にアプリケーション/プログラムレベルにあるという利点があります。

また、Natural は多様なソース（Adabas、RDBMS、VSAM、およびシーケンシャルファイル、さらに Entire System Server ではシステム情報）からデータと情報を収集できるため、アプリケーションタイプは事実上無限です（Natural で構築できるアプリケーションはすべて Web と統合できます）。

実績あるミドルウェア

Natural Web インターフェイスは、Software AG の Entire 製品群の実績あるミドルウェア製品セットが基盤となっています。

これにより、既存のクライアント/サーバー環境にシームレスに統合することが可能になります。Web 接続は、既存の Natural サーバーに接続できる単なる別のクライアントとなります。Entire Net-Work がインストールされている場合は、別のミドルウェア製品セットをインストールする必要はありません。

Natural for Windows では、このインターフェイスで Natural DCOM クラスを呼び出すことができます。特定のインターフェイスで呼び出されたメソッドは、リモートプロシージャコール（RPC）を介して使用されるものと同じサブルーチンにマップできます。

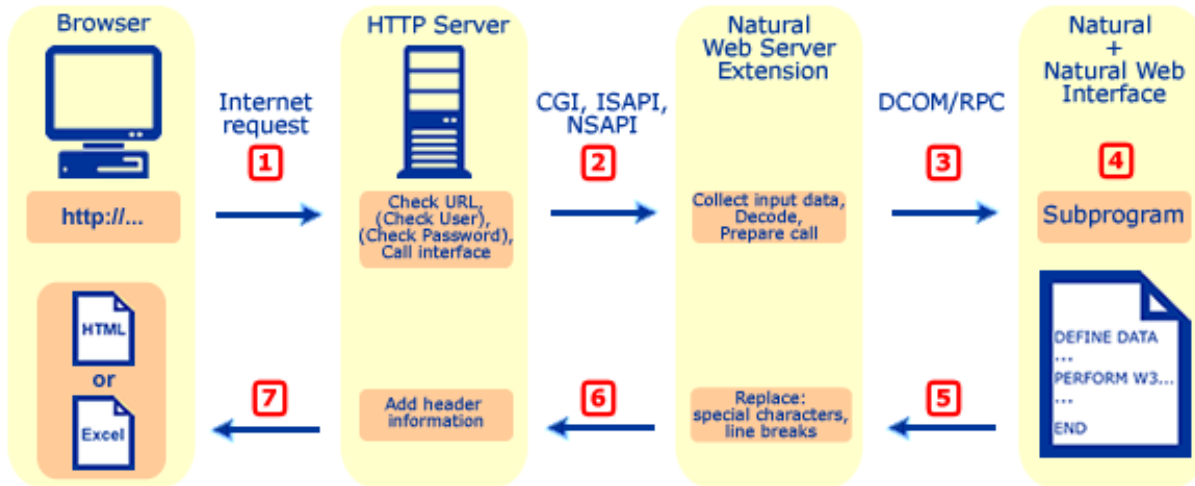
Web ページの作成

Web ページは、標準ツール（Macromedia Dreamweaver、Microsoft Frontpage など）または Natural 生成機能を使用した Web ページ作成ツールを使用して作成できます。Natural サーバーから、サブプログラムを生成できます。他のプログラミング言語や Web ページ作成ツールに関する知識を習得する必要はありません。

機能

ユーザーのブラウザの Web ページからの要求は、Web（または HTTP）サーバーに渡されます。この要求が Natural サブプログラムの実行を要求する形式である場合、この要求は EntireX RPC、PAL、または DCOM 経由で Natural サブプログラムを実行する Natural Web サーバーエクステンション部分に渡されます。プログラムは、ユーザーデータをパラメータとして取得し、その後、ユーザーにフィードバックを提供する一連のプログラムを発行します。

次の図は、Natural サブプログラムが HTML ブラウザからどのように呼び出されるかを示しています。処理の各段階は数字で識別されます。これらの段階で発生する処理の詳細については、以下で説明します。



1. HTML ブラウザは URL を要求します。
ブラウザは、サーバー側で呼び出すプログラムを識別する URL を要求します。
2. Web サーバーは、Natural Web サーバーエクステンション CGI を呼び出します。
Web サーバーは、URL を取得し、Natural Web サーバーエクステンションを呼び出します。
3. Natural Web サーバーエクステンションは、呼び出しを RPC に変換します。
Natural Web サーバーエクステンションプログラムでは、URL を Natural RPC に変換します。この Natural RPC によって、当初 URL で指定されていた Natural サーバープログラムが呼び出されます。
4. Natural サブプログラムが実行され、戻りページが生成されます。
サーバー上の Natural サブプログラムが実行され、HTML 戻りページが生成されます。
5. 戻りページは、Natural Web サーバーエクステンションに送り返されます。
HTML 戻りページは、サブルーチン呼び出しの応答として送り返されます。
6. Natural Web サーバーエクステンションは、戻りページを Web サーバーに送り返します。
Web サーバーは、ヘッダー情報を追加して、それをブラウザに送信します。
7. ブラウザは、URL に対する要求として送信したものへの回答を受信します。



注意: Natural Web インターフェイスのコンテキストでは、外部サブルーチンだけが出力を返すことができます。

セキュリティ

Natural Web インターフェイス経由で呼び出されたページは、Natural Security とともに動作することができます。このことは、以下のように行われます。

- 最初に、Natural Web サーバーエクステンションを、HTTP サーバーで制限されたページとして定義する必要があります。
- このように定義した後は、ブラウザでページを要求すると、ユーザー ID とパスワードの入力を求められます。
- 入力されたデータは、HTTP サーバーによって、データベースを使用して検証されます。
- ユーザーが認証されると、リモートユーザー名を使用して Natural Web サーバーエクステンションが呼び出されます。
- Natural RPC サーバーが Natural Security とともに開始されると、入力された名前が *USER として設定されます。
- HTTP サーバーによってすでに承認が行われているため、Natural 側でのパスワードチェックは行われません。したがって、Natural RPC サーバーは AUTO=ON で開始する必要があります。

また、2番目のシナリオとして、初期化ファイルが開始される時、Natural Security で Natural RPC サーバーと通信するために、特定の固定された定義済みユーザー ID とパスワードを設定するという方法もあります。「[Natural Security との通信](#)」も参照してください。

4 NaturalWeb インターフェイスのコンフィグレーション

このドキュメントには、以下のセクションが含まれています。

- **Natural Web インターフェイスの構成** NaturalWeb インターフェイスの構成方法について説明します。特定の製品の詳細については、対応するインストール手順を参照してください。
- **トラブルシューティング** 既知の問題に対するヒントを提供します。

5 Natural Web インターフェイスの構成

■ サポートされる HTTP サーバー	16
■ RPC および RPC サーバーの構成	16
■ Web インターフェイスの構成	18
■ HTTP サーバーの構成	19
■ Natural Security との通信	19

このセクションでは、Natural Web インターフェイスの構成方法について説明します。特定の製品の詳細については、対応する製品ドキュメントを参照してください。

このセクションでは、次のトピックについて説明します。

更新された最新のドキュメントは、Software AG のドキュメント サイト (<http://documentation.softwareag.com>) で公開されています。

サポートされる HTTP サーバー

オペレーティングシステム	HTTP サーバー
Windows (Intel)	<ul style="list-style-type: none">■ Microsoft Internet Information Server バージョン 5.0/6.0■ Apache バージョン 2.0.x■ Apache バージョン 2.2.x
(*)	<ul style="list-style-type: none">■ Apache バージョン 2.0.x■ Apache バージョン 2.2.x

RPC および RPC サーバーの構成

以下のコンフィグレーションの説明で、例に使用している ETB255 はブローカーの名前、NATWEB1 は RPC サーバーの名前です。

インストールおよびコンフィグレーションの詳細については、Natural RPC、*Entire Net-Work*、および *EntireX Communicator* のドキュメントを参照してください。

以下では次のトピックについて説明します。

- [現在のバージョンの Natural for Mainframes/UNIX/Windows](#)

- EntireX Communicator/EntireX 開発者キット

現在のバージョンの Natural for Mainframes/UNIX/Windows

Windows/UNIX システムで SYSWEB を使用する場合

RPC 環境で2つの追加の steplib にアクセスできるように NATPARM ファイルを変更するには、次の手順に従います。

- [Natural Execution Configuration] パラメータグループで、steplib パラメータサブセクションに2つの steplib (SYSWEB と SYSEXT) を追加します。

Windows/UNIX システムで SYSWEB3 を使用する場合

RPC 環境で2つの追加の steplib にアクセスできるように NATPARM ファイルを変更するには、次の手順に従います。

- [Natural Execution Configuration] パラメータグループで、steplib パラメータサブセクションに2つの steplib (SYSWEB3 と SYSEXT) を追加します。

メインフレーム環境で SYSWEB を使用する場合

Natural Security がインストールされている場合：

- 使用しているライブラリに steplib SYSWEB および SYSEXT を定義します。

Natural Security がインストールされていない場合：

- ライブラリ SYSWEB および SYSEXT の関連 FNAT システムファイルの DBID とファイル番号を入力して、ライブラリ SYSWEB の Natural プログラム WEB-STLB を修正します。必要に応じて、steplib を追加できます。
- プログラムを STOW します。
- RPC サーバーの STACK パラメータの値が STACK=(LOGON SYSWEB;WEB-STLB) に設定されている必要があります。

EntireX Communicator/EntireX 開発者キット

Windows システムの場合

環境変数の設定は必要ありません。

UNIX (すべてのプラットフォーム) の場合

EntireX 関連のすべての環境変数は HTTP サーバーによって渡される必要があります。

Web インターフェイスの構成

以下では次のトピックについて説明します。

- [Natural Web インターフェイス](#)
- [RPC 用の Natural Web サーバーエクステンション](#)
- [DCOM 用の Natural Web サーバーエクステンション](#)

Natural Web インターフェイス

メインフレーム、Windows、および UNIX 環境の場合は、コンフィグレーションは不要です。

RPC 用の Natural Web サーバーエクステンション

外部エディタを使用してコンフィグレーションファイルを調整します。

```
RPC_ETB_ID_NAME=ETB255
RPC_SERVER_NAME=NATWEB1
```

Natural RPC サーバーを非 ASCII 環境で実行している場合

コンフィグレーションファイルにパラメータ `NWW_OUT_CSS_TRANSLATE` を設定する必要があります。値は使用されるコードページに依存します。

DCOM 用の Natural Web サーバーエクステンション

ローカル DCOM (すべてのプラットフォーム)

ローカル通信の場合は、調整は不要です。

外部 DCOM (すべてのプラットフォーム)

外部通信の場合は、NaturalX ドキュメントを参照してレジストリを変更するか、外部エディタを使用してコンフィグレーションファイルを調整します。

```
DCOM_SERVER_NAME=NATWEBEXT
```

Windows (Internet Information Server) の場合

Internet Information Server を使用する場合は、匿名ログオン用のユーザー名を使用します。例えば、NATWEB を使用します。この場合、NATWEB がグループ USER に属しているか、または GUEST アカウントが有効になっている必要があります。

Windows (Apache) の場合

Apache Server を使用する場合は、httpd.conf に指定された User/Group のデフォルト設定をそのまま使用できます。

```
# User/Group: The name (or # number) of the user/group to run httpd as User nobody Group #1
```

HTTP サーバーの構成

Windows (Internet Information Server 5.0 および 6.0)

Internet Information Server を使用する場合は、匿名ログオン用のユーザー名を使用します。例えば、IUSR_NATWEB を使用します。この場合、IUSR_NATWEB がグループ USER に属しているか、または GUEST アカウントが有効になっている必要があります。

Natural Security との通信

EntireX 開発者キットの新しいバージョンでは、パスワードとユーザー ID の組み合わせを 2 つ使用できます。

1 つ目のユーザー ID は EntireX Security 経由でのアクセスに使用され、2 つ目は Natural Security 用に使用されます。

HTTP サーバーセキュリティは、3 番目のセキュリティシステムとして関係します。

HTTP サーバーセキュリティ

HTTP サーバーで NWW インターフェイスのアクセスを制限します。詳細については、HTTP サーバーのドキュメントを参照してください。

EntireX Security

コンフィグレーションファイルに NWW_USER_ID と NWW_PASSWORD を指定する必要があります。

Natural Security

2つ目のユーザーID/パスワード (RPC_USER_ID、RPC_PASSWORD) を指定する必要があります。

パラメータ USE_REMOTE_USER が有効になると、RPC_USER_ID が設定または上書きされます。
RPC_PASSWORD は変更されません。

パスワードなしでセキュリティをパスするには、Natural Security を "AUTO=ON" に設定する必要があります。RPC_USER_ID/RPC_PASSWORD のペアを設定しなかった場合は、既存の実装との互換性を確保するために NWW_USER_ID/NWW_PASSWORD が使用されます。

6 Web インターフェイスのトラブルシューティング

このセクションでは、既知の問題に関する情報を提供します。

エラー	説明	推奨される対処
NWW0003 .ini ファイルが見つかりません。	NWW 初期化ファイルが見つかりません。	<p>サーバーエクステンションの初期化ファイルを確認してください。</p> <ul style="list-style-type: none"> ■ このファイルは、実行可能ファイルと同じ名前で、拡張子が .INI である必要があります。 ■ サーバーエクステンションの初期化ファイルは、サーバーエクステンションの実行可能ファイルと同じディレクトリに置く必要があります。 ■ サーバーエクステンションがコマンドプロンプトから開始可能でも HTTP サーバーによる呼び出しで実行されない場合は、HTTP サーバーが開始されるディレクトリと同じディレクトリにコピーされた .INI ファイルを検出できるかどうかを確認します。
<p>NWW0011 ERX エラー 00000000 が発生しました。 重大度 = 成功</p> <p>メッセージ： ... 9999 NAT0935 パラメータ数が矛盾しています (サブプログラム...)。ライブラリ = ... プログラム = D3MENU.</p>	間違ったサブプログラムが呼び出されたか、または間違った steplib が使用されました。	<p>呼び出しを確認してください。</p> <ul style="list-style-type: none"> ■ 呼び出されたサブプログラムがパラメータデータエリア W3PARM を使用しているかどうかを確認します。 ■ nww* インターフェイスからの呼び出しの場合は、RPC サーバーが Steplib SYSWEB を使用しているかどうかを確認します。

エラー	説明	推奨される対処
		<ul style="list-style-type: none"> ■ nww3* インターフェイスからの呼び出しの場合は、RPC サーバーが Steplib SYSWEB3 を使用しているかどうかを確認します。 ■ 呼び出されたプログラムが、正しい SYSWEB/SYSWEB3 ライブラリでコンパイルされているかどうかを確認します。NAT-DIR（ドキュメントを参照）を呼び出し、コンパイル時に使用されたインターフェイスを確認します。
<p>NWW0011 ERX エラー 80010014 が発生しました。 重大度 = エラー 機能 = 65536 リターンコード = 20 サブ機能 = 3 場所 = 0</p> <p>メッセージ： ERX_E_SERVICE_NOT_AVAILABLE - ETB エラーコード 00070007</p>	<p>Natural RPC サーバーが開始されていないか、見つかりません。</p>	<p>RPC サーバーを確認してください。</p> <ul style="list-style-type: none"> ■ Natural RPC サーバーを開始します。 ■ または、NWW 初期化ファイルで RPC_SERVER_NAME を確認します。
<p>NWW0011 ERX エラー 80010014 が発生しました。 重大度 = エラー 機能 = 65536 リターンコード = 20 サブ機能 = 3 場所 = 0</p> <p>メッセージ： ERX_E_SERVICE_NOT_AVAILABLE - ETB エラーコード 02150148</p>	<p>ブローカーが開始されていないか、見つかりません。</p>	<p>ブローカーを確認してください。</p> <ul style="list-style-type: none"> ■ ブローカーと Natural RPC サーバーを開始します。 ■ または、NWW 初期化ファイルで RPC_SERVER_NAME と RPC_ETB_ID を確認します。
<p>ライブラリ W3RPCDMO のサブプログラム TEST の処理に失敗しました。</p> <p>メッセージ：ステータス=O、ライブラリ=W3RPCDMO、プログラム=NATSRVD、レベル=01、エラー=00082、行=4190 サブ機能=255 場所=0</p>	<p>ユーザーが呼び出したプログラムは存在しないか、またはアクセスできません。</p> <p>現時点では、Natural ライブラリをダイナミックに切り替えることはできません。</p>	<p>Natural を確認してください。</p> <ul style="list-style-type: none"> ■ プログラムが存在するかどうかを確認します。 ■ プログラムが存在する場合は、ログオンライブラリを確認するか、または、所定のライブラリが含まれている場合は、steplib または NATPARM を確認します。
<p>Natural RPC サーバークラッシュ。 WEB-ONL で同じサブプログラムに対するテストを実行すると、パラメータ</p>	<p>Natural RPC で配列の境界がチェックされません。</p>	<p>プログラムを再カタログしてください。</p>

エラー	説明	推奨される対処
3 (サブプログラム..) の配列定義に矛盾があることを知らせる WEB-ONL 1420 NAT0937 メッセージが返されま す。		
デモアプリケーションが動作しませ ん。	異なるファイル番号を 使用しています。	ライブラリ SYSWEB を再カタログしてくだ さい。
NAT3048 オープン (OP) 時のファイ ル / USERID が有効ではありません。	Natural が、異なる セッションに対して同 じ ETID を使用してい ます。	ETID パラメータを \$\$ に設定してくださ い。これにより、Natural の実行ごとに 1 つの新しい ETID が生成されます。

7 Natural Web インターフェイスの要点

Natural Web インターフェイスドキュメントのこの部分では、Natural Web インターフェイスを使用して、Web 対応の Natural サブプログラムを作成し、Web ブラウザからこれらのサブプログラムを呼び出してページを受け取る方法について説明します。

また、Natural Web インターフェイスの操作に関連する Software AG 製品の EntireX Communicator の機能についても概要を説明します。詳細については、EntireX Communicator のドキュメントを参照してください。

ユーザーは、HTML、Web ブラウザ、および Web ブラウザが動作する環境について基本的なことを知っている必要があります。また、クライアント/サーバー環境の Natural に関する十分な知識も必要です。

Natural Web インターフェイスドキュメントのこの部分には、次のセクションがあります。

- **Natural Web インターフェイスで** 環境の設定方法とサブプログラムの操作方法について説明します。
の作業
- **Natural Web サーバーエクステンション** Natural Web インターフェイスを使用して、Web 対応の Natural サブプログラムを作成し、Web ブラウザからこれらのサブプログラムを呼び出してページを受け取る方法について説明します。
- **プログラミングのヒント** より良い Web プログラムを構築するための Natural Web インターフェイスの使用に関するヒントを提供します。
- **管理** フォーマットの設定方法、エラーページの定義方法、HTML への変換方法、および URL のデコード方法について説明します。
- **デモアプリケーション - JavaScript なし** Natural Web インターフェイスの使用およびプログラミングをデモアプリケーションで示します。
- **デモアプリケーション - JavaScript 使用** より包括的なデモアプリケーションが用意されています。このデモアプリケーションでは、Java をサポートするブラウザが必要となります。
- **Natural Web インターフェイスの** Natural Web インターフェイスで作業しているときに表示される
エラーメッセージ 可能性のあるエラーメッセージのリストです。

Natural ライブラリ SYSWEBには、Natural Web インターフェイスのすべてのモジュールが含まれています。

8 Natural Web インターフェイスでの作業

- 環境の設定 28
- Natural でのサブプログラムの構築 29

このセクションでは、次のトピックについて説明します。

環境の設定

Web 環境側の前提条件

次のソフトウェアをインストールする必要があります。

Web クライアント上 ブラウザソフトウェア (Mozilla Firefox や Microsoft Internet Explorer など)

Web サーバー上 HTTP サーバーソフトウェア (Apache Server や Microsoft Internet Information Server など)

ミドルウェアの前提条件

通信が RPC で使用される場合は、異なる前提条件を満たす必要があります。

RPC Software AG 製品の EntireX Communicator のブローカーがインストールされている必要があります (インストールについては、EntireX Communicator のドキュメントを参照)。

Natural Web サーバーエクステンション部分は、Web ブラウザと Natural RPC サーバーとの通信に必要です。

Natural サーバー側の前提条件

Natural Web インターフェイスの **SYSWEB** について、次の前提条件を満たしている必要があります。

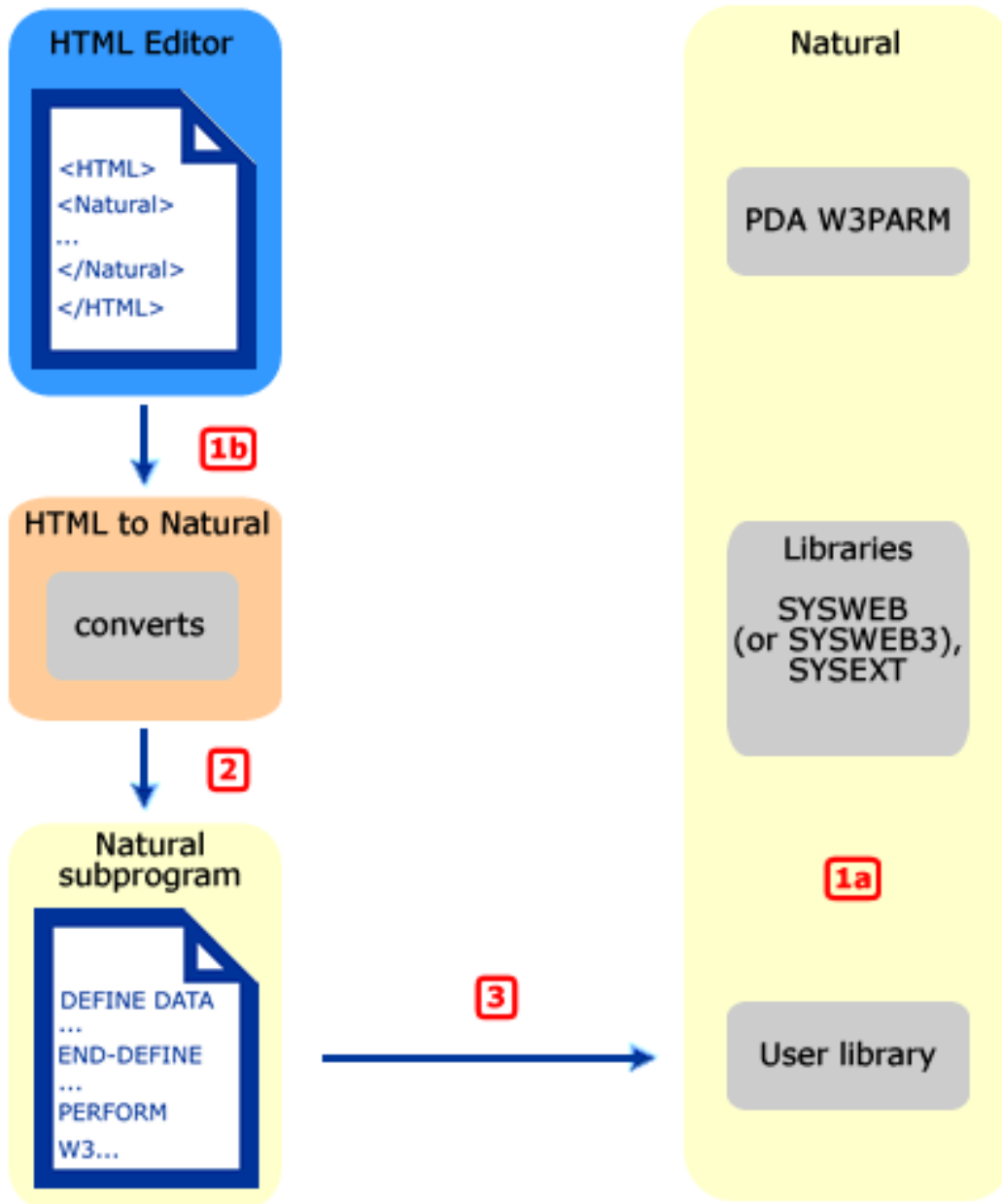
- 最新バージョンの Natural をインストールする必要があります。
- ライブラリ SYSWEB。
Natural の steplib が使用可能になっているか、ライブラリ SYSWEB の内容を SYSTEM ライブラリまたは RPC に呼び出されるユーザーライブラリにコピーしている必要があります。
- パラメータデータエリア W3PARM。
- Natural RPC スタブまたは NaturalX。

Natural でのサブプログラムの構築

以下の図は、サブプログラムを構築する方法を示しています。

1. HTML エディタを使用します。
2. HTML エディタを使用して、HTML および Natural コードを入力します。
3. 次に、それを Natural ソースに変換します。
4. 最後に、生成プログラムを Natural に移動します（Natural で直接コーディングします）。

処理の各段階は数字で識別されます。これらの段階で発生する処理の詳細については、以下で説明します。



1. ■ 1a. Natural コードは、ユーザーライブラリで作成されて保存されます。

サーバー側で、HTML タグをコードに含めることによって、または HTML タグを生成する事前に組み込まれたサブプログラムを呼び出すことによって、Natural コードを作成します。その後、サーバープログラムとして保存するか、サブプログラム WEB-WIZ を使用してデフォルトプログラムを生成します。

- 1b. Natural コードは HTML として入力されます。2 に進みます。

HTML エディタを使用して、HTML ページを作成します。

2. プログラム HTML2NAT によって、HTML から Natural ソースが生成されます。

ライブラリ SYSWEB のプログラム HTML2NAT を開始し、手順 1b で作成した HTML ページを変換します。

3. 生成された Natural ソースをユーザーライブラリに移動します。

サブプログラムを作成する前に

次のことに注意してください。

- HTML 戻りページは、送信可能な最大データに制限されます。この最大値は、戻りページ変数によって決まります。
- ライブラリ SYSWEB のサブルーチン W3INIT および W3END を呼び出して、Natural サーバーサブルーチンへのアクセスを初期化し、終了する必要があります。
- 常に、パラメータデータエリア W3PARM および W3CONST を使用してください。
- ユーザー独自のプログラムのためのフレーム（デフォルトプログラム）を生成するために、サブプログラム WEB-WIZ を使用してください。

サブプログラムを作成する方法

基本的な方法は 2 つあります。Natural でコーディングを直接開始するか、HTML エディタを使用します。

方法 1：Natural での直接コーディング

Natural で直接コーディングする方法も 2 つあります。

- 戻りページの SYSWEB サブルーチン（W3HTML や W3TEXT など）への呼び出しをプログラムエディタで入力します。ライブラリ SYSWEB のプログラムを確認します。これは、基本的なシステム関数の実行のみに役立ちます。この方法では、作成するデータタイプ（HTML や XML など）の深い知識が必要です。
- HTML タグを生成するサブプログラムを呼び出します。ライブラリ SYSWEB3（または SYSWEB）を確認します。ライブラリ SYSWEB のプログラムを使用して、基本的なシステム関数を実行できます。さらに、ライブラリ SYSWEB のプログラムで HTML タグが生成されます。この方法では、HTML の深い知識を必要とせずに、呼び出すプログラムを変更できます。

例：プログラムエディタでの **SYSWEB** サブルーチンへの呼び出しの入力

```
*
* Example E3END
*
DEFINE DATA
PARAMETER USING W3PARAM
LOCAL USING W3CONST
LOCAL
1 W3VALUE          (A250)
END-DEFINE
* --- ERROR HANDLING ---
ON ERROR
  PERFORM W3ERROR ##W3ERROR
  PERFORM W3END ##RPC
  ESCAPE ROUTINE
END-ERROR
*
* --- INITIALIZE W3 PROCESSING ---
PERFORM W3INIT ##RPC
*
* --- SET TYPE OF RETURN-PAGE ---
PERFORM W3CONTENT-TYPE 'text/html'
* --- WRITE THE DOCUMENT ---
PERFORM W3TEXT '<HTML><BODY><H2>Initialize</H2>'
*
* --- END THE HTML PAGE ---
COMPRESS '<HR>generated:' *DATE *TIME ##HTTP_NEWLINE
        '</BODY></HTML>' ##HTTP_END INTO W3VALUE
PERFORM W3TEXT W3VALUE
*
* --- END W3 PROCESSING ---
PERFORM W3END ##RPC
*
END
```

例：**HTML** タグを生成するサブプログラムの呼び出し

```
*
* Example E3IMAGE
*
DEFINE DATA
PARAMETER USING W3PARAM
LOCAL USING W3CONST
LOCAL
1 H3VALUE          (A250)
1 H3VALUE-MAX      (I004)
1 H3URL            (A250)
*
```

```
1 II (I001)
1 GIF (A064)
END-DEFINE
* --- ERROR HANDLING ---
ON ERROR
  PERFORM W3ERROR ##W3ERROR
  PERFORM W3END ##RPC
  ESCAPE ROUTINE
END-ERROR
*
* --- INITIALIZE W3 PROCESSING ---
PERFORM W3INIT ##RPC
*
* --- Pathname of picture ---
PERFORM W3READ-ENVIRONMENT "PICTURES" ' ' H3VALUE H3VALUE-MAX
IF H3VALUE-MAX EQ 0 THEN
  GIF := "/pictures"
ELSE
  GIF := H3VALUE
END-IF
*
* --- START HTML API ---
PERFORM H3-OPEN-HTML 'HTML Api -Image' " " " "
* --- THE LEVEL 2 HEADER ---
PERFORM H3-HEADER 2 'Image'
*
PERFORM H3-RULE 0
*
PERFORM H3-HEADER 4 'left:'
*
COMPRESS GIF '/natw_sam.gif' INTO H3URL LEAVING NO
PERFORM H3-IMAGE H3URL 'NATweb left' 219 229 "L"
*
FOR II 1 TO 10
  PERFORM H3-LINE-BREAK
END-FOR
PERFORM H3-RULE 80
*
PERFORM H3-HEADER 4 'small right:'
*
COMPRESS GIF '/natw_sam.gif' INTO H3URL LEAVING NO
PERFORM H3-IMAGE H3URL 'NATweb small right' 100 100 'R'
*
FOR II 1 TO 5
  PERFORM H3-LINE-BREAK
END-FOR
*
PERFORM H3-RULE 0
*
PERFORM H3-TIME_DATE
*
* --- END HTML API ---
```

```
PERFORM H3-CLOSE-HTML
* --- END W3 PROCESSING ---
PERFORM W3END ##RPC
*
END
```

方法2：HTML エディタの使用

2つの方法があります。

- スタティックページを作成します（HTMLを入力するだけで、Naturalサブプログラムに変換されます）。
- ダイナミックページを作成します（HTMLとNaturalプログラムコードを入力します）。

また、部分的にダイナミックで、部分的にスタティックなページを作成することもできます。

例：スタティックページの作成

```
<HTML>
<TITLE>NATweb - Test</TITLE>
<BODY bgColor=d3d3d3 >
<BR>
<center>
<h2>
This Natural subprogram was generated by a HTML page.
</h2>
</CENTER>
</BODY></HTML>
```

このNaturalサブプログラムは上述したHTMLページから生成されます。

```
* ----- SUBPROGRAM generated out of file:
* ----- C:\static.htm
DEFINE DATA
PARAMETER USING W3PARAM
LOCAL USING W3CONST
LOCAL
* ----- PRIVATE VARIABLES -----
1 W3VALUE          (A250)
END-DEFINE
*
* ----- ERROR HANDLER -----
ON ERROR
  PERFORM W3ERROR ##W3ERROR
  PERFORM W3END ##RPC
  ESCAPE ROUTINE
END-ERROR
```



```

* ----- INITIALISE HTTP API -----
PERFORM W3INIT ##RPC
* ----- HEADER FOR SERVER -----
PERFORM W3CONTENT-TYPE 'text/html'
*
PERFORM W3TEXTLINE '<HTML>'
PERFORM W3TEXTLINE '<TITLE>NATweb - Test</TITLE>'
PERFORM W3TEXTLINE '<BODY bgColor=d3d3d3 >'
PERFORM W3TEXTLINE '<BR>'
PERFORM W3TEXTLINE '<center>'
PERFORM W3TEXTLINE '<h2>'
PERFORM W3TEXTLINE 'This Natural subprogram was generated by a HTML page.'
PERFORM W3TEXTLINE '</h2>'
PERFORM W3TEXTLINE '</CENTER>'
PERFORM W3TEXTLINE '</BODY></HTML>'
* ----- END HTTP API -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
*
* ----- SUBROUTINES -----
*
END

```

例：ダイナミックページの作成

```

<Natural><!--
*
* Read form Pers-View starting with value given by the
* Parameter START
*
* Use HTML2NAT to generate a Natural Program
*
* 22.09.03
*
--></Natural>
<!-- Variables to read the environment --->
<Natural data><!--
* ----- DATA -----
1 H3VALUE      (A250)
1 H3MAX        (I4)
--></Natural>
<!-- Head of the HTML page --->
<HTML>
<TITLE>Natural - Environment Test</TITLE>
<BODY bgColor=d3d3d3 >
<BR>
<center>
<h2>
This Natural subprogram was generated by a HTML page. The program had been
precompiled out of a HTML page.

```

```
<br><br>
</h2>
</center>
<br>
<hr>
<! --- Subprogram to write the output to work file,
      from where the server will read it --- >
<Natural DATA><!--
1 #CONTENT (A1/1:48)
1 REDEFINE #CONTENT
  2 #PERSONNEL-NUMBER (N8)
  2 FILLER 1X
  2 #NAME (A20)
  2 FILLER 1X
  2 #FIRST-NAME (A15)
  2 FILLER 1X
  2 #AGE (N2)
--></Natural>
<Natural SUB><!--
* ----- Do the OUTPUT -----
DEFINE SUBROUTINE WRITELINE
  PERFORM W3TEXT "<LI>"
*
  #PERSONNEL-NUMBER:=PERSONNEL-NUMBER
  #NAME:=NAME
  #FIRST-NAME:=FIRST-NAME
  #AGE:=AGE
  PERFORM W3HTMLARRAY #CONTENT(*) 48
*
  PERFORM H3-LINE-BREAK
END-SUBROUTINE
--></Natural>
<UL><PRE>
<! --- Parameter used for reading data from the DATABASE --->
<Natural DATA><!--
* ----- DATA -----
1 #VALUE (A20)
1 PERS-VIEW VIEW OF PERSONNEL
  2 PERSONNEL-NUMBER
  2 NAME
  2 FIRST-NAME
  2 AGE
--></Natural>
<! --- Main program to read the data --->
<Natural NOT>
<LI>Value1
<LI>Value2
<LI>...
</Natural>
<Natural><!--
* --- READ ENVIRONMENT ---
PERFORM W3READ-ENVIRONMENT 'START' 'P' H3VALUE H3MAX
```

```

IF H3MAX GT 0 THEN
  #VALUE := H3VALUE
ELSE
  #VALUE := "A"
END-IF
*
* ----- MAIN -----
F. FIND (100) PERS-VIEW NAME > #VALUE
  IF NO
    COMPRESS 'Sorry nothing found for:' #value '!' INTO H3VALUE
    PERFORM W3HTMLLINE H3VALUE
  END-NOREC
  IF *NUMBER > 0
    PERFORM WRITELINE
  END-IF
END-FIND
*
IF *NUMBER(F.) > 0
  PERFORM H3-RULE 0
*
  COMPRESS 'well done for: ' #value '!' ##HTTP_END INTO H3VALUE
  PERFORM W3HTMLLINE H3VALUE
END-IF
--></Natural>
</PRE></UL>
<! --- The footer of the HTML page --- >
<hr>
<BR>
<center>
<A HREF="index.htm">back to Index</A>
This program has been generated.
<Natural><!--
PERFORM H3-TIME_DATE
--></Natural>
</P>
</CENTER>
</BODY></HTML>

```

この Natural サブプログラムは上述した HTML ページから生成されます。

```

* ----- SUBPROGRAM generated out of file:
* ----- C:\doit.htm
DEFINE DATA
PARAMETER USING W3PARM
LOCAL USING W3CONST
LOCAL
* ----- DATA -----
1 H3VALUE          (A250)
1 H3MAX            (I4)
1 #CONTENT (A1/1:48)

```

```
1 REDEFINE #CONTENT
  2 #PERSONNEL-NUMBER (N8)
  2 FILLER 1X
  2 #NAME (A20)
  2 FILLER 1X
  2 #FIRST-NAME (A15)
  2 FILLER 1X
  2 #AGE (N2)
* ----- DATA -----
1 #VALUE (A20)
1 PERS-VIEW VIEW OF PERSONNEL
  2 PERSONNEL-NUMBER
  2 NAME
  2 FIRST-NAME
  2 AGE
* ----- PRIVATE VARIABLES -----
1 W3VALUE (A250)
END-DEFINE
*
* ----- ERROR HANDLER -----
ON ERROR
  PERFORM W3ERROR ##W3ERROR
  PERFORM W3END ##RPC
  ESCAPE ROUTINE
END-ERROR
* ----- INITIALISE HTTP API -----
PERFORM W3INIT ##RPC
* ----- HEADER FOR SERVER -----
PERFORM W3CONTENT-TYPE 'text/html'
*
* ----- MAIN PROGRAM -----
*
* Read form Pers-View starting with value given by the
* Parameter START
*
* Use HTML2NAT to generate a Natural Program
*
* 22.09.2003
*
PERFORM W3TEXTLINE '<! --- Variables to read the environment --->'
PERFORM W3TEXTLINE '<! --- Head of the HTML page --->'
PERFORM W3TEXTLINE '<HTML>'
PERFORM W3TEXTLINE '<TITLE>Natural - Environment Test</TITLE>'
PERFORM W3TEXTLINE '<BODY bgColor=d3d3d3 >'
PERFORM W3TEXTLINE '<BR>'
PERFORM W3TEXTLINE '<center>'
PERFORM W3TEXTLINE '<h2>'
PERFORM W3TEXTLINE 'This Natural subprogram was generated by a HTML page. Th'
  -'e program had been'
PERFORM W3TEXTLINE 'precompiled out of a HTML page.'
PERFORM W3TEXTLINE '<br><br>'
PERFORM W3TEXTLINE '</h2>'
```

```

PERFORM W3TEXTLINE '</center>'
PERFORM W3TEXTLINE '<br>'
PERFORM W3TEXTLINE '<hr>'
PERFORM W3TEXTLINE '<! --- Subprogram to write the output to work file'
PERFORM W3TEXTLINE '      from where the server will read it --- >'
PERFORM W3TEXTLINE '<PRE>'
PERFORM W3TEXTLINE '<! --- Parameter used for reading data from the'
  -' DATABASE --->'
PERFORM W3TEXTLINE '<! --- Main Program to read the data --->'
* --- READ ENVIRONMENT ---
PERFORM W3READ-ENVIRONMENT 'START' 'P' H3VALUE H3MAX
IF H3MAX GT 0 THEN
  #VALUE := H3VALUE
ELSE
  #VALUE := "A"
END-IF
*
* ----- MAIN -----
F. FIND (100) PERS-VIEW NAME > #VALUE
  IF NO
    COMPRESS 'Sorry nothing found for:' #value '!' INTO H3VALUE
    PERFORM W3HTMLLINE H3VALUE
  END-NOREC
  IF *NUMBER > 0
    PERFORM WRITELINE
  END-IF
END-FIND
*
IF *NUMBER(F.) > 0
  PERFORM H3-RULE 0
*
  COMPRESS 'well done for: ' #value '!' ##HTTP_END INTO H3VALUE
  PERFORM W3HTMLLINE H3VALUE
END-IF
PERFORM W3TEXTLINE '</PRE>'
PERFORM W3TEXTLINE '<! --- The footer of the HTML page --- >'
PERFORM W3TEXTLINE '<hr>'
PERFORM W3TEXTLINE '<BR>'
PERFORM W3TEXTLINE '<center>'
PERFORM W3TEXTLINE '<A HREF="index.htm">back to Index</A>'
PERFORM W3HTMLLINE 'This program has been generated.'
PERFORM H3-TIME_DATE
PERFORM W3TEXTLINE '</P>'
PERFORM W3TEXTLINE '</CENTER>'
PERFORM W3TEXTLINE '</BODY></HTML>'
* ----- END HTTP API -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
*
*
* ----- SUBROUTINES -----
* ----- Do the OUTPUT -----

```

```
DEFINE SUBROUTINE WRITELINE
  PERFORM W3TEXT "<LI>"
*
  #PERSONNEL-NUMBER:=PERSONNEL-NUMBER
  #NAME:=NAME
  #FIRST-NAME:=FIRST-NAME
  #AGE:=AGE
  PERFORM W3HTMLARRAY #CONTENT(*) 48
*
  PERFORM H3-LINE-BREAK
END-SUBROUTINE
END
```

プログラミング全般に関する考慮事項

ローカルデータエリア W3CONST の定数値

ローカルデータエリア W3CONST には、便利な定数値が多数あります。

##HTTP_NEWLINE, ##HTTP_NEWLINE_LENGTH

##HTTP_NEWLINE 文字列を HTML に入力すると、W3TextDynamic を使用して #HTTP_NEWLINE を文字列に圧縮することによって、実際の新しい行を作成するために、ライブラリ SYSWEB3 (または SYSWEB) 内の、先頭が W3TEXT であるすべてのサブルーチンを使用できます。

##W3ERROR

W3ERROR の呼び出しに使用されるパラメータ。

##HTML_LT

"小なり" 記号 (<) の HTML 定数値。

##HTML_GT

"大なり" 記号 (>) の HTML 定数値。

##HTML_AMP

"アンパサンド" 記号 (&) の HTML 定数値。

##HTML_QUOT

"二重引用符" 記号 (") の HTML 定数値。

##HTML_REG

"登録商標" 記号の HTML 定数値。

##HTML_COPY

"著作権" 記号の HTML 定数値。

##HTML_NBSP

"改ページなし" スペース (') の HTML 定数値。

値で定義された変数

すべての入力変数は、BY VALUE で定義されます。つまり、MOVE 互換のすべての値、特に文字列を使用できます。

次ページの作成

出力が戻りページの上限を超えている可能性がある場合に、戻りページに何バイトの空きがあるかを評価するには、ライブラリ SYSWEB のサブルーチン W3COUNTER を使用します。

サブプログラムのテスト

サブプログラムをテストする方法は 3 つあります。

SYSWEB を使用している場合：

1. Web ブラウザからサブプログラムを呼び出します。
2. ライブラリ SYSWEB のサブプログラム NAT-DIR を呼び出して、Natural ライブラリの内容を確認します。また、パラメータでライブラリの名前を指定することもできます。例えば、`http://.../sysweb/NAT-DIR?LIB=SYSEXT` です。プログラムをクリックして開始します。
3. Web ブラウザからサブプログラムを呼び出さない場合は、Natural プログラム WEB-ONL を使用してリモートコールをシミュレーションできます。このプログラムの出力は Natural テキストオブジェクトとして保存されます。この "オンライン実行" では、Natural デバッガを使用できます。

Natural Web サーバーエクステンション

Natural Web サーバーエクステンションは、HTTP サーバーから呼び出されます。プログラムは、HTTP サーバーから受け取るパラメータを再パッケージし、指定された Natural サブプログラムまたはメソッドへの Entire Broker RPC または DCOM 呼び出しを実行します。

パラメータ

HTTP サーバーによって送信されたデータは認識され、前処理されます。URL デコードされた（修正された）フォームで HTTP サーバーに送信された URL は、元の状態にリセットされます。すべての非バイナリデータはデータとして送信可能であり、必要に応じて、ASCII から EBCDIC に、またはその逆に変換されます。

初期化ファイル

HTML ページに指定された変数のみが、呼び出されたサブプログラムに自動的に転送されます。転送される他のすべての変数は、.ini ファイルの ENV= エントリに指定する必要があります。この方法で、システム環境変数として処理される変数を追加できます。システム環境変数を追加するには、.ini ファイルに SETENV= エントリを指定します。

.ini ファイルの例

```
ENV=HTTP_REFERER
ENV=HTTP_HOST
;
SETENV=VERSION:=alpha
SETENV=BROKER:=local
```

エラーのログ

サーバーからファイルに送られた最後の HTML ページを保存するには、コンフィグレーションファイルに TRACE_FILE パラメータを指定します。

エラーログを返すには、コンフィグレーションファイルにログファイル名として ERROR_LOG_FILE パラメータを指定します。

独自のエラー画面を表示するには、コンフィグレーションファイルに ERROR_TEMPLATE パラメータを目的の HTML エラーページ名で指定します。接頭辞 "\$" を使用することによって、環境変数を HTML エラーページ内に指定できます。環境変数 \$NWW_ENVIRONMENT を使用すると、呼び出されたサブルーチンに送信されたすべての環境変数がコメント行としてエラーページに書き込まれます。

ライブラリ SYSWEB の命名規則

サブルーチン W3*

W3* サブルーチンは、Natural Web サーバーエクステンションの HTTP サーバーへのインターフェイスにアクセスします。このようなインターフェイスは、基本的に、パラメータデータエリアと、送信されたデータのログで構成されています。サブプログラムで使用される W3* サブルーチンは、Natural RPC を使用する HTTP サーバーに呼び出されます。

サブルーチン H3*

サブルーチンの 1 つから H3* サブルーチンを呼び出すと、基本的な HTML タグが作成されます。

サブプログラム NAT*

NAT* サブプログラムは、インターネットから呼び出すことができるユーティリティです。

Natural テキストメンバ T3*

T3* テキストメンバには、ライブラリ SYSWEB の内容と、渡すことのできるサブルーチン名およびパラメータが記載されています。それらをどのように呼び出すかのコードサンプルも提供します。このオンラインドキュメントにアクセスするには、ユーティリティ nat-docu を使用します。

サブプログラム E3*

オンラインドキュメントのサンプルコード。

メンバ D3* および D4*

D3* および D4* メンバは、デモアプリケーションです。

プログラム **Web***

Web* プログラムは、Natural の NEXT プロンプトから実行できるユーティリティです。

9 Natural Web サーバーエクステンション

このドキュメントは以下のセクションで構成されます。

- **SYSWEB** を使用する場合の概要 **SYSWEB** を使用する場合の Natural Web サーバーエクステンションの操作およびインストール手順について説明します。
- 初期化ファイル 初期化ファイルのパラメータと変数について説明します。
- エラーメッセージ エラーのリストです。

10 Natural Web サーバーエクステンションの概要 - SYSWEB を使用する場合

▪ 一般的な情報	48
▪ インストール - RPC	48
▪ 変換	49
▪ 変数	49
▪ エラーログとメッセージ	49
▪ プログラムの呼び出し	50

このセクションでは、次のトピックについて説明します。

全般的な情報

Natural Web サーバーエクステンション部分は、基本的には HTTP サーバーから呼び出されるプログラムです。Natural Web サーバーエクステンションは、HTTP サーバーから渡されたパラメータを再パッケージし、要求された Natural プログラムへのブローカー RPC 呼び出しを標準パラメータデータエリアを使用して実行します。呼び出しは、EntireX Communicator に付属している EntireX Broker によって転送されます。

Natural バージョン 4.1 と同様に、3つの HTTP サーバーインターフェイスがサポートされます。

- サポートされているサーバーおよびプラットフォーム用の Common Gateway Interface (CGI) 。
- Windows 上の Microsoft Internet Information Server 専用 Internet Server Application Programming Interface (ISAPI) 。
- Netscape FastTrack Server 専用 Netscape Server Application Programming Interface (NSAPI) 。

インストール - RPC

各 Natural Web サーバーエクステンションは、2つのファイルから構成されています。

- 実行可能ファイル
- 初期化ファイル

これらのファイル名は変更できます。初期化ファイルは実行可能ファイルと同じ名前ですが、拡張子が .ini です。この2つのファイルは同じディレクトリに存在する必要があります。

ファイルを Web サーバーの適切な場所にコピーするか、Web サーバーがファイルに直接アクセスするように Web サーバーをパラメータ化します。

	RPC
CGI	nwwcgi nwwcgi.ini
ISAPI	nwwisapi.dll nwwisapi.ini
NSAPI	nwwnsapi.dll nww/nsapi
パラメータ	RPC_ETB_ID_NAME = ブローカー名 RPC_SERVER_NAME = サービス名 NWW_INOUT_LENGTH = データ転送量



注意: 一部の HTTP サーバーは、拡張子 .exe のない実行可能ファイルに対応しています。このため、.exe 拡張子のある実行可能ファイルとない実行可能ファイルの両方を使用できます。

変換

インターフェイス経由で HTTP サーバーによって送信されるパラメータは、特定の変数または転送エリアによって指定されます。転送エリアまたは変数 QUERY_STRING に含まれているユーザーデータは認識され、前処理されます。特に、URL のエンコードは元に戻されます。

Natural Web サーバーエクステンションの仕様では、非バイナリデータの送信のみ可能になっています。これは、データが必要に応じて ASCII と EBCDIC の間で変換されるためです。

変数

HTML ページに指定された変数だけが、呼び出されたプログラムに自動的に転送されます。HTTP サーバーから使用できる他の変数を指定する必要があります。

各変数が転送されるには、[初期化ファイル](#)にエントリが必要です。

また、システム環境変数として処理される変数を追加することもできます。

エラーログとメッセージ

特定の HTML ページを使用して独自のエラー画面を設定できます。環境の変数はこのエラーページで指定できます。

最後に転送したページをファイルにコピーでき、またエラーをエラーログファイルに書き込むことができます。

プログラムの呼び出し

ブラウザからプログラムを呼び出すには、HTTPサーバーの名前と CGI対応ディレクトリ（Natural Web サーバーエクステンションのファイルをコピーした場所）の名前を含む Uniform Resource Locator（URL）を指定する必要があります。続けて、Natural Web サーバーエクステンションプログラム名、Natural ライブラリ名、Natural サブプログラム名の順に指定する必要があります。

RPC の URL	
CGI	<code>http://server-name/cgi-library/nwwcgi/your-library/your-program</code>
ISAPI	<code>http://server-name/cgi-library/nwwisapi.dll/your-library/your-program</code>
NSAPI	<code>http://server-name/nww/nsapi/your-library/your-program</code>

11 Natural Web サーバーエクステンション-初期化ファ イル

■ 全般的な情報	52
■ RPC パラメータ	52
■ DCOM パラメータ	53
■ Natural Web サーバーエクステンション設定	53
■ HTTP サーバー変数	56
■ 追加の変数	56
■ エラーテンプレート	56

このセクションでは、次のトピックについて説明します。

全般的な情報

Natural Web サーバーエクステンションでは、初期化ファイルのランタイムパラメータを処理します。実行可能ファイルは、現在の作業ディレクトリにある、同名で拡張子が .ini である初期化ファイルを探します。

すべての変数は WWW で使用されるため、変数名の大文字と小文字は区別されません。変数は最大 72 文字です。空白は文字として認識されるため、パラメータを複数回指定できます。

RPC パラメータ

これらのパラメータは、EntireX RPC との通信に必要です。

パラメータ	説明
RPC_CLASS_NAME	使用されるサービスのクラスを定義します。 常に RPC を使用します。
RPC_ETB_ID_NAME	呼び出される EntireX Broker の名前です。
RPC_NO_LOGON	URL に指定されたライブラリにログオンします。デフォルトは 0 です。
RPC_SERVER_NAME	呼び出される Broker サービスの名前です。
RPC_SERVICE_NAME	呼び出されるサービスを定義します。 常に CALLNAT を使用します。
RPC_TIME_OUT	呼び出しのタイムアウトを定義します。 デフォルトは 7000 です。
RPC_USER_ID	RPC で使用されるユーザー ID です。 指定しなかった場合は、以下のいずれかが使用されます。 ■ NWW_USER_ID ■ REMOTE_USER (REMOTE_USER が 1 に設定されている場合)
RPC_PASSWORD	RPC で使用されるユーザーパスワードです。 指定しなかった場合は、NWW_PASSWORD が使用されます。
RPC_SSL_PARAMETER	SSL を使用する RPC の接続文字列です。

DCOM パラメータ

このパラメータは、DCOM との通信に必要です（Windows プラットフォームのみ）。

パラメータ	説明
DCOM_SERVER_NAME	呼び出される DCOM サーバーの名前です。 同じコンピュータで Natural サーバーを実行していない場合のみ指定します。

Natural Web サーバーエクステンション設定

このグループのパラメータは、Natural Web サーバーエクステンションの設定を定義します。

パラメータ	説明
ECHO_ENVIRONMENT	このパラメータは、デフォルトエラーページを使用する場合にのみ役立ちます。このパラメータを1（ユーザー定義エラーページの \$NWW_ENVIRONMENT と同じ値）に設定すると、すべての環境変数がコメント行としてエラーページに書き込まれます。
ERROR_LOG_FILE	エラーログ用のファイルを定義します。このパラメータを指定しなかった場合は、ログが無効になります。 各ログエントリは、レイアウトが同一であり、CGI 文字列を検索することでエラーログファイル内で簡単に見つけることができます。 ログエントリの例： [Thu Jun 28 10:51:19 2005] nwwcgi.exe 04.02.05 Win32: processing of /cgi-bin/nwwcgi.exe failed for Lib:{library} Sub:{subprogram} Path:{path_info}, for natweb.software-ag.de reason NWW0001 No subprogram and library specified.
ERROR_STDERR	このパラメータを1に設定すると、すべてのエラーが stderr 経由で記録されます。ログファイルの場所は、使用する HTTP サーバーとパラメータ化の方法によって異なります。 ERROR_LOG_FILE も参照してください。 一部の HTTP サーバーでは、stderr の使用をサポートしていません。
ERROR_TEMPLATE	エラーテンプレートファイルを定義します。このパラメータを指定しなかった場合は、デフォルトエラーページが生成されます。後述の「 エラーテンプレート 」を参照してください。
NWW_INOUT_FORMAT NWW_INOUT_LENGTH	注意: これらのパラメータは SYSWEB でのみ使用します。 転送データ量を定義します。これらのパラメータは、IDL ファイルのパラメータ Out_Page の次元を定義します。

パラメータ	説明																		
	<p>使用される IDL ファイル：</p> <pre> DEFINE DATA PARAMETER 1 Version-Nr (A15) In 1 Log-Time (A30) In 1 Out_Page (A RPC_INOUT_FORMAT 1:RPC_INOUT_LENGTH) In Out 1 Out_Page_Count(I04) In Out 1 Result (I04) Out END-DEFINE </pre>																		
NWW_PASSWORD	ユーザー ID のパスワードを定義します。																		
NWW_PATH_INFO	<p>Natural Web サーバーエクステンションをスタンドアロンモード（テスト環境）でテストするには、このパラメータにライブラリとプログラム名を指定します。Natural Web サーバーエクステンションを通常モード（HTTP サーバー）で使用する場合は、このパラメータを無効にする必要があります。</p> <p>例： NWW_PATH_INFO=/syshtml/nat-env</p>																		
NWW_PATHINFO_PREFIX	<p>このパラメータは、ISAPI インターフェイスを使用する場合にのみ使用できます。このインターフェイスがアプリケーションマッピング（ディレクトリ nww と拡張子 .nww など）として定義されている場合、PATH_INFO 変数の値はディレクトリとファイル名が先頭に付加された URL になります（例：<i>/nww/my.nww/sysweb/nat-env</i>）。この接頭辞（斜体で示した部分）は削除する必要があります。このパラメータを使用して、指定した接頭辞を削除できます。</p> <p>例： NWW_PATHINFO_PREFIX=/nww/my.nww</p>																		
NWW_OUT_CSS	<p>文字列を特定の文字に置換します。</p> <table border="0"> <tr> <td>文字列</td> <td>文字</td> </tr> <tr> <td>&#09;</td> <td>--> (タブ)</td> </tr> <tr> <td>&#64;</td> <td>@</td> </tr> <tr> <td>&#91;</td> <td>[</td> </tr> <tr> <td>&#92;</td> <td>\</td> </tr> <tr> <td>&#93;</td> <td>]</td> </tr> <tr> <td>&#123;</td> <td>{</td> </tr> <tr> <td>&#124;</td> <td>/</td> </tr> <tr> <td>&#125;</td> <td>}</td> </tr> </table> <p>この設定は、カスケードスタイルシートが使用されていて、EBCDIC コードを使用するコンピュータに RPC サーバーが配置されている場合に便利です。デフォルトは 0 です。1 を使用すると有効になります。</p>	文字列	文字			--> (タブ)	@	@	[[\	\]]	{	{	|	/	}	}
文字列	文字																		
		--> (タブ)																		
@	@																		
[[
\	\																		
]]																		
{	{																		
|	/																		
}	}																		
NWW_OUT_CSS_TRANSLATE	指定された文字を対応する 16 進値に置換します。																		

パラメータ	説明
	<p>(ASCII のデフォルト値)</p> <p>文字 16 進値</p> <p>--> (タブ) 09</p> <p>@ 40</p> <p>[5B</p> <p>\ 5C</p> <p>] 5D</p> <p>{ 7B</p> <p> 7C</p> <p>} 7D</p> <p>英語 EBCDIC (コードページ 37) の例 :</p> <p>#(tab), @, [, \,], {, , }</p> <p>NWW_OUT_CSS_TRANSLATE=05,7C,AD,61,BD,C0,4F, D0</p>
NWW_USER_ID	RPC で使用されるユーザー ID です。
NWW_RETRY	このパラメータは、NAT3009エラー (トランザクションの中止) が発生した場合に、プログラムを再度呼び出す回数を定義します。デフォルトは 3 です。
INI_RELOAD	最初の呼び出し中に初期化ファイルを 1 回だけロードします。CGI インターフェイス用ではありません。デフォルトは 1 です。
REMOVE_USER_DOMAIN	NT 上の IIS サーバーは、REMOTE_USER として、ユーザーが属するドメインの名前が先頭に付加されたユーザー名を渡します。Natural が処理できるユーザー名は最大 8 文字です。USE_REMOTE_USER を 1 に設定し、REMOVE_USER_DOMAIN も 1 に設定した場合は、指定した REMOTE_USER 名の使用ドメイン名が削除されます。つまり、最後の "/" 以降の情報がユーザー名として Natural に渡されます。
TRACE_FILE	ファイル名を指定した場合は、HTTP サーバーに返された最後のページがこのファイルに保存されます。このパラメータを指定した場合は、出力は書き込まれません。
USE_REMOTE_USER	RPC_USER_ID を REMOTE_USER に置き換えます。1 に設定すると有効になります。

HTTP サーバー変数

呼び出したプログラムに転送されるすべての HTTP サーバー変数を指定する必要があります。これを行うには、変数 ENV に、転送する変数の名前を指定します。ENV 変数は複数回指定できます。

有効な変数の一部を示します。

```
ENV=REMOTE_HOST
ENV=REMOTE_ADDR
ENV=SCRIPT_NAME
ENV=HTTP_REFERER
ENV=HTTP_HOST
ENV=HTTP_COOKIE
```

変数の詳細については、<http://hoohoo.ncsa.uiuc.edu/cgi/env.html> を参照してください。

追加の変数

Natural Web サーバーエクステンションでは、呼び出したプログラムに追加の変数を転送できます。これを行うには、変数 SETENV に、転送する変数の名前、:=、および値を指定します。SETENV 変数は複数回指定できます。

例：

```
SETENV=PICTURES:=/pictures
```

エラーテンプレート

デフォルトエラーレポート

パラメータ ERROR TEMPLATE が指定されていない場合は、デフォルトが使用されます。

以下にデフォルトエラーレポートの例を示します。

nwwcgi.exe Error Report	
<i>Natural Web Interface NWW5100c Win32</i>	
The following error has been logged in the error log file:	
/cgi-bin/nwwcgi.exe:	processing of subprogram/method NAT-INFO at library/class SYSWEB failed.
reason:	NWW0011 ERX error 80010014 occurred. Severity = Error Facility = 65536 Returncode = 20 Subfacility = 3 Location = 0 Message: ERX_E_SERVICE_NOT_AVAILABLE - ETB error code 02150148
for:	pcnatweb.software-ag.de:80
path:	/sysweb/nat-info
NWW Error - Fri Mar 15 10:20:28 2005	<u>Natural</u>

独自のエラーテンプレートの指定

独自のエラーテンプレートを指定することもできます。基本的に、エラーテンプレートは通常の戻りページです。他の戻りページと同様に、コンテンツタイプを設定する必要があります。唯一の追加事項は、変数の置き換えです。このことを行うには、先頭が\$記号である環境変数を指定します。下記の「[エラーテンプレートの例](#)」を参照してください。

エラーテンプレートには、次の環境変数を追加できます。

環境変数	説明
NWW_LOGTIME	エラーが記録される日時（ <code>ERROR_LOG_FILE</code> が指定されている場合）。
NWW_VERSION	Natural Web サーバーエクステンションのバージョン番号。
NWW_RUN	呼び出されたプログラムの名前。
NWW_ERROR	発生したエラーの件数。
NWW_LIBRARY NWW_CLASS	呼び出されたライブラリ/クラスの名前。

環境変数	説明
NWW_SUBPROGRAM NWW_METHOD	呼び出されたサブプログラム/メソッドの名前。
NWW_ENVIRONMENT	すべての環境変数がコメント行としてエラーページに書き込まれます。

エラーテンプレートの例

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
  <META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <TITLE>$NWW_RUN Error Report - $NWW_LOGTIME</TITLE>
</HEAD>
<BODY bgcolor="#FFFFFF" text="#000000">
<TABLE border="0" width="100%" cellspacing="0" cellpadding="5">
  <TR bgcolor="#CCFFCC">
    <TD align="center">
      <H2 align="center">
        $NWW_RUN Error Report
      </H2>
      <P align="center">
        <I><SMALL>Natural Web Server Extension Interface: $NWW_VERSION</SMALL></I></TD>
    </TR>
    <TR>
      <TD align="center">
        <B>The following error has been logged in the error log file:</B></TD>
    </TR>
  </TABLE>
<TABLE border="0" width="100%" cellspacing="15" cellpadding="0">
  <TR valign="top">
    <TD align="right"><B>$SCRIPT_NAME:</B></TD>
    <TD align="left"><TT>processing of subprogram/method <B>$RPC_SUBPROGRAM</B><BR>
      at library/class <B>$RPC_LIBRARY</B> failed.</TT></TD>
  </TR>
  <TR valign="top">
    <TD align="right"><B>reason:</B></TD>
    <TD align="left"><PRE>$RPC_ERROR
  </PRE>
  </TD>
  </TR>
  <TR valign="top">
    <TD align="right"><B>for:</B></TD>
    <TD align="left"><TT>$SERVER_NAME:$SERVER_PORT</TT></TD>
  </TR>
  <TR valign="top">
    <TD align="right"><B>path:</B></TD>
    <TD align="left"><TT>$PATH_INFO</TT></TD>
  </TR>
</TABLE>
<TABLE border="0" width="100%" cellspacing="0" cellpadding="5">
```



```
<TR bgcolor="#CCFFCC">
  <TD>NWW Error Template - $NWW_LOGTIME</TD>
  <TD align="right">Natural</TD>
</TR>
</TABLE>
<P>
$NWW_ENVIRONMENT
</BODY></HTML>
```


12 NaturalWeb サーバーエクステンション・エラーメッセージ

このセクションでは、NaturalWeb サーバーエクステンションで作業しているときに表示される可能性のあるエラーメッセージについて説明します。

エラー番号	エラーメッセージ	説明	ユーザー	プログラマ	管理者
NWW0001	ライブラリおよびサブプログラムが指定されていません。	指定した URL が正しくありません。ライブラリとサブプログラムの名前がありません。	URL を修正してください。	なし	なし
NWW0002	ライブラリが指定されていません。	指定した URL が正しくありません。	URL を修正してください。	なし	なし
NWW0003	ファイル ... が見つかりません。	アダプタの初期化ファイルが見つかりません。	なし	なし	インストールを確認してください。
NWW0004	サブプログラムが指定されていません。	指定した URL が正しくありません。	URL を修正してください。	なし	なし
NWW0010	RPC 呼び出しに失敗しました。	EntireX RPC を初期化できません。	なし	なし	インストールを確認してください。

エラー番号	エラーメッセージ	説明	ユーザー	プログラマ	管理者
NWW0011	ERX エラー ... が発生しました ...	内部 ERX エラー。詳細については、EntireX Communicator のドキュメントを参照してください。 エラーに以下の部分が含まれている場合： メッセージ： ... プログラム = NATSRVD ... エラー = 00082 ... 呼び出されたプログラムは機能していません。	URL を修正してください。	プログラムを確認し格納してください。	インストールを確認してください。
NWW0012	ERX エラーレジスタ。	EntireX RPC サービスを初期化できません。	なし	なし	コンフィグレーションを確認してください。
NWW0013	erx.dll をロードできません。サブコード：...	EntireX Communicator erx.dll が見つかりません。	なし	なし	インストールを確認してください。
NWW0014	ERX ログオンに失敗しました。	EntireX Communicator にログオンできません。	ユーザー ID とパスワードを確認してください。	インストールファイルでユーザー ID とパスワードを確認してください。	インストールを確認してください。
NWW0015	ERX ログオフに失敗しました。	EntireX Communicator からのログオフに失敗しました。	なし	なし	Software AG にお問い合わせください。
NWW0033	ファイル ... が見つかりません (エラー：...)。	アダプタの初期化ファイルが見つかりません。	なし	なし	NSAPI の obj.conf 設定を確認してください。
NWW0034	NWW_USER_ID が長すぎます。	ユーザー ID に使用できる文字数は最大 8 文字です。	なし	なし	他のシステムで許可されていても、ユーザー ID は 8 文字以内で指定してください。
NWW0035	NWW_PASSWORD が長すぎます。	パスワードに使用できる文字数は最大 8 文字です。	なし	なし	他のシステムで許可されていても、(ユーザー) パスワードは 8 文字以内で指定してください。

エラー番号	エラーメッセージ	説明	ユーザー	プログラマ	管理者
NWW0036	Natural ライブラリ名が長すぎます。	Natural でライブラリ名に使用できる文字数は最大 8 文字です。	URL を確認してください。	URL 指定を確認してください。	なし
NWW0037	Natural サブプログラム名が長すぎます。	Natural サブプログラム名に使用できる文字数は最大 8 文字です。	URL を確認してください。	URL 指定を確認してください。	なし
NWW0099	CONTENT_TYPE : ... はサポートされていません。	CONTENT_TYPE = application/x-www-form-urlencoded のデータのみがサポートされています。	なし	FORM タグに属性 ENCTYPE を使用しないでください。	なし
NWW0100	RPC_INOUT_LENGTH が 30000 を超過しています。	HTTP サーバーに返される出力は、Natural RPC の制限によって限定されます。	なし	なし	コンフィグレーションを 変更してください。
NWW0101	パラメータの数が 200 を超過しています。	HTTP サーバーから渡される入力パラメータ数の上限は 200 です。	なし	Web インターフェイスに転送されるパラメータの数を減らしてください。	なし
NWW0200	ヘッダーが指定されていません。	各ページでは、戻りページにヘッダーセクションが必要です。	なし	各ページに CONTENT_TYPE が含まれている必要があります。ヘッダーセクションとデータは空行で区切る必要があります。	なし
NWW0201	ページにデータが含まれていません。	すべての戻りページにはデータが含まれている必要があります。	なし	プログラムを修正してください。	なし
NWW0815	インターフェイス A1(1:v) は現在サポートされていません。	間違ったインターフェイスが指定されています。	なし	なし	NWW_INOUT_FC および NWW_INOUT_LI パラメータを削除 してください。
NWW1001	クラスおよびメソッドが指定されていません。	指定した URL が正しくありません。クラスとメソッドの名前がありません。	URL を修正してください。	なし	なし

エラー番号	エラーメッセージ	説明	ユーザー	プログラマ	管理者
			ださい。		
NWW1002	クラスが指定されていません。	指定した URL が正しくありません。	URL を修正してください。	なし	なし
NWW1004	メソッドが指定されていません。	指定した URL が正しくありません。	URL を修正してください。	なし	なし
NWW1005	ASCII - Unicode 変換に失敗しました。	転送データを変換する必要があります。	なし	なし	Software AG にお問い合わせください。
NWW1006	Unicode - ASCII 変換に失敗しました。	転送データを変換する必要があります。	なし	なし	Software AG にお問い合わせください。
NWW1007	メソッド ... が見つかりません。	指定したメソッドを呼び出すことができません。	URL を修正してください。	クラスにメソッドを追加してください。	レジストリコンフィグレーションを確認してください。
NWW1008	クラス ... が見つかりません。	指定したクラスを呼び出すことができません。	URL を修正してください。	クラスを作成し、REGISTER * を使用して登録してください。	レジストリコンフィグレーションを確認してください。
NWW1009	クラス ... の初期化に失敗しました。	指定したクラスを呼び出すことができません。	URL を修正してください。	クラスを作成し、REGISTER * を使用して登録してください。	レジストリコンフィグレーションを確認してください。
NWW1010	DCOM 呼び出しに失敗しました。エラー ...。	DCOM への呼び出しに失敗しました。	なし	なし	コンフィグレーションを確認してください。
NWW1011	DCOM エラー ... が発生しました ...	内部 DCOM エラー。詳細については、DCOM のドキュメントを参照してください。	URL を修正してください。	プログラムを修正してください。	コンフィグレーション / インストールを修正してください。

エラー番号	エラーメッセージ	説明	ユーザー	プログラマ	管理者
NWW1012	DCOM の初期化に失敗しました。	DCOM への初期コールに失敗しました。	なし	なし	コンフィグレーションインストールを修正 さい。
NWW1013	DCOM のリリースに失敗しました。	クラスの削除と DCOM のクローズに失敗しました。	なし	なし	コンフィグレーションインストールを修正 さい。
NWW1036	DCOM クラス名が長すぎます。	Natural でライブラリ名に使用できる文字数は最大 32 文字です。	URL を確認 してく ださい。	URL 指定を確認し てください。	なし
NWW1037	DCOM メソッド名が長すぎます。	Natural でサブプログラム名に指定できる文字数は最大 32 文字です。	URL を確認 してく ださい。	URL 指定を確認し てください。	なし

13 プログラミングのヒント

▪ 小文字での編集	68
▪ 引用符と アポストロフィ	68
▪ 値で定義された変数	69
▪ リソースへのアクセス	69
▪ 定数値	69
▪ 新しいページの作成	70
▪ DCOM/RPC	70

このセクションでは、Natural Web インターフェイスの使用についていくつかのヒントを提供します。

このセクションでは、次のトピックについて説明します。

小文字での編集

メインフレームで Natural を使用している場合は、エディタで以下を設定できます。

エディタを小文字に設定します。

1. メニュー構造の **[Profile]** > **[Additional Options]** > **[General Defaults]** > **[Editing in Lower Case]** を順に選択します。
 2. **[Editing in Lower Case]** フィールドに「**Y**」を入力します。
- Natural Web サーバーエクステンションで提供されるすべてのプログラムでは、'（一重引用符）と"（二重引用符）のどちらで囲まれているかによって大文字に変換するかどうかが決まります。
 - '（一重引用符）で囲まれた文字列は大文字に変換されず、"（二重引用符）で囲まれた文字列は大文字に変換されます。

引用符と アポストロフィ

アプリケーションで引用符とアポストロフィの両方を使用するには、Natural パラメータ TQ（引用符の変換）を確認してください。このパラメータは、Natural テキスト定数内の引用符（"）の変換を制御します。コンパイル時にのみ影響があります。このパラメータを OFF にするか、W3-QUOTE-DQUOTE を使用してください。

パラメータ

```
1 W3QUOTE           (A001) /* o/ : Quote (")
1 W3APOSTROPHE      (A001) /* o/ : Apostrophe (')
```

呼び出し方法

```
PERFORM W3-QUOTE-DQUOTE W3QUOTE W3APOSTROPHE
```

値で定義された変数

すべての入力変数は、**BY VALUE** で定義されます。つまり、**MOVE** 互換のすべての値、特に定数文字列を使用できます。

リソースへのアクセス

すべてのリソース（画像、音声、Java アプレットなど）は、HTTP サーバーに保存されます。プログラムを作成して再配置する場合は、これらのリソースのパス名をハードコーディングしないでください。

環境変数を定義するときに、リソースの現在のパスを指定します。環境変数は、Natural Web サーバーエクステンションで設定できます。変数を設定しない場合は、デフォルト設定を使用します。

定数値

パラメータデータエリア W3CONST には、便利な定数値がいくつかあります。

##HTTP_NEWLINE

戻りページに書き込むと、文字列 ##HTTP_NEWLINE を文字列に圧縮することによって、実際に新しい行を作成できます。

##HTTP_NEWLINE_LENGTH

文字列 ##HTTP_NEWLINE の長さは、実装ごとに異なる場合があります。##HTTP_NEWLINE の長さが必要な場合は、##HTTP_NEWLINE_LENGTH を使用します。

新しいページの作成

出力が戻りページの上限を超えている可能性がある場合に、戻りページに何バイトの空きがあるかを評価するには、W3COUNTER を使用します。

DCOM/RPC

RPCとDCOMの両方で動作するアプリケーションを作成する場合は、以下の点を考慮する必要があります。

- Natural ライブラリとサブプログラムの名前記号制限を超えないでください。DCOM インターフェイスでは、クラスとそのメソッドの名前には最大32文字を使用できます（NaturalXのドキュメントを参照）。
- クラスと、すべてのサブプログラムが含まれているライブラリには、同じ名前を使用してください。これはオブジェクト指向の設計原理に従っていないかもしれませんが、これによりRPCまたはDCOM経由でサブプログラムにアクセスできる可能性があります。EntireX Communicatorでは、指定されたNaturalライブラリに対するダイナミックログオンがサポートされています。
- 現在、ライブラリはクラスと同等であり、そのライブラリに含まれているすべてのプログラムはこのクラスのメソッドです。RPCでの呼び出しができるようになりました。DCOMで呼び出すには、すべてのサブプログラムをクラスのメソッドとして指定するのみです。
- Natural Web インターフェイスでは、プログラム W3-R2DC (SYSWEB) を使用して、Natural ライブラリのクラスを生成できます。このプログラムは、すべてのサブプログラムでW3PARAMがパラメータデータエリアとして使用されているかどうかをチェックし、これらのサブプログラムをメソッドとして生成クラスに含めます。

14 Web インターフェイスの管理

- 戻りページトランスポートバッファサイズの設定 72
- ユーザー定義エラーページの作成 73
- ユーザー定義エラーページ XML スタイルの作成 73
- 英数字から HTML への変換 74
- 英数字から URL への変換 74

このセクションでは、次のトピックについて説明します。

戻りページトランスポートバッファサイズの設定

このセクションの内容は、SYSWEB にのみ適用されます。

トランスポート送信バッファの幅の変更

▶手順 14.1. トランスポート送信バッファの幅を変更するには、次の手順に従います。

- 1 パラメータデータエリア W3PARAM の変数 RETURN_PAGE の上限を変更します。

この値を、Natural Web サーバーエクステンションプログラム用の初期化ファイルのパラメータ NWW_INOUT_LENGTH と、ローカルデータエリア W3LIMITS の値 ##HTTP_RETURN_PAGE_PART の初期化に使用します。

これにより、トランスポートバッファの最大長が定義されます。

- 2 ライブラリ SYSWEB から W3* ソースをすべて再カタログします。
- 3 Natural Web サーバーエクステンションを使用して呼び出されるすべてのサブプログラムである NAT-*、HTTP*、および NAT-* プログラムをすべてライブラリ SYSWEB から再カタログします。

受信データバッファの幅の変更

▶手順 14.2. 受信データバッファの幅を変更するには、次の手順に従います。

- 1 ローカルデータエリア W3LIMITS の ##HTTP_ENVIRONMENT_MAX を初期化します。

これにより、受信データの最大長が定義されます。

この値は、トランスポートバッファの最大長（上記参照）以下にする必要があります。

- 2 ライブラリ SYSWEB から W3* ソースをすべて再カタログします。
- 3 Natural Web サーバーエクステンションを使用して呼び出されるすべてのサブプログラムである NAT-*、HTTP*、および NAT-* プログラムをすべてライブラリ SYSWEB から再カタログします。

戻りページの変更

▶手順 14.3. 戻りページを変更するには、次の手順に従います。

- 1 ローカルデータエリア W3LIMITS の ##HTTP_RETURN_PAGE_MAX を初期化します。
これにより、戻りページの最大長が定義されます。
- 2 ライブラリ SYSWEB から W3* ソースをすべて再カタログします。
- 3 Natural Web サーバーエクステンションを使用して呼び出されるすべてのサブプログラムである NAT-*、HTTP*、および NAT-* プログラムをすべてライブラリ SYSWEB から再カタログします。

ユーザー定義エラーページの作成

Natural でエラーが発生し、デフォルトの ON ERROR ブロックが指定されている場合は、W3ERROR が呼び出されて定義済みのエラーページが生成されます。

このエラーページを変更するには、サブルーチン W3ERROR-TEMPLATE (SYSWEB/W3ERRTMP) を変更します。

このプログラムは完全な HTML ページを生成します。

ユーザー定義エラーページ XML スタイルの作成

Natural でエラーが発生し、デフォルトの ON ERROR ブロックが指定されている場合は、W3ERROR が呼び出されて定義済みのエラーページが生成されます。

このエラーページを XML 対応 HTML に変更するには、次の手順に従います。

- 1 サブルーチン (SYSWEB/W3ERRTMP) をアンカタログします。
- 2 サブルーチン (SYSWEB/W3ERXTMP) を開きます。
- 3 W3ERROR-TEMPLATE-XML を W3ERROR-TEMPLATE に名前変更します。
- 4 プログラムを格納します。

これで、このプログラムは完全な XML 対応 HTML ページを生成します。

英数字から HTML への変換

HTML に変換するために、特殊文字を適切な HTML 表現に置換する必要があります。

- サブルーチン W3-ASCII-HTML-TABLE (SYSWEBP/W3AS2HT) には、文字置換のための設定が含まれています。
- W3INIT と W3-TEXT-TO-HTML は、W3-ASCII-HTML-TABLE を呼び出します。

最大 128 件の置換を保存できます。

定義に 16 進数値が使用されている場合 (例: 引用符) は、ASCII 文字セットの値と EBCDIC 文字セットの値を定義する必要があります。定義しないと、ファイルを移植できません。

英数字から URL への変換

URL デコードのために、特殊文字を適切な URL 対応表現に置換する必要があります。

- サブルーチン H3-ASCII-URL-TABLE (SYSWEB/H3AS3URL) には、文字置換のための設定が含まれています。
- H3-ASCII-URL-TABLE は、H3-TEXT-TO-URL によって呼び出されます。

最大 128 件の置換を保存できます。

定義に 16 進数値が使用されている場合 (例: 引用符) は、ASCII 文字セットの値と EBCDIC 文字セットの値を定義する必要があります。定義しないと、ファイルを移植できません。

15 デモアプリケーション - JavaScript なし

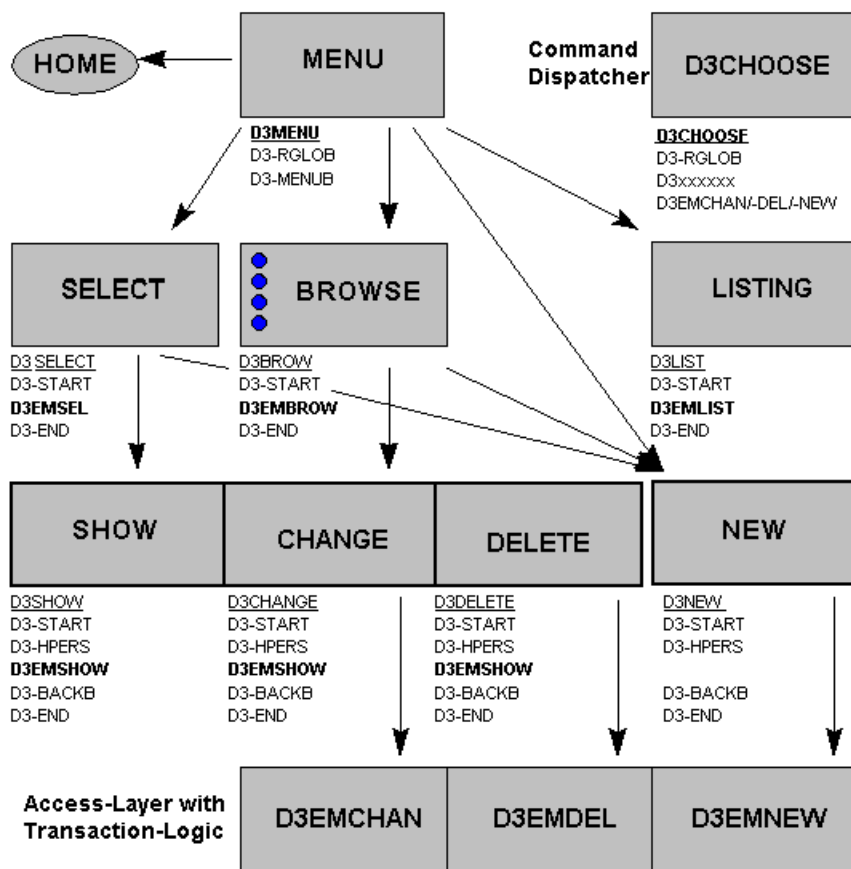
■ ビジネス要件	76
■ 設計に関する決定	77
■ ライブラリ、モジュール、および命名規則	77
■ デモアプリケーションの開始	78
■ Natural Web インターフェイスオンラインマニュアルの開始	78
■ 要件	78

このセクションでは、次のトピックについて説明します。

ビジネス要件

デモアプリケーションでは、NaturalWeb インターフェイスの使用とプログラミングについて示します。このアプリケーションには、以下の図に示すような、さまざまな選択機能を備えたシンプルなファイルメンテナンス機能があります。

このデモは、プラットフォームに依存せず、Adabas ファイル EMPLOYEES および VEHICLES を基盤としています。



Legend: Module HTML-Form
Module: NATURAL object type subprogram
 → Call of dispatcher module D3CHOOSE

設計に関する決定

HTML-GUI には、アプリケーション設計に対して以下の制限事項があります。

- ブラウザごとに固有のレイアウトを使用することはできません。
- HTML-GUI 要素の機能は限定されています。例えば、選択ボックスの入力がない、事前定義されたフォントおよびボタンのみをサブミットに使用できる（デフォルトボタンがない）などの制限があります。

このため、デモアプリケーションは以下のようにします。

- サブミットボタン付きのフォームを使用します。
- フォーム上の非表示フィールドとのグローバルデータ交換を実行します。
- フォーム取得メソッド GET (URL およびブックマークの表示可能パラメータ) を使用します。
- 処理ルールの実装に VB/JavaScript を使用しません。
- ナビゲーションのためのコマンドディスパッチャモジュール (D3CHOOSE) を使用します。
- 著作権上の理由により、グループ／男性／女性の標準画像を使用します。

ライブラリ、モジュール、および命名規則

デモにはモジュールが 1 つ含まれています ([Natural Web](#) サーバーエクステンションのインストールも参照)。

SYSWEB

このライブラリには次のモジュールが含まれています。

T3 オンラインドキュメント用の HTML テキスト

E3 オンラインドキュメントの例

D3 デモアプリケーションモジュール

デモアプリケーションの開始

デモンストレーションの開始モジュールは D3MENU です。

デモアプリケーションを開始するには、Natural Web サーバーエクステンションのインストールに応じて、ライブラリ SYSWEB のサブプログラム D3MENU を呼び出します。

SYSWEB でデモアプリケーションを呼び出す URL の例：

`http://yourserver/yourcgi/sysweb/d3menu`

Natural Web インターフェイスオンラインマニュアルの開始

Natural Web インターフェイスからオンラインドキュメントを開始できます。

デモンストレーションの開始モジュールは D3MENU です。

オンラインマニュアルを開始するには、ライブラリ SYSWEB のサブプログラム D3MENU を呼び出します。

SYSWEB でデモアプリケーションを呼び出す URL の例：

`http://yourserver/yourcgi/sysweb/d3menu`

要件

次のソフトウェアをインストールする必要があります。

- Natural Web サーバーエクステンション (Natural Web インターフェイスの一部)
- Adabas およびファイル EMPLOYEES

デモアプリケーションを有効にするには、ライブラリ SYSWEB のプログラム D3* に対して CATALL を実行します。

Natural Web サーバーエクステンションによって提供される例の画像を表示するには、HTTP サーバーのディレクトリ pictures にすべての画像をコピーするか、Natural Web サーバーエクステンションの環境変数 PICTURES を特定のディレクトリに設定します。

16 デモアプリケーション - JavaScript 使用

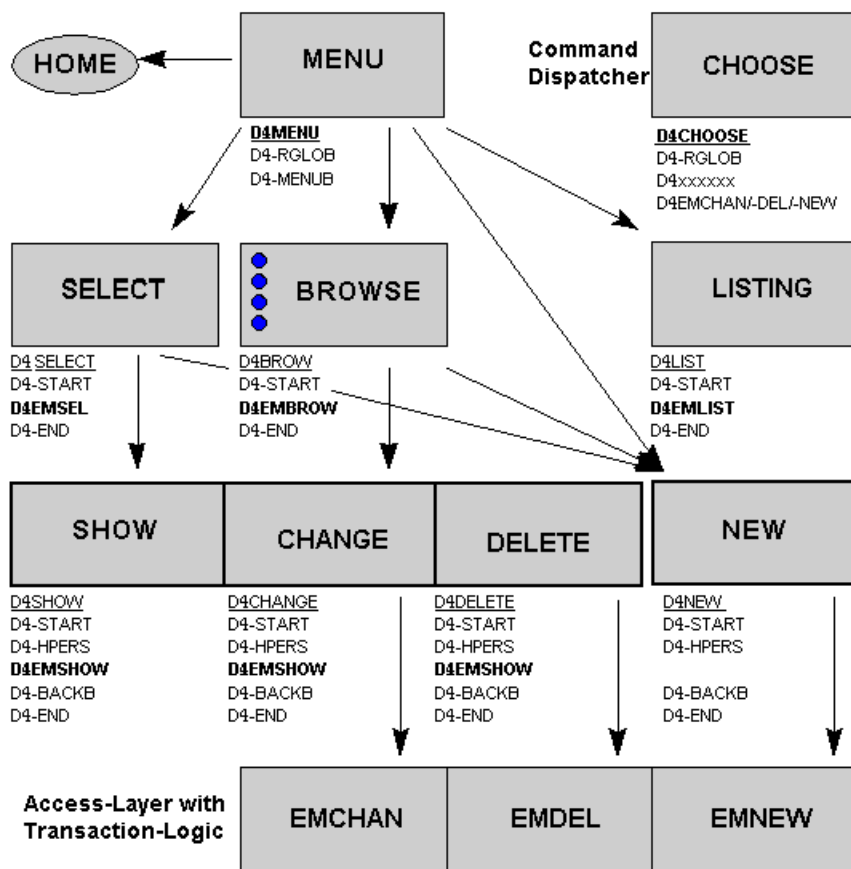
■ ビジネス要件	80
■ 設計に関する決定	81
■ デモアプリケーションの開始	81
■ 要件	81

このセクションでは、次のトピックについて説明します。

ビジネス要件

デモアプリケーションでは、Natural Web インターフェイスの使用とプログラミングを示します。このアプリケーションには、以下の図に示すような、さまざまな選択機能を備えたシンプルなファイルメンテナンス機能があります。

クロスプラットフォーム可用性のために、このデモは Adabas ファイル EMPLOYEES および VEHICLES を基盤としています。



Legend: Module HTML-Form
Module: NATURAL object type subprogram
 → Call of dispatcher module D4CHOOSE

設計に関する決定

最新の Web 設計を使用します。

- Javascript。
- フォーム上の非表示フィールドとのグローバルデータ交換を実行します。
- フォーム取得メソッド GET (URL およびブックマークの表示可能パラメータ) を使用します。
- ナビゲーションのためのコマンドディスパッチャモジュール (D4CHOOSE)

デモアプリケーションの開始

デモの開始モジュールは D4ENTER です。Natural Web サーバーエクステンションのインストールに応じて、ライブラリ SYSWEB のサブプログラム D4ENTER を呼び出します。

デモアプリケーションを呼び出す URL の例：

`http://yourserver/yourcgi`

要件

Natural Web サーバーエクステンション (Natural Web インターフェイスの一部) と、Adabas およびファイル Employee がインストールされている必要があります。デモアプリケーションを有効にするには、ライブラリ SYSWEB のプログラム D4* に対して CATAL を実行します。

例に画像を表示するには、Natural Web サーバーエクステンションのデモ部分を HTTP サーバーの root にインストールする必要があります。

17 Natural Web インターフェイスのエラーメッセージ

- エラーメッセージ 84

このセクションでは、Natural Web インターフェイスで作業しているときに表示される可能性のあるエラーメッセージについて説明します。エラーの内容と対処法を示します。

エラーメッセージ

エラー番号	エラーメッセージ	説明	対処
NWW9002	エレメントが定義されていません。	配列数の値が0に設定されています。	プログラムを修正してください。
NWW9003	FORM タグ内でのみ使用できます。	このタグは FORM タグ内でのみ使用できます。	FORM を H3-OPEN-FORM で初期化してください。
NWW9004	FORM タグには ACTION を指定する必要があります。	各 FORM には ACTION を指定する必要があります。	プログラムを修正してください。
NWW9005	リストの外に LI タグを指定することはできません。	LI はリスト内に置く必要があります。	FORM を H3-OPEN-LIST で初期化してください。
NWW9006	リストのネストが深すぎます：...	リストはレベルの数は 10 までのみサポートされています。	レベルを減らしてください。
NWW9007	ラジオボタングループに名前がありません。	ラジオボタングループを生成するには、名前が必要です。	名前を追加してください。
NWW9008	エレメント ... に名前がありません。	チェックボックスグループの各要素には名前が必要です。	名前を追加してください。
NWW9009	テキストエリアに名前がありません。	テキストエリアを生成するには名前が必要です。	名前を追加してください。

18 Natural Web オンラインドキュメント SYSWEB

■ 全般的な情報	86
■ 基本モジュール	86
■ 出力の後処理	88
■ HTML エクステンション	88
■ ユーティリティ	90
■ デモアプリケーション	90


このセクションでは、次のトピックについて説明します。

全般的な情報

オンラインドキュメントファイルの接頭辞は E3* と T3* です。オンラインドキュメントには、オンラインで表示および実行できるプログラム例が含まれています。Natural Web インターフェイスのインストールに応じて、ライブラリ SYSWEB からサブプログラム NAT-DOCU を呼び出して、Web ブラウザでオンラインドキュメントのメインページを表示します。

オンラインドキュメントを呼び出す URL の例

`http://yourserver/yourcgi/sysweb/nat-docu`

 **注意:** オンラインドキュメントを表示するには、Natural Web インターフェイスの HTTP サーバーエクステンションがインストールされており、適切な Natural RPC/DCOM サーバーが起動されている必要があります。ライブラリ SYSEXT のプログラム USR1057N にアクセスするには、steplib を SYSEXT に追加するか、またはプログラムをシステムライブラリにコピーします。


パラメータの定義	
i/	入力変数
o/	出力変数
/o	オプション変数
/m	必須変数（指定する必要があります）
/M	必須変数。指定されていない場合は、特定の部分が生成されません。
/H	変数は HTML に変換されます。
/X	変数は XML に変換されます。
/U	変数は URL に変換されます。

基本モジュール

Natural Web インターフェイスの基本モジュール名は、接頭辞 W3 で始まります。

基本モジュールによって、Natural サブプログラムと HTTP サーバーエクステンションの通信が可能になります。Natural Web インターフェイスの他のすべてのプログラムでは、これらのプログラムを使用します。

転送されるデータの量とフォーマットを定義したり、変換テーブルを変更したり、エラーページを変更したりするために、管理用の変更を加えることができます。

 **注意:** 以下の表では、使用可能な新しいプログラムはすべて *italics* フォントで示されています。カッコ内のプログラム名は、近い将来に廃止されます。説明に記述されているプログラム名を使用してください。

プログラム	説明
W3CLEAR	出力ページをクリアします。
W3CONTENT-TYPE	ドキュメントのコンテンツタイプを設定します。
W3COUNTER	出力エリアの最大バイト数、および現在書き込まれているバイト数を返します。
W3ERROR W3ERROR-TEMPLATE W3ERROR-TEMPLATE-XML W3ERROR-TEXT	デフォルトのエラーページを生成します。
W3HTML W3HTMLLINE <i>W3HTMLDYNAMIC</i> <i>W3HTMLLINEDYNAMIC</i> W3HTMLARRAY	出力ページに HTML 文字列 を書き込み、特殊文字を HTML で有効な表現に変換します。
W3HTTP <i>W3HTTPDYNAMIC</i> W3HTTP-HEADER W3HTTARRAY	HTTP 設定 を出力ページに書き込みます。
W3INFO	内部設定を返します。
W3INIT W3END	SYSWEB を初期化し、HTTP サーバーに返されるドキュメントを準備します。
W3LIST-ENVIRONMENT W3LIST-ENVIRONMENT-TO-DYNAMIC	すべての変数をリストします。
W3LOCATION	このページの代わりに呼び出されるページの場所を設定します。
W3READ-ENVIRONMENT W3READ-ENVIRONMENT-ARRAY <i>W3READ-ENVIRONMENT-TO-DYNAMIC</i>	HTTP サーバーによって送信される変数を読み取ります。
W3READ-ENVIRONMENT-TEXTAREA	テキストエリアによって設定される変数を読み取り、その変数を個別の行に分割します。
W3READ-ENVIRONMENT-GROUP	同じ名前を持つ環境変数をすべて読み取ります。
W3TEXT W3TEXTLINE <i>W3TEXTDYNAMIC</i> <i>W3TEXTLINEDYNAMIC</i> W3TEXTARRAY	テキスト文字列を出力ページに書き込みます。
W3NEWLINE	改行を出力ページに書き込みます。
(W3SPACE)	W3TEXTDYNAMIC " " に置き換えます。
W3-QUOTE-DQUOTE	文字セット非依存の特殊文字を返します。

プログラム	説明
W3TEXT-T0-HTML W3-ASCII-HTML-TABLE	ASCII を HTML の特定のエンコードに変換します。
W3TEXT-T0-XML W3-ASCII-XML-TABLE	ASCII を XML の特定のエンコードに変換します。
W3TEXT-T0-URL W3-ASCII-URL-TABLE	ASCII を URL の特定のエンコードに変換します。

出力の後処理

プログラム	説明
W3REPLACE	出力ページで特定の文字列を検索し、新しい文字列に置き換えます。
W3READ-OUTPUT	すでに書き込まれている出力ページを読み取ります。

HTML エクステンション

HTML エクステンションのすべてのプログラム名に、接頭辞 H3 が使用されます。この外部サブルーチン（ソースコードが付属しています）では HTML を生成し、Natural Web インターフェイスの基本モジュールを使用します。

このプログラムは、HTML の構文に完全に対応しているわけではありません。また、Web ブラウザによっては、一部の拡張機能がサポートされていない場合があります。拡張機能が必要な場合は、付属のプログラムのソースコードを拡張するか、または独自のプログラムを作成します。

プログラム	説明	HTML タグ
H3-ANCHOR	アンカータグを作成します。	<A...>...
H3-BUTTON	リセット／実行ボタンを作成します。	<INPUT...>
H3-CHECKBOX-GROUP	チェックボックスグループを生成します。	<INPUT...>
H3-COMMENT	コメント行を作成します。	<!...>
H3-HEADER	ヘッダータグを生成します。	<Hn>
H3-IMAGE	イメージタグを生成します。	<IMG...>
H3-INPUT	テキスト、パスワードまたは非表示の入力フィールドを生成します。	<INPUT...>
H3-LINE-BREAK H3-LINE_BREAK	改行を設定します。場合によっては後ろにテキストが追加されます。	

プログラム	説明	HTML タグ
H3-OPEN-FORM H3-CLOSE-FORM	入力フィールドの フォーム タグを開始します。	<FORM>...</FORM>
H3-OPEN-HTML H3-OPEN-HTML-JAVASCRIPT H3-CLOSE-HTML	HTML ドキュメントを介し及び終了します。	<HTML>...</HTML>
H3-OPEN-LIST H3-LIST-ITEM H3-CLOSE-LIST	整列された、または整列されていない、メニューあるいはディレクトリ リスト を生成します。 <DIR>......</DIR> <MENU>......</MENU>
H3-PARAGRAPH	テキストが追加された パラグラフ を生成します。	<P...>
H3-RADIO-GROUP	ラジオボタン グループを生成します。	<INPUT...>
H3-RULE	横罫線 を設定します。	<HR...>
H3-SCROLLING-LIST	スクロール リストを生成します。	<SELECT> ...<OPTION>... <SELECT>
H3-TABLE H3-TABLE-COLOR	テーブル を生成します。	<TABLE > ... <TR> <TH>...</TH> </TR> <TR> <TD...>...</TD> </TR> ... </TABLE>
H3-TAG	汎用 タグ を生成します。	<tag>
H3-TEXT-AREA	' テキストエリア 'を生成します。	<TEXTAREA>...</TEXTAREA>
H3-TEXT-TO-HTML	Natural 文字列の内容を ' HTML ' に変換します。 W3TEXT-TO-HTML に置き換えます。	
H3-TEXT-TO-URL H3-ASCII-URL-TABLE	Natural 文字列の内容を ' URL デコード形式 ' に変換します。 W3TEXT-TO-URL に置き換えます。	
H3-TIME_DATE H3-TIME-DATE	' 時刻／日付 ' 文字列を生成します。	generated: Mon, 17 Jan 2005 15:35:18 GMT

ユーティリティ

オンライン

プログラム	説明
WEB-WIZ	NATURAL Web インターフェイスによって呼び出されるサブプログラムのスケルトンを生成します。
WEB-ONL	Natural Web インターフェイスのサブプログラムをオンラインで実行します。

リモート

プログラム	説明
NAT-LIB	すべての Natural ライブラリをリストします。
NAT-DIR	特定の Natural ライブラリの内容をリストします。
NAT-ENV	呼び出された Natural サブプログラムに渡されるすべてのパラメータをリストします。
NAT-HTML	HTML を含む Natural ソースを表示します。
NAT-INFO	Natural Web インターフェイスの現在の設定を表示します。
NAT-LIST	Natural ソースオブジェクトを表示します。
NAT-DOCU	オンラインドキュメントを表示します。

デモアプリケーション

付属のデモアプリケーションでは、選択機能を使用した簡単なファイルメンテナンスを行います。デモンストレーションは、Adabas のファイル EMPLOYEES に基づいています。このアプリケーションを実行するには、Adabas がアクティブである必要があります。active。

デモアプリケーションは 2 つの実装が提供されています。

1. JavaScript を使用する実装。名前の接頭辞は D4*。
HTTP サーバーエクステンションのインストールに応じて、ライブラリ SYSWEB からサブプログラム D4ENTER を呼び出します。

デモアプリケーションを呼び出す URL の例

`http://yourserver/yourcgi/sysweb/d4enter`

2. 標準 HTML 3.2 を使用する実装。名前の接頭辞は D3*。
HTTP サーバーエクステンションのインストールに応じて、ライブラリ SYSWEB からサブプログラム D3MENU を呼び出します。

デモアプリケーションを呼び出す URL の例

`http://yourserver/yourcgi/sysweb/d3menu`

使用されるすべての画像は、Natural Web インターフェイスに付属しています。これらの画像は、HTTP サーバーのリモート ディレクトリ PICTURES の pictures ディレクトリに保存してください。別のリモート ディレクトリ名を使用する場合は、HTTP サーバーエクステンションの初期化ファイルにある環境変数 PICTURES に、該当のリモートディレクトリ名を設定します。

D4* の例のための JavaScript ファイルは、Natural Web インターフェイスに付属しています。これらのファイルは、HTTPサーバーのリモートディレクトリ /javascript の javascript ディレクトリに保存してください。別のリモートディレクトリ名を使用する場合は、HTTPサーバーエクステンションの初期化ファイルにある環境変数 JAVASCRIPT に、該当のリモートディレクトリ名を設定します。

19 出力エリアのクリア

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3CLEAR	E3CLEAR	E3CLEAR

説明

出力ファイルにすで書き込まれているデータをすべて削除します。

パラメータ

```
*/ NONE
```

呼び出し方法

```
PERFORM W3CLEAR
```


20 ドキュメントのコンテンツタイプの設定

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3CONTENT-TYPE	E3CONTYP	E3CONTYP

説明

ドキュメントのコンテンツタイプを設定します。この設定は、コンテンツをどのように表示するかを判断するために、ブラウザプログラムによって使用されます。

ドキュメントの最初の出力は、W3CONTENT-TYPE または W3LOCATION である必要があります。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

パラメータ

```
1 H3VALUE (A) DYNAMIC /* i /m : Content type to be set
```

呼び出し方法

```
PERFORM W3CONTENT-TYPE H3VALUE
```


21 出力エリアのサイズのカウンタ

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3COUNT	E3COUNT	E3COUNT

説明

出力エリアの現在のサイズとすでに書き込まれたバイト数を返します。

パラメータ

```
1 W3WRITTEN (I4) /* o/m : Currently written bytes
1 W3MAXPAGE (I4) /* o/m : Maximum bytes possible
1 W3FREE    (I4) /* o/m : Free bytes
```

呼び出し方法

```
PERFORM W3COUNTER W3WRITTEN W3MAXPAGE W3FREE
```


22 エラーページの生成

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3ERROR W3ERROR-TEMPLATE W3ERROR-TEMPLATE-XML W3ERROR-TEXT	E3ERROR	E3ERROR

説明

Natural ランタイムによって生成されたエラーは、画面に出力されないように処理する必要があります。このため、Natural Web インターフェイスを使用して呼び出されるすべてのプログラムには ON ERROR セクションを追加する必要があります。また、PDA W3CONST を追加する必要もあります。

エラーが発生すると、サブルーチン W3ERROR-TEMPLATE が呼び出されます。このルーチンは必要に応じて変更できます。

サブルーチン W3ERROR-TEMPLATE-XML では、エラーページを XHTML ページとして返します。このルーチンは必要に応じて変更できます。このテンプレートを有効にするには、W3ERROR-TEMPLATE をアンカタログし、このサブルーチンの名前を W3ERROR-TEMPLATE-XML から W3ERROR-TEMPLATE に変更して格納します。

サブルーチン W3ERROR-TEXT は内部使用専用です。

パラメータ

```
1 ##W3ERROR
2 NR          (I4) /* i /m : Number of the error
2 LINE        (I4) /* i /m : Line in the Natural program
2 SUBPROGRAM (A008) /* i /m : Subprogram name
2 SUBROUTINE (A032) /* i /m : Subroutine name
2 TEXT        (A250) /* i /m : Error text
```

呼び出し方法

```
ON ERROR  
  PERFORM W3ERROR ##W3ERROR  
  PERFORM W3END ##RPC  
  ESCAPE ROUTINE  
END-ERROR
```

23 ドキュメントへの書き込みおよび有効な HTML への変換

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3HTML W3HTMLDYNAMIC W3HTMLLINE W3HTMLINEDYNAMIC W3HTMLARRAY	E3HTMLA	E3HTMLA

説明

文字列をドキュメントに書き込み、"<"、">"、"Ã¼"などの特殊文字を変換します。

出力の後に改行を作成する場合は、W3HTMLLINE または W3HTMLLINEDYNAMIC を使用します。
文字列の内部に改行を作成する場合は、##HTTP-NEWLINE を文字列内に挿入します。

W3HTML および W3HTMLLINE を文字列に適用すると、文字列の末尾の空白が削除されます。

パフォーマンスを向上させるには、ダイナミック変数を使用します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

パラメータ

1. W3HTML

```
1 H3HTML (A) DYNAMIC BY VALUE /* i /mH: Output string
```

2. W3HTMLDYNAMIC

```
1 H3DYNAMIC (A) DYNAMIC BY VALUE /* i /mH: Output string
```

3. W3HTMLLINE

```
1 H3HTML (A) DYNAMIC BY VALUE /* i /mH: Output string
```

4. W3HTMLLINEDYNAMIC

```
1 H3DYNAMIC (A) DYNAMIC BY /* i /mH: Output string
```

5. W3HTMLARRAY

```
1 H3ARRAYVALUE (A/1:v) DYNAMIC /* i /mH: Output array  
1 H3VALUELENGTH (I4) /* i /m : Length of output array
```

呼び出し方法

```
PERFORM W3HTML H3HTML  
PERFORM W3HTMLDYNAMIC H3DYNAMIC  
PERFORM W3HTMLLINE H3HTML PERFORM W3HTMLLINEDYNAMIC H3DYNAMIC  
PERFORM W3HTMLARRAY H3ARRAYVALUE H3VALUELENGTH
```

24 ドキュメントへの HTTP 設定の書き込み

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3HTTP W3HTTPDYNAMIC W3HTTP-HEADER W3HTTPARRAY	E3HTTP	E3HTTP

説明

テキスト行をドキュメントのヘッダーに書き込みます。これらのテキスト行設定には、HTTP と互換性のあるドキュメントに対するクッキー、有効期限などの設定を指定できます。

Natural 文字列に `##HTTP_NEWLINE` を挿入することによって、出力内に物理改行を作成できます。

文字列の内部に改行を作成する場合は、`##HTTP-NEWLINE` を文字列内に挿入します。

W3HTTP を文字列に適用すると、文字列の末尾の空白が削除されます。

パフォーマンスを向上させるには、ダイナミック変数を使用します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

パラメータ

1. W3HTTP

```
1 W3STRING      (A) DYNAMIC    /* i /m : Header name value pairs
```

2. W3HTTPDYNAMIC

```
1 W3DYNAMIC     (A) DYNAMIC    /* i /m : Header name value pairs
```

3. W3HTTP-HEADER

```
1 W3HTTPNAME    (A) DYNAMIC    /* i /m : Header name
1 W3HTTPVALUE   (A) DYNAMIC    /* i /m : Header value
```

4. W3HTTPARRAY

```
1 W3ARRAYVALUE  (A/1:V) DYNAMIC /* i /m : Header name value pairs
1 W3VALUELENGTH (I4)          /* i /m : Length of output array
```

呼び出し方法

```
PERFORM W3HTTP W3STRING
PERFORM W3HTTPDYNAMIC W3DYNAMIC
PERFORM W3HTTP-HEADER W3HTTPNAME W3HTTPVALUE
PERFORM W3HTTPARRAY W3ARRAYVALUE W3VALUELENGTH
```

25 内部値に関する情報

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3INFO	E3INFO	E3INFO

説明

このプログラムを使用すると、Web インターフェイスの一部の情報を設定および書き込むことができます。

読み取り (アクション 'I')

ERROR-NR VERSION、LOG-TIME、FORM、LIST(*) および LIST_MAX の値が返されます。

リストの設定 (アクション 'L')

H3-OPEN-LIST、H3-CLOSE-LIST および H3-LIST-ITEM では、生成されたリストのスタイルを保存するために内部配列が使用されます。このスタイルは、適切な終了タグを生成するために使用されます。

フォームの設定 (アクション 'F')

すべてのプログラムについて、FORM タグ内でのみ使用できるタグを生成できます。フォームが開いているかどうかを確認するために、フラグを呼び出すことができます。このフラグは H3-OPEN-FORM and H3-CLOSE-FORM によって変更できます。

パラメータ

```
LOCAL USING WPINFO
```

PDA W3PINFO

```
1 ##W3INFO
2 ACTION (A1) /* i /m : Action to be called
2 LOG-TIME (A030) /* o/m : Log time set by the Natural Web Interface
2 VERSION (A015) /* o/m : Version set by the HTTP Server Extension
2 WEBAPI (A015) /* o/m : Version set by the Natural Web Interface
2 ERROR-NR (I4) /* o/m : Error number set by the Natural Web Interface
2 FORM ( L) /* io/m : Indicates whether a FORM is open
2 LIST (A001/1:10) /* io/m : Saves the type of LIST
2 LIST_MAX (N002) /* io/m : Current number of nested LISTS
```

呼び出し方法

```
PERFORM W3INFO ##W3INFO
```


26 ドキュメントの終了および初期化

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3INIT W3END	E3END	E3END

説明

各プログラムでは、特別なプログラムを使用して Web インターフェイス を初期化および終了する必要があります。初期化するには、W3INIT を使用します。

渡されたパラメータを今後の使用に備えて初期化するために、W3PARMPDA を渡す必要があります。

W3END では、ドキュメントを終了し、HTTP サーバーに戻る準備をします。初期プログラムで定義された W3PARM PDA は W3END に渡す必要があります。

W3END の後に W3* 呼び出しが実行された場合、書き込まれた出力は HTTP サーバーに転送されません。

パラメータ

1. W3INIT

```
USING W3PARM /* io/m : Parameter of Subprogram
```

2. W3END

```
USING W3PARAM /* io/m : Parameter of Subprogram
```

3. W3PARAM

```
1 ##RPC
2 VERSION          (A010)      /* i /m : Interface version
2 LOG-TIME         (A030)      /* i /m : Timestamp
2 RETURN_PAGE     (A250/1:V) /* io/m : Transfer area
2 RETURN_PAGE_COUNT (I004)     /* io/m : Bytes sent
2 ERROR-NR        (I004)     /* o/m : Error number
```

呼び出し方法

```
PERFORM W3INIT ##RPC
PERFORM W3END ##RPC
```

27 すべての環境変数のリスト

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3LIST-ENVIRONMENT W3LIST-ENVIRONMENT-TO-DYNAMIC	E3ENVLIS	E3ENVLIS

説明

HTTP サーバーによって送信されたすべての変数をリストします。

パラメータ

1. W3LIST-ENVIRONMENT

```
1 W3START          (I4)      /* io/m : Offset to be started at
*                  /*          out: 0 no occurrences
*
*                  /*          out: >0 more occurrences
1 W3ARRAYCOUNTER   (A72/1:V) /* io/m : Length of array returned values
1 W3ARRAYNAME      (A250/1:V) /* o/m : Name of variables
1 W3ARRAYVALUE     (I4/1:V)  /* o/m : Value of variables
1 W3ARRAYVALUELENGTH (L/1:V) /* o/m : Length of variables
1 W3ARRAYVALUESERVER (I4)    /* o/m : Variable belongs to
1 W3ARRAYMAXIMUM   /* o/m : Total number of variables
```

2. W3LIST-ENVIRONMENT-TO-DYNAMIC

```
1 W3START          (I4)      /* io/m : Offset to be started at
*                  /*          out: 0 no occurrences
*                  /*          out: >0 more occurrences
1 W3ARRAYCOUNTER   (A/1:V)dynamic /* io/m : Length of array returned values
1 W3ARRAYNAME      (A/1:V)dynamic /* o/m : Name of variables
1 W3ARRAYVALUE     (I4/1:V)    /* o/m : Value of variables
1 W3ARRAYVALUELENGTH (L/1:V)   /* o/m : Length of variables
```

すべての環境変数のリスト

```
1 W3ARRAYVALUESERVER (I4)      /* o/m : Variable belongs to
1 W3ARRAYMAXIMUM               /* o/m : Total number of variables
```

呼び出し方法

```
PERFORM W3LIST-ENVIRONMENT W3START W3NAME W3ARRAYCOUNTER
        W3ARRAYNAME(*) W3ARRAYVALUE(*) W3ARRAYVALUELENGTH(*)
        W3ARRAYVALUESERVER(*) W3ARRAYMAXIMUM
```

28 ドキュメントの場所の設定

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3LOCATION	E3LOCAT	E3LOCAT

説明

ロードされるドキュメントの場所を設定します。このサブルーチンを使用すると、ダイナミックページではなくスタティックページを Natural プログラムから呼び出すことができます。

ドキュメントの最初の出力は、W3LOCATION または **W3CONTENT-TYPE** である必要があります。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

パラメータ

```
1 W3CONTENT (A) DYNAMIC /* i /m : Content type to be set
```

呼び出し方法

```
PERFORM W3CONTENT-TYPE W3CONTENT
```


29 環境変数の読み取り

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3READ-ENVIRONMENT-ARRAY W3READ-ENVIRONMENT W3READ-ENVIRONMENT-TO-DYNAMIC	E3ENVARY	E3ENVARY

説明

個別の変数について、最初に出現する 1 件を読み取ります。W3READ-ENVIRONMENT-ARRAY を使用すると、オフセットから始まる変数を読み取ることができます。これにより、複数回出現する同一の変数を読み取ることができます。

パラメータ

1. W3READ-ENVIRONMENT-ARRAY

```
1 W3START      (I4)      /* io/m : Offset to be started at
*                /*                out: 0 no occurrences
*                /*                out: >0 more occurrences
1 W3NAME       (A072)   /* i /m : Name of the variable to
*                /*                be searched for
1 W3VALUESERVER (A1)    /* i /m : Search for variables from
*                /*                'S' server
*                /*                'P' page or URL
*                /*                ' ' both
1 W3ARRAYCOUNTER (I4)   /* io/m : Size of array,
*                /*                characters read
1 W3VALUEARRAY (A001/1:V) /* o/m : Array with the returned value
1 W3VALUELENGTH (I4)   /* o/m : length of the value
```

2. W3READ-ENVIRONMENT

```
1 W3NAME          (A072) /* i /m : Name of the variable
                        /*          searched for
1 W3VALUESERVER  (A1)   /* i /m : Search for variables in
*                  /*          'S' server
*                  /*          'P' page or URL
*                  /*          ' ' both
1 W3VALUE        (A250) /* o/m : Returned value
1 W3VALUELENGTH (I4)   /* o/m : Length of the value
```

3. W3READ-ENVIRONMENT-TO-DYNAMIC

```
1 W3NAME          (A072) /* i /m : Name of the variable
                        /*          searched for
1 W3VALUESERVER  (A1)   /* i /m : Search for variables in
*                  /*          'S' server
*                  /*          'P' page or URL
*                  /*          ' ' both
1 W3VALUEDYNAMIC (A) DYNAMIC /* o/m : Returned value
```

呼び出し方法

```
PERFORM W3READ-ENVIRONMENT-ARRAY W3START W3NAME
        W3VALUESERVER W3ARRAYCOUNTER
        W3VALUEARRAY(*) W3VALUELENGTH

PERFORM W3READ-ENVIRONMENT W3NAME W3VALUESERVER W3VALUE
        W3VALUELENGTH

PERFORM W3READ-ENVIRONMENT-TO-DYNAMIC W3NAME W3VALUESERVER
        W3VALUEDYNAMIC
```


30 環境変数グループの読み取り

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3READ-ENVIRONMENT-GROUP	E3ENVGRO?test=a&test=bb&test=cc	E3ENVGRO

説明

複数の選択から設定するなどして、同じ名前を持つ変数をすべて読み取ります。

パラメータ

```
1 W3START          (I4)          /* io/m : Offset to be started at
*                               /*          out: 0 no occurrences
*                               /*          out: >0 more occurrences
1 W3NAME           (A) DYNAMIC /* i /m : Name of variable
1 W3VALUESERVER   (A1)         /* i /m : Search for variables in
*                               /*          'S' server
*                               /*          'P' page or URL
*                               /*          ' ' both
1 W3ARRAYCOUNTER  (I4)         /* io/m : Length of array,
*                               /*          returned values
1 W3ARRAYVALUES   (A250/1:V) /* o/m : Values of variable
1 W3ARRAYMAXIMUM  (I4)         /* o/m : Total number of variables
```

呼び出し方法

```
PERFORM W3READ-ENVIRONMENT-GROUP W3START W3NAME
        W3VALUESERVER W3ARRAYCOUNTER
        W3ARRAYVALUES(*) W3ARRAYMAXIMUM
```


31 環境テキストエリア変数の読み取り

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3READ-ENVIRONMENT-TEXTAREA	E3ENVTX	E3ENVTX E3ENVTX1

説明

テキストエリアタグによって設定された変数を読み取り、テキスト行を分割します。

パラメータ

```
1 W3START          (I4)      /* io/m : Offset to be started at
*                  /*          out: 0 no occurrences
*                  /*          out: >0 more occurrences
1 W3NAME           (A072)   /* i /m : Name of variable
1 W3VALUESERVER    (A1)     /* i /m : Search for variables in
*                  /*          'S' server
*                  /*          'P' page or URL
*                  /*          ' ' both
1 W3ARRAYCOUNTER   (A250/1:V) /* io/m : Length of array,
*                  /*          returned values
1 W3ARRAYVALUE     (I4/1:V) /* o/m : Value of variables
1 W3ARRAYVALUELENGTH (I4)   /* o/m : Length of variables
1 W3ARRAYMAXIMUM   /* o/m : Total number of variables
```

呼び出し方法

```
PERFORM W3READ-ENVIRONMENT-TEXTAREA W3START W3NAME  
W3VALUESERVER W3ARRAYCOUNTER(*)  
W3ARRAYVALUE(*) W3ARRAYVALUELENGTH(*)  
W3ARRAYMAXIMUM
```

32 ドキュメントへのテキストの書き込み

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3TEXT W3TEXTDYNAMIC W3TEXTLINE W3TEXTLINEDYNAMIC W3TEXTARRAY	E3TEXT	E3TEXT

説明

文字列をドキュメントに書き込みます。

出力の後に改行を作成する場合は、W3HTMLLINE または W3HTMLLINEDYNAMIC を使用します。

文字列の内部に改行を作成する場合は、##HTTP-NEWLINE を文字列内に挿入します。

W3TEXT および W3TEXTLINE を文字列に適用すると、文字列の末尾の空白が削除されます。

パフォーマンスを向上させるには、ダイナミック変数を使用します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

パラメータ

1. W3TEXT

```
1 W3TEXT      (A) DYNAMIC    /* i /m : Output string
```

2. W3TEXTDYNAMIC

```
1 W3DYNAMIC   (A) DYNAMIC    /* i /m : Output array
```

3. W3TEXTLINE

```
1 W3TEXT      (A) DYNAMIC    /* i /m : Output string with new line
```

4. W3TEXTLINEDYNAMIC

```
1 W3DYNAMIC   (A) DYNAMIC    /* i /m : Output string with new line
```

5. W3TEXTARRAY

```
1 H3ARRAYVALUE (A/1:v) DYNAMIC /* i /m : Output array  
1 H3VALUELENGTH (I4)           /* i /m : Length of output array
```

呼び出し方法

```
PERFORM W3TEXT W3TEXT  
PERFORM W3TEXTDYNAMIC W3TEXT  
PERFORM W3TEXTLINE W3TEXT  
PERFORM W3TEXTLINEDYNAMIC W3TEXT  
PERFORM W3TEXTARRAY W3ARRAYVALUE W3VALUELENGTH
```

33 出力エリアへの改行の書き込み

サブルーチン名

W3NEWLINE

説明

出力エリアに1つの改行（##HTTP_NEWLINE）を追加します。

このサブルーチンは、今後のバージョンのいずれかで削除されます。代わりに、W3TEXTDYNAMIC ##HTTP_NEWLINE を使用してください。

パラメータ

*/ NONE

呼び出し方法

PERFORM W3NEWLINE

34 テキストから HTML

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
W3TEXT-TO-HTML W3-ASCII-HTML-TABLE	E3TX2HTM	E3TX2HTM

説明

文字列を HTML 構文に変換します。特殊文字が含まれている場合に便利です。

サブプログラム W3-ASCII-HTML-TABLE は W3TEXT-TO-HTML および W3HTML から呼び出されます。このサブプログラムには、実行されるすべての変換のリストが含まれています。

このプログラムは必要に応じて変更および拡張できます。

パラメータ

```
1 W3HTML (A) DYNAMIC /* io/mH: HTML text conversion
```

呼び出し方法

```
PERFORM W3TEXT-TO-HTML W3HTML
```


35 テキストから XML

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
W3TEXT-TO-XML W3-ASCII-XML-TABLE	E3TX2XML	E3TX2XML

説明

文字列を XML 構文に変換します。特殊文字が含まれている場合に便利です。

サブプログラム W3-ASCII-XML-TABLE は W3TEXT-TO-XML から呼び出されます。このサブプログラムには、実行されるすべての変換のリストが含まれています。
このプログラムは必要に応じて変更および拡張できます。

パラメータ

```
1 W3XML (A) DYNAMIC /* io/mX: XML text conversion
```

呼び出し方法

```
PERFORM W3TEXT-TO-XML W3XML
```


36 テキストから URL

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
W3TEXT-TO-URL W3-ASCII-URL-TABLE	E3TX2URL	E3TX2URL

説明

文字列を URL 構文に変換します。特殊文字が含まれている場合に便利です。

サブプログラム W3-ASCII-URL-TABLE は W3TEXT-TO-URL から呼び出されます。このサブプログラムには、実行されるすべての変換のリストが含まれています。
このプログラムは必要に応じて変更および拡張できます。

パラメータ

```
1 W3URL (A) DYNAMIC /* io/mU: URL text conversion
```

呼び出し方法

```
PERFORM W3TEXT-TO-URL W3URL
```


37 返されるドキュメント内での置換

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3REPLACE	E3REPLAC	E3REPLAC

説明

すで書き込まれた出力ページ内で特定の文字列を検索し、検出された文字列をすべて新しい文字列に置換します。エンコードパラメータを使用すると、置換が実行される前に、データがエンコードされます。

- "" はエンコードなし
- "HTML" は HTML エンコード (例: < が < になります)
- "URL" は URL エンコード
- "XML" は XML エンコード (例: < が < になります)

パラメータ

```
1 W3ENCODING (A) DYNAMIC BY VALUE /* i /m : encoding
1 W3OLD      (A) DYNAMIC BY VALUE /* i /m : old string
1 W3NEW      (A) DYNAMIC BY VALUE /* i /m : new string
```

呼び出し方法

```
PERFORM W3REPLACE "$weather$" "fine, no clouds"
```


38 出力ページの読み取り

サブルーチン名	実行可能ファイルの例	表示可能ファイルの例
W3READ-OUTPUT	E3ROUT	E3ROUT

説明

すで書き込まれている出力ページをダイナミック変数に読み込みます。

パラメータ

```
1 W3HEADER (A) DYNAMIC /* o/m : written header
1 W3BODY   (A) DYNAMIC /* o/m : written body
```

呼び出し方法

```
PERFORM W3READ-OUTPUT W3HEADER W3BODY
```


39 アンカー

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-ANCHOR	H3ANCHOR	E3ANCHOR	E3ANCHOR

説明

ハイパーリンクの作成

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<A HREF="URL"> </A>
```

サポートされている属性

```
NAME=string
```

パラメータ

```
1 H3URL          (A) DYNAMIC /* i /m : URL of the Link. Enter
*                /*          'THIS' to reference
*                /*          the current page as URL.
1 H3NAME         (A) DYNAMIC /* i /M : Name of the anchor.
1 H3STRING       (A) DYNAMIC /* i /MH: String to be displayed
*                /*          as anchor text.
```

アンカー

呼び出し方法

```
PERFORM H3-ANCHOR H3URL H3NAME H3STRING
```

40 ボタン

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-BUTTON	H3BUTTON	E3BUTTON	E3BUTTON

説明

リセット／実行ボタンを作成します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<INPUT TYPE="submit|reset|image" NAME="string"> </INPUT>
```

サポートされている属性

```
VALUE="string", SRC="URL"
```

パラメータ

```
1 H3TYPE (A1) /* i /m : 'R' reset button
* /* 'S' submit button
* /* 'I' submit button with image
1 H3NAME (A) DYNAMIC /* i /M : Name of the button
1 H3VALUE (A) DYNAMIC /* i /M : Value of the input variable
1 H3URL (A) DYNAMIC /* i /M : URL of the picture to be used
```

ボタン

呼び出し方法

```
PERFORM H3-BUTTON H3TYPE H3NAME H3VALUE H3URL
```

41 チェックボックスグループ

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-CHECKBOX-GROUP	H3RCGROU	E3RCGROU	E3RCGROU

説明

チェックボックスグループを作成します。このグループはテーブル内でフォーマットできません。

表示

```
<INPUT TYPE="CHECKBOX" NAME="string">
```

サポートされている属性

```
VALUE=string, CHECKED
```

パラメータ

```
1 H3ARRAYCOUNT (I4) /* i /m : Number of group elements
1 H3ARRAYNAME (A/1:V) dynamic /* i /m : Name of the group variable
1 H3ARRAYVALUE (A/1:V) dynamic /* i /M : Default value of the
* /* group variable
1 H3ARRAYLABEL (A/1:V) /* i /MH: Label of the group element
1 H3ARRAYCHECKED ( L/1:V) /* i /M : Button selected by default
1 H3LINEBREAK ( L) /* i /m : Set line breaks between by
* /* the elements
1 H3ROW (N4) /* i /m : Set number of rows for
* /* tables
1 H3COLUMN (N4) /* i /m : Set number of columns for
* /* tables
```

呼び出し方法

```
PERFORM H3-CHECKBOX-GROUP H3ARRAYCOUNT H3ARRAYNAME(*)  
H3ARRAYVALUE(*) H3ARRAYLABEL(*)  
H3ARRAYCHECKED(*) H3LINEBREAK H3ROW H3COLUMN
```


42 コメント行

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-COMMENT	H3COMMEN	E3COMMEN	E3COMMEN

説明

HTML ページ内にコメント行を作成します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<!-- value -->
```

パラメータ

```
1 H3VALUE (A) DYNAMIC /* i /m : Value to set as comment
```

呼び出し方法

```
PERFORM H3-COMMENT H3VALUE
```


43 レベル n のヘッダー

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-HEADER	H3HEADER	E3HEADER	E3HEADER

説明

指定されたレベルのヘッダーを作成します。レベル 1 から 6 まで作成できます。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

<H1> </H1> または

<H2> </H2> または

<H3> </H3> または

<H4> </H4> または

<H5> </H5> または

<H6> </H6>

レベルnのヘッダー

パラメータ

```
1 H3LEVEL (N2) /* i /m : Level of the header
1 H3HTML (A) DYNAMIC /* i /mH: HTML text to be set
```

呼び出し方法

```
PERFORM H3-HEADER H3LEVEL H3HTML
```

44 イメージ

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-IMAGE	H3IMAGE	E3IMAGE	E3IMAGE

説明

イメージを表示します。イメージ自体は Natural 内に保存できません。このため、すべての画像は HTTP サーバーを使用して保存する必要があります。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<IMG SRC="URL">
```

サポートされている属性

```
ALT=string, HEIGHT=n, WIDTH=n, ALIGN=left|right|top|middle|bottom
```

パラメータ

```
1 H3URL      (A) DYNAMIC /* i /m : URL of the picture source
1 H3STRING  (A) DYNAMIC /* i /M : Alternative name string
*                               /*           for non-GUI browsers
1 H3HEIGHT  (N4)        /* i /M : Height if the picture
1 H3WIDTH   (N4)        /* i /M : Width of the picture
1 H3ALIGN   (A1)        /* i /M : Align the picture to
*                               /*           'L' Left
*                               /*           'R' Right
*                               /*           'T' Top
*                               /*           'B' Bottom
*                               /*           'M' Middle
```

呼び出し方法

```
PERFORM H3-IMAGE H3URL H3STRING H3HEIGHT H3WIDTH H3ALIGN
```

45 入力

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-INPUT	H3INPUT	E3INPUT	E3INPUT

説明

入力フィールドを作成します。作成可能なフィールドタイプは、テキスト、パスワード、および非表示です。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<INPUT TYPE="text|password|hidden" NAME="string">
```

サポートされている属性

```
VALUE=string, MAXLENGTH=n, SIZE=n
```

パラメータ

```
1 H3TYPE (A1) /* i /m : Type of the input field
* /* 'T' Text (default)
* /* 'P' Password
* /* 'H' Hidden
1 H3NAME (A) DYNAMIC /* i /M : Name of the input variable
1 H3VALUE (A) DYNAMIC /* i /M : Default value of the input variable
1 H3SIZE (N4) /* i /M : Size of the input box
1 H3MAX (N4) /* i /M : Maximum length of the input text
```

入力

呼び出し方法

```
PERFORM H3-INPUT H3TYPE H3NAME H3VALUE H3SIZE H3MAX
```


46 改行

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-LINE-BREAK H3-LINE_BREAK	H3LBREA H3LBREAK	E3LBREAK	E3LBREAK

説明

強制的に改行します。場合によっては後ろに HTML テキストが追加されます。

以前のバージョンからの変更点

パラメータはオプションとしてマークされているため、両方のバージョンを同等に使用できます。バージョン H3-LINE-BREAK のみを使用してください。H3-LINE-BREAK は今後のバージョンで削除されます。

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

パラメータ

1. H3-LINE-BREAK

```
1 H3HTML (A) DYNAMIC /* i /OH: HTML text after the line break
```

2. H3-LINE_BREAK

```
1 H3HTML (A) DYNAMIC /* i /OH: HTML text after the line break
```

呼び出し方法

```
PERFORM H3-LINE-BREAK  
PERFORM H3-LINE-BREAK 1X  
PERFORM H3-LINE-BREAK H3HTML
```

47 フォーム

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-OPEN-FORM H3-CLOSE-FORM	H3OFORM H3CFORM	E3FORM	E3FORM

説明

フォームを作成します。作成前に H3-OPEN-FORM を実行し、作成後に H3-CLOSE-FORM を実行する必要があります。

H3-CLOSE-FORM を実行しない場合は、開いているすべてのフォームが H3-CLOSE-HTML によって閉じられます。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<FORM ACTION="URL" METHOD="get|post"> </FORM>
```

サポートされている属性

```
ACTION=URL, METHOD=get|post
```

フォーム

パラメータ

1. H3-OPEN-FORM

```
1 H3METHOD (A1)          /* i /m : 'G' GET
*                        /*      : 'P' POST
1 H3URL      (A) DYNAMIC /* i /m : URL to be called
```

2. H3-CLOSE-FORM

```
/* none
```

呼び出し方法

```
PERFORM H3-OPEN-FORM H3METHOD H3URL
PERFORM H3-CLOSE-FORM
```

48 HTML ドキュメント

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-OPEN-HTML H3-OPEN-HTML-JAVASCRIPT H3-CLOSE-HTML	H3OHTML H3CHTML	E3HTML	E3HTML

説明

ヘッダー、タイトル、および本文の先頭が記述された HTML ドキュメントを作成します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

1. H3-OPEN-HTML

```
<HTML>
<HEAD>
<TITLE>TITLE</TITLE>
</HEAD>
<BODY BACKGROUND="URL", BGCOLOR="#RPG">
```

2. H3-OPEN-HTML-JAVASCRIPT

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE='JavaScript' SRC='URL'></SCRIPT>
<SCRIPT LANGUAGE='JavaScript'>
<!-- hide script from old browsers
PROGRAM
// end hiding from old browsers -->
```

```
</SCRIPT>
<TITLE>TITLE</TITLE>
</HEAD>
<BODY BACKGROUND="URL", BGCOLOR="#RGB">
```

3. H3-CLOSE-HTML

```
</BODY>
</HTML>
```

サポートされている属性

BACKGROUND=URL, BGCOLOR=#RGB, SRC='URL'

パラメータ

1. H3-OPEN-HTML

```
1 H3TITLE      (A) DYNAMIC /* i /m: Title of the HTML document
1 H3BGCOLOR    (A) DYNAMIC /* i /M: Background colour
1 H3BGPICTURE (A) DYNAMIC /* i /M: Background picture
```

2. H3-OPEN-HTML-JAVASCRIPT

```
1 H3TITLE      (A) DYNAMIC /* i /m: Title of the HTML document
1 H3BGCOLOR    (A) DYNAMIC /* i /M: Background colour
1 H3BGPICTURE (A) DYNAMIC /* i /M: Background picture
1 H3JAVASRC    (A) DYNAMIC /* i /M: ULR to a JavaScript source
1 H3JAVA       (A/1:V) DYNAMIC /* i /M: JavaScript
1 H3JAVACOUNTER (I4) /* i /M: Number of JavaScript source lines
1 H3ONLOAD     (A) DYNAMIC /* i /M: onload event handler
1 H3ONUNLOAD   (A) DYNAMIC /* i /M: onunload event handler
```

3. H3-CLOSE-FORM

```
/* none
```

呼び出し方法

```
PERFORM H3-OPEN-HTML H3TITLE H3BGCOLOR H3BGPICTURE
PERFORM H3-OPEN-HTML-JAVASCRIPT H3TITLE H3BGCOLOR H3BGPICTURE H3JAVASRC H3JAVA
H3JAVACOUNTER H3ONLOAD H3ONUNLOAD
PERFORM H3-CLOSE-HTML
```

49 List

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-OPEN-LIST H3-LIST-ITEM H3-CLOSE-LIST	H3OLIST H3LISTI H3CLIST	E3LIST	E3LIST

説明

さまざまなタイプのリストを作成します。指定可能なタイプは次のとおりです。

- 整列されていないリスト
- 整列されたリスト
- メニュー項目リスト
- ディレクトリリスト

最大 10 レベルのカスケードリストがサポートされています。また、一度に複数のレベルを閉じることもできます。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

1. H3-OPEN-LIST

<DIR> または
<MENU> または
 または

2. H3-LIST-ITEM

3. H3-CLOSE-LIST

</DIR> または
</MENU> または
 または

サポートされている属性

TYPE=*disc|square|circle* TYPE=1|a|A|i|I

パラメータ

1. H3-OPEN-LIST

```
1 H3TYPE (A1) /* i /m: Set list as:
* /* 'O' ordered list
* /* 'U' unordered list
* /* 'D' directory list
* /* 'M' menu list
1 H3BULLET (A1) /* i /m: Type of list if ordered list:
* /* '1' Arabic numbers (default) (1, 2, 3, ...)
* /* 'a' Alphanumeric, lowercase (a, b, c, ...)
* /* 'A' Alphanumeric, uppercase (A, B, C, ...)
* /* 'i' Roman numbers, lowercase (i, ii, iii, ...)
* /* 'I' Roman numbers, uppercase (I, II, III, ...)
* /* i /m: Type of bullet if unordered list:
* /* 'D' Disc
* /* 'S' Square
* /* 'C' Circle
```

2. H3-LIST-ITEM

```
1 H3VALUE (A) DYNAMIC /* i /m: Item text
```

3. H3-CLOSE-LIST

```
1 H3LEVEL (N2) /* i /m: Levels to be closed
```

呼び出し方法

```
PERFORM H3-OPEN-LIST H3TYPE H3BULLET  
PERFORM H3-LIST-ITEM H3VALUE  
PERFORM H3-CLOSE-LIST H3LEVEL
```


50 パラグラフ

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-PARAGRAPH	H3PARAGR	E3PARAGR	E3PARAGR

説明

新しいパラグラフを作成します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

<P ALIGN=*left|right|center*> or <P>

サポートされている属性

ALIGN=*left|right|center*

パラメータ

```
1 H3ALIGN (A1)            /* i /m : Align the paragraph to:
*                        /*        'L' Left (default)
*                        /*        'R' Right
*                        /*        'C' Center
1 H3HTML (A) DYNAMIC /* i /mh: HTML text after the paragraph
```

パラグラフ

呼び出し方法

```
PERFORM H3-PARAGRAPH H3ALIGN H3HTML
```

51 ラジオボタングループ

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-RADIO-GROUP	H3RBGROU	E3RBGROU	E3RBGROU

説明

ラジオボタングループを作成します。このグループはテーブル内でフォーマットできます。

表示

```
<INPUT TYPE=RADIO NAME=string>
```

サポートされている属性

```
VALUE=string, CHECKED
```

パラメータ

```
1 H3ARRAYCOUNT (I4) /* i /m : Number of group elements
1 H3NAME (A) dynamic /* i /m : Name of the group variable
1 H3ARRAYVALUE (A/1:V) dynamic /* i /M : Default value of the default group
* /* variable
1 H3ARRAYLABEL (A/1:V) /* i /mH: Label of the group element
1 H3ISCHECKED (I4) /* i /M : Number of default selected
* /* button
1 H3LINEBREAK (L) /* i /M : Set line breaks between
* /* buttons
1 H3ROW (N4) /* i /m : Set number of rows for tables
1 H3COLUMN (N4) /* i /m : Set number of columns for
* /* tables
```

呼び出し方法

```
PERFORM H3-RADIO-GROUP H3ARRAYCOUNT H3NAME  
H3ARRAYVALUE(*) H3ARRAYLABEL(*)  
H3ISCHECKED H3LINEBREAK H3ROW H3COLUMN
```

52 横罫線

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-RULE	H3RULE	E3RULE	E3RULE

説明

横罫線を作成します。幅はパーセントで指定します。

表示

<HR> または
<HR WIDTH="p%">

サポートされている属性

WIDTH=p%

パラメータ

```
1 H3WIDTH (N4) /* i /m : Width in percent
```

呼び出し方法

```
PERFORM H3-RULE H3WIDTH
```


53 スクロールリスト

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-SCROLLING-LIST	H3SCLIST	E3SCLIST	E3SCLIST

説明

スクロールリストを作成します。作成したリストは、コンボボックスまたはリストボックスとして表示できます。以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<SELECT NAME=string>  
<OPTION> </OPTION>  
</SELECT>
```

サポートされている属性

SIZE=*n*, MULTIPLE

VALUE=*string*, SELECTED

スクロールリスト

パラメータ

```
1 H3SIZE          (N4)          /* i /m : Number of lines:
*                  /*          =1 combo box
*                  /*          >1 list box
1 H3ARRAYCOUNT  (I4)          /* i /m : Number of list elements
1 H3NAME          (A)dynamic    /* i /m : Name of the group variable
1 H3ARRAYVALUE    (A/1:V) dynamic /* i /M : Default value of the list values
1 H3ARRAYLABEL    (A/1:V) dynamic /* i /MH: Label of the list elements
1 H3ARRAYSELECTED (L/1:V)      /* i /M : Elements selected by
*                  /*          default
1 H3MULTIPLE      (L)          /* i /M : Multiple selection allowed
```

呼び出し方法

```
H3-SCROLLING-LIST H3SIZE H3ARRAYCOUNT H3NAME
                  H3ARRAYVALUE(*) H3ARRAYLABEL(*)
                  H3ARRAYSELECTED(*) H3MULTIPLE
```

54 テーブル

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-TABLE H3-TABLE-COLOR	H3TABLE H3TABLEC	E3TABLE	E3TABLE

説明

所定の配列の単純なテーブルを作成します。H3-TABLE-COLORを使用すると、見出しとテーブルに異なる背景色を設定できます。行は交互に色付けされます。

表示

```
<TABLE>
<TR>
<TH>... </TH>
</TR>
...
<TR>
<TD>... </TD>
</TR>
...
</TABLE>
```

サポートされている属性

ALIGN=*left|right|center* , BORDER=*n*, NOWRAP

パラメータ

1. H3-TABLE

1	H3ROW	(N4)	/* i /m : Number of rows
1	H3COLUMN	(N4)	/* i /m : Number of columns
1	H3ARRAY2VALUE	(A/1:V,1:V)dynamic	/* i /mh: Table elements
1	H3ARRAY2ALIGN	(A1/1:V,1:V)	/* i /M : Alignment of the table cells
*			/* 'L' Left (default)
*			/* 'R' Right
*			/* 'C' Center
1	H3ARRAY2NOWRAP	(L/1:V,1:V)	/* i /m : No automatic wrapping
1	H3HEADLINE	(L)	/* i /M : 1st line as headline
1	H3ALIGN	(A1)	/* i /M : Alignment of the table
*			/* 'L' Left (default)
*			/* 'R' Right
*			/* 'C' Center
1	H3SUPPRESSEEMPTY	(L)	/* i /m : Set to TRUE if cell is
*			/* to be displayed
*			/* despite being empty
1	H3ISHTML	(L)	/* i /m : Transform value to
*			/* HTML
1	H3BORDER	(N4)	/* i /M : Set border size

2. H3-TABLE-COLOR

1	H3ROW	(N4)	/* i /m : Number of rows
1	H3COLUMN	(N4)	/* i /m : Number of columns
1	H3TITLECOLOR	(A032)	/* i /M : Color of headline
1	H3LINECOLOR	(A032)	/* i /M : Color of lines
1	H3ARRAY2VALUE	(A/1:V,1:V) dynamic	/* i /mh: Table elements
1	H3ARRAY2ALIGN	(A001/1:V,1:V)	/* i /m : Alignment of the table cells
*			/* 'L' Left (default)
*			/* 'R' Right
*			/* 'C' Center
1	H3ARRAY2NOWRAP	(L/1:V,1:V)	/* i /m : No automatic wrapping
1	H3HEADLINE	(L)	/* i /m : 1st line as headline
1	H3ALIGN	(A1)	/* i /M : Alignment of the table
*			/* 'L' Left
*			/* 'R' Right
*			/* 'C' Center (default)
1	H3SUPPRESSEEMPTY	(L)	/* i /m : Set to TRUE if cell is
*			/* to be displayed
*			/* despite being empty
1	H3ISHTML	(L)	/* i /m : Transform value to
*			/* HTML
1	H3BORDER	(N4)	/* i /m : Set border size

呼び出し方法

```
PERFORM H3-TABLE H3ROW H3COLUMN H3ARRAY2VALUE(*,*)  
    H3ARRAY2ALIGN(*,*) H3ARRAY2NOWRAP(*,*)  
    H3HEADLINE H3ALIGN H3SUPPRESSEEMPTY  
    H3ISHTML H3BORDER  
  
PERFORM H3-TABLE-COLOR H3ROW H3COLUMN H3TITLECOLOR H3LINECOLOR  
    H3ARRAY2VALUE(*,*) H3ARRAY2ALIGN(*,*) H3ARRAY2NOWRAP(*,*)  
    H3HEADLINE H3ALIGN H3SUPPRESSEEMPTY  
    H3ISHTML H3BORDER
```


55 汎用タグ

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-TAG	H3TAG	E3TAG	E3TAG

説明

HTML ページ内に汎用タグ（タグテンプレート）を作成します。このタグテンプレートでは、コードを書き込むことができるフレームワークを作成します。

以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<tag> </tag>
```

パラメータ

```
1 H3PRE (A) DYNAMIC /* i /m : Open tag value
1 H3HTML (A) DYNAMIC /* i /m : HTML inside the tag
1 H3POST (A) DYNAMIC /* i /m : Close tag value
```

呼び出し方法

```
PERFORM H3-TAG H3PRE H3HTML H3POST
```


56 テキストエリア

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-TEXT-AREA	H3TXAREA	E3TXAREA	E3TXAREA

説明

テキストエリアを作成します。以前のバージョンからの変更点

インターフェイスのすべての (A250) BY VALUE 変数が (A) DYNAMIC BY VALUE に変更されました。

古いインターフェイス (A250) は新しいインターフェイスと互換性があるため、プログラムを再カタログする必要はありません。

表示

```
<TEXTAREA NAME="string"> </TEXTAREA>
```

サポートされている属性

```
ROWS=n, COLS=n
```

パラメータ

```
1 H3ARRAYCOUNT (I4) /* i /m : Number of text
1 H3NAME (A) dynamic /* i /m : Name of the text variable
1 H3ARRAYTEXT (A/1:V) dynamic /* i /M : Default value of the text variable
1 H3ROW (N4) /* i /M : Set number of rows
1 H3COLUMN (N4) /* i /M : Set number of columns
```

呼び出し方法

```
PERFORM H3-TEXT-AREA H3ARRAYCOUNT H3NAME H3ARRAYTEXT(*)  
H3ROW H3COLUMN
```

57 テキストから URL デコード形式

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-TEXT-TO-URL H3-ASCII-URL-TABLE	H3TX2URL	E3TX2URL	E3TX2URL

説明

文字列を URL-decoded 構文に変換します。特殊文字が含まれている場合に便利です。H3-TEXT-TO-URL の代わりに [W3-TEXT-TO-URL](#) を使用してください。H3-ASCII-URL-TABLE の代わりに [W3-ASCII-URL-TABLE](#) を使用してください。

サブプログラム W3-ASCII-URL-TABLE は H3-TEXT-TO-URL から呼び出されます。このサブプログラムには、実行されるすべての変換のリストが含まれています。このプログラムは必要に応じて変更および拡張できます。

パラメータ

```
1 H3COUNT (I4) /* o/m : Length of the converted string
1 H3STRING (A250) /* io/m : URL-decoded text after conversion
```

呼び出し方法

```
PERFORM H3-TEXT-TO-URL H3COUNT H3STRING
```


58 時刻／日付文字列

サブプログラム名	ソース名	実行可能ファイルの例	表示可能ファイルの例
H3-TIME_DATE H3-TIME-DATE	H3TIMDAT H3TIMDA	E3TIMDAT	E3TIMDAT

説明

LOG時間を使用して"generated: ..."という文字列を作成するか、または現在の時刻／日付を使用して、オフセットを含み、HTTPと互換性のある時刻／日付文字列を作成します（GMTまたはGMTへのオフセットはNaturalによって認識されないため）。

表示

generated: *time/date*

パラメータ

1. H3-TIME_DATE

```
/* none
```

2. H3-TIME-DATE

```
1 H3ADDMINUTE (I4) /* i /m : Adds minutes to time
1 H3ADDDAY (I4) /* i /m : Adds days to date
1 H3DATETIME (A29) /* o/m : Generated string
```

呼び出し方法

```
PERFORM H3-TIME_DATE  
PERFORM H3-TIME-DATE H3ADDMINUTE H3ADDDAY H3DATETIME
```

59 すべての Natural ライブラリのリスト

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
NAT-LIB	NAT-LIB NAT-LIB?FNAT=N	NAT-LIB

説明

HTML ページを生成し、すべての使用可能な Natural ライブラリを表示します。FNAT パラメータが指定されていない場合は、デフォルトのユーザーライブラリが表示されます。

パラメータ

FNAT=	N = システムライブラリ U = ユーザーライブラリ (デフォルト)
EXPIRE=	現在の日付に日数を加算し、その日付を有効期限として設定します。
START=	表示されるオブジェクトセットを選択するワイルドカード

呼び出し方法

```
NAT-LIB NAT-LIB?FNAT=N
```


60 オンライン Natural Web インターフェイス サブプログラムの実行

Natural プログラム
WEB-ONL

説明

デバッグまたはテストに備えて、オンラインで Natural Web インターフェイス サブプログラムを実行しておくると便利です。生成されたページの出力は Natural テキストオブジェクトとして保存されます。92 文字以上の行は分割されます。環境変数を設定できます。変数をサーバーの変数として設定する必要がある場合は、名前の先頭にアンパサンドを追加します。

呼び出し方法

プログラム WEB-ONL を Natural の NEXT プロンプトから実行します。

61 Natural Web インターフェイスで使用する Natural サ ブプログラムの生成

Natural プログラム	実行可能ファイル生成の結果	表示可能ファイル生成の結果
WEB-WIZ	基本サブルーチン HTML エクステンション	HTTPApi HTMLApi

説明

デフォルトのプログラムを生成します。この機能は、今後のバージョンのいずれかで削除されます。を使用します。

入力マップ

```
12:12:40          ***** Natural Web Subprogram Wizard *****          2003-01-15
                                     - Main Menu -                          Library SYSWEB

Subprogram Name ..... DUMMY_
Title ..... HTTP/HTML API WIZARD_____
Header ..... HTTP/HTML API WIZARD_____

Use HTML extension .. X
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---  
Exit
```

呼び出し方法

プログラム WEB-WIZ を Natural の *NEXT* プロンプトから実行します。

62 Natural ライブラリのディレクトリのリスト

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
NAT-DIR	NAT-DIR?lib=sysweb NAT-DIR?lib=sysweb&start=E3*	NAT-DIR

説明

Natural ライブラリのディレクトリ情報が記載された HTML ページを生成します。ライブラリパラメータが定義されていない場合は、現在のライブラリが表示されます。

パラメータ

LIB=	Natural ライブラリ
EXPIRE=	現在の日付に日数を加算し、その日付を有効期限として設定します。
START=	表示されるオブジェクトセットを選択するワイルドカード

呼び出し方法

```
nat-dir?lib=sysweb
```


63 呼び出された Natural サブプログラムに HTTP サーバーから渡されたすべてのパラメータのリスト

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
NAT-ENV	NAT-ENV	NAT-ENV

説明

呼び出された Natural サブプログラムに HTTP サーバーから渡されたすべてのパラメータが記載された HTML ページを生成します。

呼び出し方法

```
nat-env
```


64 Naturalソースオブジェクトとして保存されたHTML ページを返す

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例	テキストオブジェクト
NAT-HTML	NAT-HTML?lib=sysweb&source=t3-html	NAT-HTML	T3-HTML

説明

Natural ソースオブジェクトとして保存された HTML ページを表示します。

パラメータ

LIB=	Natural ライブラリ
SOURCE=	ソース名

呼び出し方法

```
nat-html?lib=sysweb&source=HTML
```


65 Natural Web インターフェイスの現在の設定をリストします。

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
NAT-INFO	NAT-INFO	NAT-INFO

説明

HTTP ブラウザ、HTTP サーバー、通信ソフトウェア（RPC/DCOM）、および Natural 環境に関する情報が記載された HTML ページを生成します。

呼び出し方法

```
nat-info
```

66 Natural オブジェクトのソースのリスト

サブプログラム名	実行可能ファイルの例	表示可能ファイルの例
NAT-LIST	NAT-LIST?lib=sysweb&source=h3image	NAT-LIST

説明

Natural ソースオブジェクトのリストが記載された HTML ページを生成します。

パラメータ

LIB=	Natural ライブラリ
SOURCE=	ソース名
EXPIRE=	現在の日付に日数を加算し、その日付を有効期限として設定します。
LINE-NUMBERS=	指定可能な値は OFF のみです。

呼び出し方法

```
nat-list?lib=sysweb&source=H3IMAGE
```


67 オンラインドキュメント

サブプログラム名	実行可能ファイルの例
NAT-DOCU	NAT-DOCU

説明

Natural ソースオブジェクトとして保存されたオンラインドキュメントを表示します。

パラメータ

LIB=	Natural ライブラリ
SOURCE=	ソース名
EXPIRE=	現在の日付に日数を加算し、その日付を有効期限として設定します。

呼び出し方法

```
nat-docu
```


68 XML ツールキット

はじめに

XML (eXtensible Markup Language: 拡張マークアップ言語) は非常に重要な関連標準であり、Web アプリケーションの推進力となっています。Natural for Mainframes バージョン 4 以降では、XML ドキュメントの作成、処理、および利用が可能となりました。高度なネスト構造を持つ XML ドキュメントの一式は非常に容量が大きくなることがあるため、Natural のデータ定義およびデータ操作の機能は拡張されて、ラージ変数 (1 GB 以下) およびダイナミック変数の使用が可能となりました。

このドキュメントの内容

XML ツールキットを使用すると、Natural で XML ドキュメント処理機能を使用できます。Natural データ定義を XML 文書型定義 (DTD) から生成することも、その逆の処理を行うこともできます。Natural 変数の内容は XML ドキュメントにシリアル化できます。また、XML ドキュメントを解析して Natural 変数にすることもできます。

このドキュメントでは、外部プログラム部分のない Natural for Mainframes 環境での XML の使用を例示するサンプルアプリケーションについて説明します。

次のトピックについて説明します。

- [はじめに](#)
- [XML ツールキットの使用](#)
- [特定の生成オプションの設定](#)
- [Natural データソースの使用](#)
- [外部データソースの使用](#)
- [Natural のシンプルな XML パーサー](#)

- 例
- 解析エラーメッセージ

ここに記載されている仕様は変更されることがあります。変更内容は今後のリリースノートおよび新しいエディションで公開されます。

69 はじめに

■ XML ツールキットの特徴	198
■ XML ツールキットの説明	198
■ 今後の方針	202
■ 考慮事項と制限事項	202

次のトピックについて説明します。

XML ツールキットの特徴

- ダイナミック変数を使用する、Natural ベースの XML パーサー。
- 以下の機能があります。
 - Natural データ構造の DTD 定義への変換
 - Natural データ構造を XML ドキュメントとして保存するための COMPRESS ステートメントの生成
 - Natural ベースのパーサーに対するコールバックの生成

XML ツールキットの説明

目的

Natural XML ツールキットでは、Natural に追加の XML 機能を提供し、Natural アプリケーションと XML の統合を強化します。

一般的なアーキテクチャ

Natural XML ツールキットは、Natural プログラムの集合で構成されています。XML ツールキットプログラムを顧客のアプリケーションに統合すると、XML データへのアクセスを可能にしたり、XML フォーマットのデータを Natural から提供したりできます。

Natural XML ツールキットでは、次の機能を呼び出します。

XML ツールキットの機能

1. Natural データ定義と DTD の間のマッピング
2. XML トークンから NAT データへの変換
Natural データ構造を作成した後、XML ドキュメントを解析してそのデータ構造に保存する必要があります。指定したデータを Natural データ構造に保存できる Natural の実装が生成されます。
3. NAT データから XML ドキュメントへの変換（シリアライズ）
シリアライズとは、Natural データ構造に保存されたデータを取得し、指定された記述に従って XML ドキュメントを作成する処理のことです。

XML ツールキット機能へのユーザーインターフェイスは、Natural ダイアログによって実装されます。DTD はワークファイルとしてアクセスされ、生成された Natural オブジェクトは Natural システムファイルに直接保存されます。

Natural データ定義の DTD へのマップ

このマッピングは、Natural データ構造を XML タグと結合するための最初の手順であり、Natural データを XML タグとして表現するために必要となります。以下の例は、Natural と DTD の間のマッピング、およびいくつかの明確な違いを示しています。

Natural PDA

```

                                Press ESC to enter command mode
Mem: EMPL      Lib: SYSEXXT  Type: PARAMETER  Bytes: 1072  Line:   0 of: 26
C T  Comment
*   *** Top of Data Area ***
  1  EMPLOYEE
  2  ATTRIBUTES_OF_EMPLOYEE
  3  PERSONNEL-ID                A           8
*
  2  FULL-NAME
  3  FIRST-NAME                 A          20
  3  NAME                       A          20
*
  2  FULL-ADDRESS
  3  C@ADDRESS-LINE             I           4
  3  ADDRESS-LINE              A          20 (1:6)
  3  CITY                      A          20
  3  ZIP                       A          20
  3  COUNTRY                   A           3
*
  2  TELEPHONE
  3  AREA-CODE                 A           6
  3  PHONE                     A          15

```

生成された DTD

```

<!ELEMENT EMPLOYEE (PERSONNEL-ID, FULL-NAME, FULL-ADDRESS, TELEPHONE, INCOME*)>
<!ELEMENT PERSONNEL-ID (#PCDATA ) >
<!ELEMENT FULL-NAME (FIRST-NAME, NAME )>
    <!ELEMENT FIRST-NAME (#PCDATA )>
    <!ELEMENT NAME (#PCDATA )>
<!ELEMENT FULL-ADDRESS (ADDRESS-LINE*, CITY, ZIP, COUNTRY )>

```

```
<!ELEMENT ADDRESS-LINE (#PCDATA )>
<!ELEMENT CITY (#PCDATA )>
<!ELEMENT ZIP (#PCDATA )>
<!ELEMENT COUNTRY (#PCDATA )>
...
```

生成された DTD は、後で XML ドキュメントへのシリアルライズに使用されます（下記参照）。

データの XML へのシリアルライズ

Natural プログラムの実行中、DEFINE DATA ステートメントに定義されたデータの内容は、"実際の" 内容で埋められます。この内容は、シリアルライズ中に XML 形式でダイナミック変数に書き込まれます。このとき、以前に生成された DTD が入力として使用されます。

データをシリアルライズするプログラムは、XML ツールキットによって生成されます。

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<EMPLOYEE PERSONNEL-ID="30016509">
<FULL-NAME>
  <FIRST-NAME>ELSPETH</FIRST-NAME>
  <NAME>TROWBRIDGE</NAME>
</FULL-NAME>
<FULL-ADDRESS>
  <ADDRESS-LINE>91 BACK LANE</ADDRESS-LINE>
  <ADDRESS-LINE>BILSTON</ADDRESS-LINE>
  <ADDRESS-LINE>STAFFORDSHIRE</ADDRESS-LINE>
  <CITY>BILSTON</CITY>
  <ZIP>ST2 3KA</ZIP>
  <COUNTRY>UK</COUNTRY>
</FULL-ADDRESS>
<TELEPHONE>
  <PHONE>863322</PHONE>
  <AREA-CODE>0602</AREA-CODE>
</TELEPHONE>
...
```

DTD の Natural データ定義へのマップ

DTD の Natural データ構造へのマッピングにも、違いがあります。DTD では XML ドキュメントに何人分のレコードが含まれるかを指定しないため、ツールキットでは、最大 "v" 人分のレコードが含まれることを想定します。アプリケーションプログラマは、正確な数を把握すれば、この数に合わせてデータ構造を修正できます。同じような制限がデータ長にも存在します。DTD では、各人のレコードにデータ長に関する情報が含まれていません。このため、ツールキットで、データ構造に長さ 253（現在の最大長）のフィールドを作成します。

```

* DTD E:\SAG\nat\4.2\fnat\SYSEXXT\RES\empl.dtd
COMPRESS &1& '<EMPLOYEE'
  ' PERSONNEL-ID="'EMPLOYEE.PERSONNEL-ID "'
  '>' INTO &1& LEAVING NO
/* now the children
COMPRESS &1& '<FULL-NAME'
  '>' INTO &1& LEAVING NO
/* now the children
COMPRESS &1& '<FIRST-NAME'
  '>'
  EMPLOYEE.FIRST-NAME
  '</FIRST-NAME>' INTO &1& LEAVING NO
COMPRESS &1& '<NAME'
  '>'
  EMPLOYEE.NAME
  '</NAME>' INTO &1& LEAVING NO
/*
COMPRESS &1& '</FULL-NAME>' INTO &1& LEAVING NO
COMPRESS &1& '<FULL-ADDRESS'
  '>' INTO &1& LEAVING NO
/* now the children
FOR &2& = 1 TO EMPLOYEE.C@ADDRESS-LINE
  COMPRESS &1& '<ADDRESS-LINE'
    '>'
    EMPLOYEE.ADDRESS-LINE(&2&)
    '</ADDRESS-LINE>' INTO &1& LEAVING NO
END-FOR
...

```

XML ファイルの解析および Natural データへの割り当て

```

* DTD E:\SAG\nat\4.2\fnat\SYSEXXT\RES\empl.dtd
DECIDE ON FIRST &1&
  VALUE 'EMPLOYEE'
    RESET INITIAL EMPLOYEE
  VALUE 'EMPLOYEE/@PERSONNEL-ID'
    /* #REQUIRED
    EMPLOYEE.PERSONNEL-ID := &3&
  VALUE 'EMPLOYEE/FULL-NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME/$'
    EMPLOYEE.FIRST-NAME := &3&
  VALUE 'EMPLOYEE/FULL-NAME/NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/NAME/$'
    EMPLOYEE.NAME := &3&

```

...

今後の方針

XML ツールキットによって、Natural で XML がより高度にサポートされるようになりました。XML ツールキットは、最初にリリースされた後、拡張される可能性があります。ただし、主な目標は、将来のリリースにおいて Natural の強力な言語の一部として XML 機能を実装することです。

考慮事項と制限事項

XML ツールキットを使用する場合は、次の制限を考慮する必要があります。

■ 非常に大きなデータ構造

非常に大きなデータ構造

データ構造のデータフィールドおよびグループの数が約700を超えた場合は、次のメッセージが表示されます。

```
Input Structure too big
```

解決策

データ構造をより小さいセクションに分割します。

70 XML ツールキットの使用

▪ 前提条件	204
▪ ワークファイルの処理	204
▪ 出力ファイル	207
▪ アプリケーションの起動	207
▪ PF キーの割り当て	208

次のトピックについて説明します。

前提条件

ストレージの要件

XML ツールキットでは、処理された XML DTD のサイズ、複雑さ、または再帰の深さに応じて、DATSIZE バッファ内に最大数百 KB の容量が必要になる場合があります。

生成されたコールバックルーチンを使用して XML ドキュメントを解析するには、ドキュメント全体がダイナミック変数に含まれている必要があります。

スキーマのサポート

メインフレームコンピュータの XML ツールキットでは、スキーマはサポートされておらず、DTD のみを処理できます。

ワークファイルの処理

デフォルトで、XML ツールキットでは、Natural ワークファイル 12 が DTD への入出力ファイルとして使用され、ワークファイル 13 がエラーログファイルとして使用されます。

XML ツールキットに付属のパーサーのサンプルプログラムでは、ワークファイル 12 から解析対象の XML ドキュメントを読み込み、解析プロセスの出力をワークファイル 13 に書き込みます。

ワークファイルのサポート

Entire Connection を使用中の PC の場合には、PC ワークファイルを使用して、Natural ワークファイルをサポートしていない TP モニタで XML ツールキットを実行できます。

生成されたコールバックルーチンを使用して解析される XML ドキュメント、または生成されたシリアライゼーションコードを使用して生成される XML ドキュメントは、ワークファイル経由でもアクセスできます。データは、データタイプ ALPHA DYNAMIC の変数を使用してアクセスする必要があります。ワークファイルのデータタイプは UNFORMATTED である必要があります。

PC ファイルには、SYSEXXT ライブラリで提供されるサブプログラム XML2PCWR を使用してアクセスできます。XML2PCWR は、PC ワークファイルの ALPHA DYNAMIC 変数の書き込みまたは読み込みを行います。

XML2PCWR の呼び出し規則：

```
CALLNAT 'XML2PCWR' XML-PAGE FILENUMBER OPERATION RETCODE
```

パラメータの定義：

```
DEFINE DATA LOCAL
  1 XML-PAGE          (A)      DYNAMIC          /* XML page
  1 FILENUMBER        (I2)
  1 OPERATION         (A2)     /* W/R: write/read
  1 RETCODE           (I4)     /* must be 0!
END-DEFINE
```

例：

次の例では、生成されたドキュメントを Standard Work File 10 および PC Work File 15 に書き込みます。

```
* -----
* CLASS NATURAL XML TOOLKIT - UTILITIES
*
* SDEMO_P1
*
* DESCRIPTION
* Serialize a given Data structure.
*
*
* AUTHOR SAG 01.2005
*
* VERSION 4.12.
*
* (c) Copyright Software AG 2001-2005. All rights reserved.
*
* -----
*
DEFINE DATA
LOCAL USING EMPL /* add generated data structure
LOCAL
1 XML (A) DYNAMIC
*
1 OUT (A72)
1 II (I4)
*
1 #CX (I4)
1 #CY (I4)
1 #CZ (I4)
1 FILENUMBER (I2)
```

```
1 OPERATION (A2)
1 RETCODE (I4)
END-DEFINE
/*[ initialize
EMPLOYEE.PERSONNEL-ID := 4711
*
EMPLOYEE.FIRST-NAME := "ADKINSON"
EMPLOYEE.NAME := "MARTHA"
*
EMPLOYEE.CÂ$ADDRESS-LINE := 2
EMPLOYEE.ADDRESS-LINE(1) := "8603 GARLAND COURT"
EMPLOYEE.ADDRESS-LINE(2) := "FRAMINGHAM"
EMPLOYEE.ADDRESS-LINE(2) := "MA"
EMPLOYEE.CITY := "FRAMINGHAM"
EMPLOYEE.ZIP := "17010"
EMPLOYEE.COUNTRY := "USA"
*
EMPLOYEE.AREA-CODE := "617"
EMPLOYEE.PHONE := "210-4703"
*
EMPLOYEE.JOB-TITLE := "MANAGER"
EMPLOYEE.CÂ$INCOME := 2
EMPLOYEE.SALARY(1) := 47000
EMPLOYEE.CÂ$BONUS(1) := 2
EMPLOYEE.BONUS(1,1) := 10500
EMPLOYEE.BONUS(1,2) := 7875
*
EMPLOYEE.SALARY(2) := 47000
EMPLOYEE.CÂ$BONUS(2) := 1
EMPLOYEE.BONUS(2,1) := 35700
*
INCLUDE EMPL-C "XML" "#CX" "#CY" "#CZ"
/* add generated Serialize
/*]
*
ASSIGN FILENUMBER = 15 /* PC FILE
ASSIGN OPERATION = 'W'
CALLNAT 'XML2PCWR' XML FILENUMBER OPERATION RETCODE
PRINT XML
DEFINE WORK FILE 10 TYPE 'UNFORMATTED' /* STD WORK FILE
WRITE WORK FILE 10 VARIABLE XML
CLOSE WORK FILE 10
*
END
```

出力ファイル

XML ツールキットは Report 2 に書き込みます。

アプリケーションの起動

XML ツールキットはライブラリ SYSEXXT に含まれています。

▶手順 70.1. XML ツールキットを使用するには

- Natural コマンド行で、「LOGON SYSEXXT」と入力します。

「Menu」と入力します。

メインメニューが表示されます。

```
10:22:54          *** NATURAL XML Toolkit ***          2007-01-19
                  - Main Menu -                          Library SYSEXXT

Code  Function

      L  Generate from Natural Data Structure
      D  Generate from Document Type Definition
      O  Set up Specific Generation Options

Function Code .. _

Press PF4 or PF5 to start generation.
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit  LDA   DTD                                Optio      Canc
```

次の機能を使用できます。

- **Natural** データ構造からの生成：Natural データエリアをデータソースとして使用します。
- **DTD**（文書型定義）から生成：文書型定義（.dtd）をデータソースとして使用します。
- 特定の生成オプションの設定

詳細については、対応するセクションを参照してください。

PF キーの割り当て

次のファンクションキーをナビゲーションおよび処理に使用できます。

PF1	Help	コンテキスト関連ヘルプ。詳細については、オンラインドキュメントを参照してください。
PF3	Exit	オプションマップ上：変更を保存して機能を終了します。 生成マップ上：生成を完了してから機能を終了します。 メインメニューマップ上：アプリケーションを終了します。
PF7	Prev	前の手順（前のマップ）。
PF8	Next	次の手順（次のマップ）。
PF9	Finis(h)	生成を完了してから機能を終了します。
PF12	Cancel	変更を保存せずに機能を終了します。 まだ生成が行われていない場合は、機能を終了します。

71 特定の生成オプションの設定

■ 生成オプション設定画面の起動	210
■ 最初の画面	210
■ 2 番目の画面	212
■ オプションの永続的な保存	214

生成オプションは、2つの画面に配置され、データフィールドとパス定義に分類されています。以下では次のトピックについて説明します。

生成オプション設定画面の起動

▶手順 71.1. 生成オプション機能の最初の画面を開くには

- メインメニュー画面で、PF10 **Optio** (n) キーを押します。

オプション設定機能の最初の画面が表示されます。マップフィールドについては、以下で説明します。

最初の画面

XMLで無効な特殊文字は、有効な名前に変換する必要があります。次のマップで、デフォルトの変換設定を必要に応じて変更できます。

```
09:51:35                *** NATURAL XML Toolkit ***                2007-01-17
User DEFAULT            - Options -                                Library SYSEXXT

Additional fields
Counter separator character ..... @

XML name replacements
Namespace separator character ':' with .. $
Dot sign '.' with ..... /

Natural variable name replacements
Plus sign '+' with ..... plus_____
Hash / Number sign '#' with ..... hash_____
Slash sign '/' with ..... slash_____
At sign '@' with ..... at_____
Paragraph sign 'Â$' with ..... paragraph__
Ampersand sign '&' with ..... ampersand__
Dollar sign '$' with ..... dollar_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help       Exit                               Next                               Canc
```


フィールドの説明

Counter Separator Character

所属グループ:	Additional Fields
フォーマット/長さ:	A1
デフォルト値:	@

Namespace Separator Character ":" WITH:

所属グループ:	XML Name Replacements
フォーマット/長さ:	A1
デフォルト値:	\$

Dot Sign '.' WITH:

所属グループ:	XML Name Replacements
フォーマット/長さ:	A1
デフォルト値:	/

Plus Sign '+' WITH:

所属グループ:	XML Variable Name Replacements
フォーマット/長さ:	A11
デフォルト値:	plus

Hash / Number Sign '#' WITH:

所属グループ:	Natural Variable Name Replacements
フォーマット/長さ:	A11
デフォルト値:	hash

Slash Sign '/' WITH:

所属グループ:	Natural Variable Name Replacements
フォーマット/長さ:	A11
デフォルト値:	slash

At Sign '@' WITH:

所属グループ:	Natural Variable Name Replacements
フォーマット/長さ:	A11
デフォルト値:	at

Paragraph Sign '\$' WITH:

所属グループ:	Natural Variable Name Replacements
フォーマット/長さ:	A11
デフォルト値:	para

Ampersand Sign '&' WITH:

所属グループ:	Natural Variable Name Replacements
フォーマット/長さ:	A11
デフォルト値:	amp

Dollar Sign '\$' WITH:

所属グループ:	Natural Variable Name Replacements
フォーマット/長さ:	A11
デフォルト値:	dollar

2 番目の画面

オプション機能の 2 番目のマップでは、変換に使用するターゲットまたはソース DTD ファイルの場所を定義します。

▶手順 71.2. 生成オプション機能の 2 番目の画面を開くには

- 1 メインメニュー画面で、PF10 **Optio** (n) キーを押します。

- 2 最初の画面で、PF8 **Next** を押します。

オプション機能の 2 番目の画面が表示されます。マップフィールドについては、以下で説明します。

```

11:04:10          *** NATURAL XML Toolkit ***          2007-01-17
User DEFAULT          - Options -          Library SYSEXXT

External file:
/nat_64/proj/natc/42/samples/sysexxt/empl.dtd_____

Natural library:
SYSEXXT_

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit          Prev          Canc
    
```

フィールドの説明

External file

フォーマット／長さ：	A253
デフォルト値：	(現在のライブラリのリソースディレクトリ)

Natural library

フォーマット／長さ：	A8
デフォルト値：	(現在のライブラリ)

オプションの永続的な保存

このオプションメニューのすべての設定は、テキストメンバ XML-INI に書き込まれます。

新しい Natural パッチレベルまたはサービスパックをインストールすると、オプションメニューのすべての設定が上書きされます。

設定を永続的に保持するには、パッチレベルまたはサービスパックをインストールする前に、テキストメンバ XML-INI を FUSER に保存することをお勧めします。

72 Natural データソースの使用

▪ Natural データエリアの選択	216
▪ ルートグループの選択	217
▪ DTD 定義でのファイルの生成	218
▪ XML ドキュメントのパースの生成	219
▪ 生成レポートの表示	220

この機能を使用すると、Naturalのローカル、グローバル、またはパラメータデータエリアに保持されているデータ定義からXMLドキュメントを生成できます。

次のトピックについて説明します。

以下の項目も参照してください。

- [外部データソースの使用](#)
- [特定の生成オプションの設定](#)
- [PF キーの割り当て](#)

Natural データエリアの選択

この画面では、LDA、GDA、またはPDAを入力データエリアとして選択します。



注意: 次のダイアログに表示されているエントリは、デフォルト値または例の値です。

```
08:07:48          ***** NATURAL XML Toolkit *****          2007-01-18
                - Generate from Natural Data Structure -          Library SYSEXXT
```

```
Select LDA, GDA or PDA as input Data Area.
```

```
Press 'Next' to read the Data Area.
```

```
Select a Level 1 group that should be used for further generations.
```

```
Select Data Area for generation
```

```
Library:  Type:  Name:
```

```
SYSEXXT_  L      _____
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
        Help      Exit                        Next                        Canc
```

フィールドの説明

Library

所属グループ:	Select Input Data Area
フォーマット/長さ:	A8
デフォルト値:	(すべてのライブラリ)

Type

所属グループ:	Select Input Data Area
フォーマット/長さ:	A1
設定可能値:	<p>L: ローカルデータエリア</p> <p>A: パラメータデータエリア</p> <p>G: グローバルデータエリア</p>

Name

所属グループ:	Select Input Data Area
フォーマット/長さ:	A8
デフォルト値:	(選択されたライブラリおよびタイプのすべてのオブジェクト)

PF8 キーを押して次に進みます。

ルートグループの選択

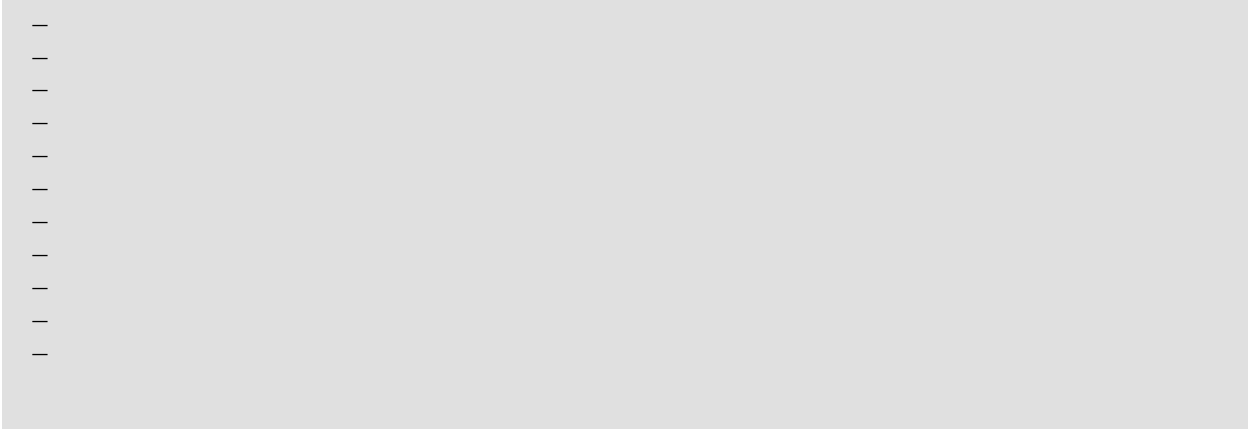
この画面では、データタイプを選択します。

```

10:18:34          *** NATURAL XML Toolkit ***          2007-01-18
                  - Select Root Group -                Library SYSEXXT

Root groups
X EMPLOYEE
-
-
-

```



フィールドの説明

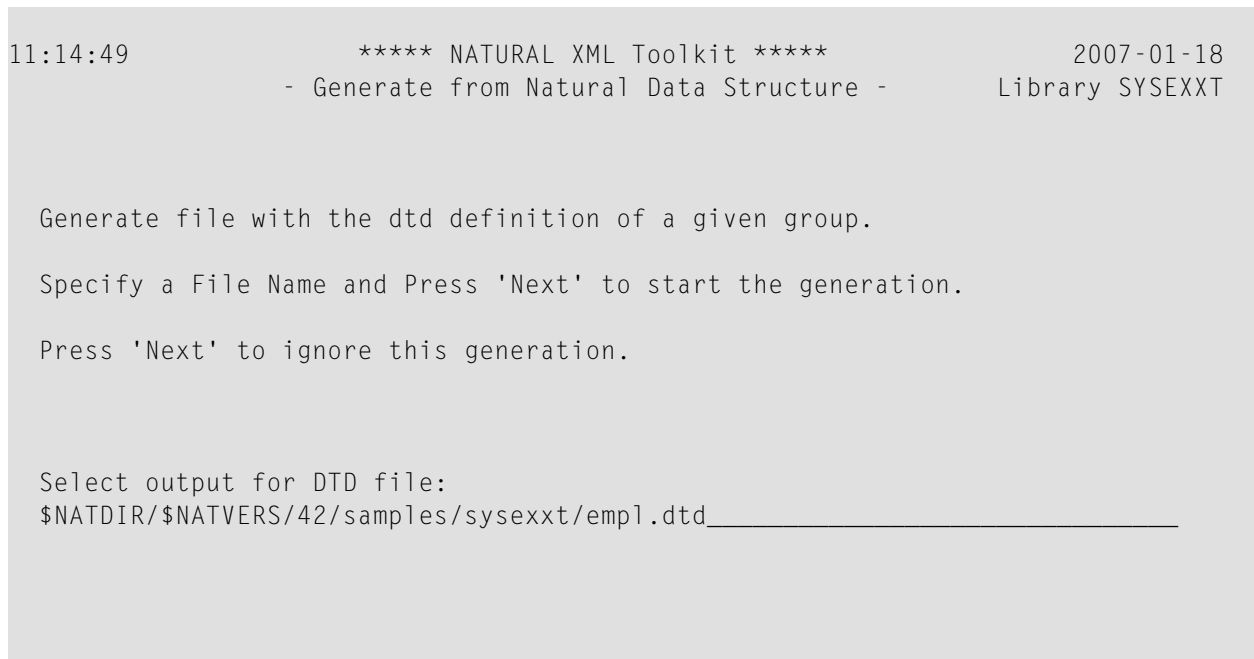
フォーマット/長さ:	A1
デフォルト値:	(レベル1のすべてのグループ)

目的の要素 (EMPLOYEE など) を **X** でマークし、Enter キーを押します。

PF8 を押して次に進みます。

DTD 定義でのファイルの生成

この画面では、指定したグループの DTD 定義でファイルを生成します。




```
Serialize Copycode generation done.
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit                      Prev  Next                      Canc
```

フォーマット/長さ: A253

PF8 を押して次に進みます。

XML ドキュメントのパーサーの生成

この画面を使用して、指定したグループをXMLドキュメントにシリアライズする実装としてコピーコードを生成します。

```
10:56:06          ***** NATURAL XML Toolkit *****          2007-01-18
                - Generate from Natural Data Structue -          Library SYSEXXT

Generate Copycode as implementation for the serialization of the given
group into a XML document.

Specify a Name and Press 'Next' to start the generation.

Press 'Next' to ignore this generation.

Select output for Serialize Copycode
Library:  Type: Name:
SYSEXXT_ C      _____

Parse Copycode generation done.
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit                      Prev  Next                      Canc
```

パーサーCALLBACKコピーコードを生成します。『例』ドキュメントの「[パーサーCALLBACKコピーコード](#)」も参照してください。

フィールドの説明

Library

所属グループ:	Select Output Copycode
フォーマット/長さ:	A8
デフォルト値:	(すべてのライブラリ)

Type

所属グループ:	Select Output Copycode
フォーマット/長さ:	A21
デフォルト値:	コピーコード

Name

所属グループ:	Select Output Copycode
フォーマット/長さ:	A8
デフォルト値:	(選択されたライブラリおよびタイプのすべてのオブジェクト)

PF8 を押して次に進みます。

生成レポートの表示

生成が完了すると、生成レポートが表示されます。

```
11:48:29          ***** NATURAL XML Toolkit *****          2007-01-18
                - Generate from Natural Data Structure -          Library SYSEXXT

Generation Results
Generate for Data Area
  Library ...: SYSEXXT
  Object ...: EMPL
  Read Data Area done.
Parser (Callback) Copycode
  Library ...: SYSEXXT
  Source ...: A1
  Parse Copycode generation done.
Serialize (Compress XML) Copycode
  Library ...: SYSEXXT
  Source ...: A2
```

```
Serialize Copycode generation done.  
File with dtd Definition  
File .....: $NATDIR/$NATVERS/42/samples/sysext/esi.dtd  
Generate DTD file.  
  
Generation done.  
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---  
      Help      Exit      Prev      Finis      Canc
```

フィールドの説明

要約

フォーマット/長さ: A253/1:v

PF9 **Finis** (h) をクリックして生成処理を終了します。

73 外部データソースの使用

▪ 文書型定義からの生成	224
▪ ルートエレメントの選択	225
▪ Natural データエリアの生成	226
▪ XML ドキュメントへのシリアライズ	227
▪ コピーコードの生成	229
▪ 生成結果の表示	230

外部データソースの使用

この機能を使用して、XML ドキュメントを解析して、ローカル、グローバル、またはパラメータデータエリアで定義されている Natural 変数に変換できます。

次のトピックについて説明します。

以下の項目も参照してください。

- **Natural** データソースの使用
- 特定の生成オプションの設定
- **PF** キーの割り当て

文書型定義からの生成

この画面は、入力文書型として文書型定義または Tamino スキーマを選択するために使用します。



注意: 次の画面に表示されているフィールドエントリは、デフォルト値または例の値です。

```
12:30:09          ***** NATURAL XML Toolkit *****          2007-01-19
                - Generate from Document Type Definition -          Library SYSEXXT

Select Document Type Definition/Tamino Schema as input Document Type.

Press 'Next' to read the Document Type.

Select a root element or Tamino Document Type that should be used
for further generations.

Select DTD/Tamino Schema for generation:
/nat_64/proj/natc/42/samples/sysexxt/empl.dtd_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help       Exit                               Next           Canc
```

フィールドの説明

Input File

フォーマット/長さ: A253

DTD の代わりに Tamino 2.1.x スキーマを使用します。

PF8 **Next** を押して次に進みます。

ルートエレメントの選択

この画面を使用して、XML ドキュメントのルートである要素を選択します。

```
12:42:17          *** NATURAL XML Toolkit ***          2007-01-19
                  - Select Root Element -          Library SYSEXXT

Element
X EMPLOYEE
_ FULL-NAME
_ FIRST-NAME
_ NAME
_ FULL-ADDRESS
_ ADDRESS-LINE
_ CITY
_ ZIP
_ COUNTRY
_ TELEPHONE
_ AREA-CODE
_ PHONE
_ JOB-TITLE
_ INCOME
_ SALARY
```

フィールドの説明

Root Element (DTD の場合)

デフォルト値: (すべての要素)

目的の要素 (EMPLOYEE など) を **X** でマークし、Enter キーを押します。

Natural データエリアの生成

ここでは、XML ドキュメントを表すグループの定義を使用して Natural データエリアを生成します。

```
13:25:40          ***** NATURAL XML Toolkit *****          2007-01-19
- Generate from Document Type Definition -          Library SYSEXXT
```

```
Generate Data Area with definition of a group that represents
the XML document.
```

```
Specify a Name and Press 'Next' to start the generation.
```

```
Press 'Next' to ignore this generation.
```

```
Select output LDA/GDA/PDA
```

```
Library: Type: Name:
```

```
SYSEXXT_ L      A3_____
```

```
Generate Data Area.
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit          Prev  Next          Canc
```


フィールドの説明

Library

所属グループ:	Select Output Copycode
フォーマット/長さ:	A8
デフォルト値:	(すべてのライブラリ)

Type

所属グループ:	Select Output Copycode
フォーマット/長さ:	A21
デフォルト値:	L: ローカルデータエリア

Name

所属グループ:	Select Output Copycode
フォーマット/長さ:	A8
デフォルト値:	(選択されたライブラリおよびタイプのすべてのオブジェクト)

次に進みます。

XML ドキュメントへのシリアライズ

この画面を使用して、指定したグループをXML ドキュメントにシリアライズする実装としてコピーコードを生成します。

```
13:10:40          ***** NATURAL XML Toolkit *****          2007-01-19
- Generate from Natural Data Structure -          Library SYSEXXT
```

```
Generate Copycode as implementation for the serialization of the given
group into a XML document.
```

```
Specify a Name and Press 'Next' to start the generation.
```

```
Press 'Next' to ignore this generation.
```

```
Select output for Serialize Copycode
Library: Type: Name:
```

SYSEXXT_ C A2_____

Parse Copycode generation done.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---

Help

Exit

Prev Next

Canc

『例』ドキュメントの「[コピーコードのシリアライズ](#)」も参照してください。

フィールドの説明

Library

所属グループ：	Select Output Copycode
フォーマット／長さ：	A8
デフォルト値：	(すべてのライブラリ)

Type

所属グループ：	Select Output Copycode
フォーマット／長さ：	A21
デフォルト値：	Copycode

Name

所属グループ：	Select Output Copycode
フォーマット／長さ：	A8
デフォルト値：	(選択されたライブラリおよびタイプのすべてのオブジェクト)

PF8 **Next** を押して次に進みます。

コピーコードの生成

この画面では、指定グループのXMLパーサーコールバックの実装としてコピーコードを生成します。

```

13:02:32          ***** NATURAL XML Toolkit *****          2007-01-19
          - Generate from Document Type Definition -          Library SYSEXXT

Generate Copycode as implementation for the XML Parser Callback for the
given group.

Specify a Name and Press 'Next' to start the generation.

Press 'Next' to ignore this generation.

Select output for Parse Copycode:
Library:  Type:  Name:
SYSEXXT  C

Read DTD/Tamino Schema done.
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
          Help          Exit          Prev  Next          Canc

```

パーサーCALLBACKコピーコードを生成します。『例』ドキュメントの「[パーサーCALLBACKコピーコード](#)」も参照してください。

フィールドの説明

Library

所属グループ:	Select Output Copycode
フォーマット/長さ:	A8
デフォルト値:	(すべてのライブラリ)

Type

所属グループ:	Select Output Copycode
フォーマット/長さ:	A1
デフォルト値:	コピーコード

Name

所属グループ:	Select Output Copycode
フォーマット/長さ:	A8
デフォルト値:	(選択されたライブラリおよびタイプのすべてのオブジェクト)

PF8 **Next** を押して次に進みます。

生成結果の表示

生成が完了すると、生成結果の要約が表示されます。

```
13:51:11          ***** NATURAL XML Toolkit *****          2007-01-19
                - Generate from Document Type Definition -          Library SYSEXXT

Generation Results
Generate for DTD/ino schema
  File .....: /nat_64/proj/natc/42/samples/sysexxt/empl.dtd
  Read DTD/Tamino Schema done.
Parser (Callback) Copycode
  Library ..: SYSEXXT
  Source ...: A1
  Parse Copycode generation done.
Serialize (Compress XML) Copycode
  Library ..: SYSEXXT
  Source ...: A2
  Serialize Copycode generation done.
Data Area
  Library ..: SYSEXXT
  Source ...: A3
  Data Area Generation done.

Generation done.
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit      Prev      Finis      Canc
```

フィールドの説明

要約

フォーマット/長さ: A253/1:v

PF9 **Finis** (h) をクリックして生成処理を終了します。

74 Natural のシンプルな XML パーサー

- パーサーの説明と例 234
- パーサーの制限事項 241

次のトピックについて説明します。

パーサーの説明と例

Natural のシンプルな XML パーサーでは、標準の Natural プログラムを使用して XML ドキュメントを解析できます。パーサーは、ドキュメントの次の部分が解析されると、イベントを送信するか、内部サブルーチンのコールバックを実行します。インラインサブルーチンの "CALLBACK" は、xpath 同様の構文に現在の要素、テキスト、コメントの名前を指定して呼び出されます。パーサーエンジンは、コピーコード "PARSER_X" として組み込まれます。解析中に、ドキュメント形式が無効などのエラーが発生すると、"PARSER_ERROR" インラインサブルーチンが呼び出され、パーサーは "ESCAPE SUBROUTINE" でキャンセルされます（「[パーサーの制限事項](#)」も参照）。

オペランド 6 のエラーメッセージテキストとオペランド 7 のエラー番号の値を -9000 以下に変更することによって、エラー処理を拡張できます。これにより、"PARSER_ERROR" インラインサブルーチンが呼び出され、（サブ）プログラムは "ESCAPE SUBROUTINE" でキャンセルされます。他の値が -8000 以下の場合、パーサーのみが "ESCAPE SUBROUTINE" でキャンセルされます。

パーサーの主な変数は、ローカルデータエリア "PARSER-X" で定義されます。

パーサーコピーコードでは、次のオペランドを使用します。

オペランド	フォーマット／長さ	説明
1	A	解析される XML ファイル
2	A	要素構造を表す ex-XPATH
3	A1	XPATH の内容のタイプ： ? 処理命令 D DOCTYPE ! コメント C CDATA セクション T 開始タグ @ 属性 / 終了タグ
4	A	解析されたデータ
5	L	解析されたデータが空の場合、TRUE
6	A	エラーメッセージのテキスト
7	I4	エラー番号

XPATH データの戻り値：

ex-Xpath	XML 構造
?	<? ... ?>
!DOCTYPE	<!DOCTYPE ... >
!DOCTYPE[<!DOCTYPE .. [...]>
![CDATA[<![CDATA[...]]>
!--	<!-- -->
!	<! .. >
doc	<doc>
doc doc/foo doc/foo/\$ doc/foo// doc//	<doc><foo>text</foo></doc>
doc doc/@a1 doc//	<doc a1="a" />
doc doc/@a1 doc/@a2 doc/\$ doc//	<doc a1="a" a2="b">text</doc>
doc doc/\$ doc/foo doc/foo/\$ doc/foo// doc/\$ doc//	<doc> <foo>text</foo> </doc>
doc doc/![CDATA[doc//	<doc><![CDATA[...]]></doc>
doc doc/!-- doc//	<doc><!-- ... --></doc>

プログラム例：

```

* -----
* CLASS NATURAL XML TOOLKIT - UTILITIES
*
*     PARSE
*
* DESCRIPTION
*     Parse given XML
*
*
* AUTHOR      SAG    01.2006
*
* VERSION     6.2.
*
* (c) Copyright Software AG 2006. All rights reserved.
*
* -----
*
DEFINE DATA LOCAL
1 XML_PARSER_INPUT           (A) DYNAMIC
1 XML_PARSER_ERROR_TEXT     (A253)
1 XML_PARSER_RESPONSE       (I4)
LOCAL USING PARSE-X         /* parser internal data - do not change
LOCAL
1 XML_PARSER_XPATH          (A) DYNAMIC
1 XML_PARSER_XPATH_TYPE     (A1)
1 XML_PARSER_CONTENT        (A) DYNAMIC
1 XML_PARSER_CONTENT_IS_EMPTY (L)
*
1 ANFANG                    (T)
* OUT                        (A) DYNAMIC
1 OUT                        (A126)
*
END-DEFINE
*
FORMAT (0) LS=128 PS=40
*
DEFINE WORK FILE 12 "E:\EMPLOYEE1.XML" TYPE "UNFORMATTED"
READ WORK FILE 12 XML_PARSER_INPUT
END-WORK
CLOSE WORK FILE 12
*
*
* ----- INCLUDE THE PARSE
INCLUDE PARSE_X 'XML_PARSER_INPUT' /* XML file to be parsed
'XML_PARSER_XPATH' /* XPATH to represent element...
'XML_PARSER_XPATH_TYPE' /* Type of callback
'XML_PARSER_CONTENT' /* Content of element found
'XML_PARSER_CONTENT_IS_EMPTY' /* Is TRUE if element is empty

```

```

XML_PARSER_ERROR_TEXT'          /* error Message
XML_PARSER_RESPONSE'            /* Error NR; 0 = OK
*
*
DEFINE SUBROUTINE CALLBACK
IF XML_PARSER_CONTENT_IS_EMPTY THEN
  IF XML_PARSER_XPATH_TYPE NE "T" AND XML_PARSER_XPATH_TYPE NE "/" THEN
    COMPRESS XML_PARSER_XPATH "(NULL)" INTO OUT WITH DELIMITER "="
  ELSE
    OUT := XML_PARSER_XPATH
  END-IF
ELSE
  COMPRESS XML_PARSER_XPATH XML_PARSER_CONTENT INTO OUT WITH DELIMITER "="
END-IF
WRITE OUT
END-SUBROUTINE
/*
DEFINE SUBROUTINE PARSER_ERROR
OUT := XML_PARSER_ERROR_TEXT
WRITE OUT
END-SUBROUTINE
END

```

Employee データに対する Tamino からの結果ドキュメントでは、このプログラムの結果は次のようになります。

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<Employee xmlns:ino="http://namespaces.softwareag.com/tamino/response2" ino:id="560"
Personnel-ID="20006900">
<Full-Name>
<First-Name>JOE</First-Name>
<Name>ATHERTON</Name>
</Full-Name>
<Mar-Stat>S</Mar-Stat>
<Sex>M</Sex>
<Birth>1941-02-21</Birth>
<Full-Address>
<Address-Line>11603 HUNTERS GREEN</Address-Line>
<Address-Line>SYRACUSE</Address-Line>
<Address-Line>NY</Address-Line>
<City>SYRACUSE</City>
<Zip>13201</Zip>
<Post-Code>13201</Post-Code>
<Country>USA</Country>
</Full-Address>
<Telephone>
<Phone>173-9859</Phone>
<Area-Code>315</Area-Code>
</Telephone>
<Dept>TECH10</Dept>

```

```
<Job-Title>ANALYST</Job-Title>
<Income>
<Curr-Code>USD</Curr-Code>
<Salary>43000</Salary>
</Income>
<Income>
<Curr-Code>USD</Curr-Code>
<Salary>39500</Salary>
</Income>
<Income>
<Curr-Code>USD</Curr-Code>
<Salary>36700</Salary>
</Income>
<Income>
<Curr-Code>USD</Curr-Code>
<Salary>34400</Salary>
</Income>
<Income>
<Curr-Code>USD</Curr-Code>
<Salary>32600</Salary>
</Income>
<Leave-Data>
<Leave-Due>19</Leave-Due>
<Leave-Taken>4</Leave-Taken>
</Leave-Data>
<Leave-Booked>
<Leave-Start>19980112</Leave-Start>
<Leave-End>19980112</Leave-End>
</Leave-Booked>
<Leave-Booked>
<Leave-Start>19980605</Leave-Start>
<Leave-End>19980605</Leave-End>
</Leave-Booked>
<Leave-Booked>
<Leave-Start>19980916</Leave-Start>
<Leave-End>19980916</Leave-End>
</Leave-Booked>
<Lang>ENG</Lang>
</Employee>
```



注意: ドキュメント全体に改行はありません。

上記の Natural プログラムの結果は次のようになります。

```
?=xml version="1.0" encoding="ISO-8859-1"
Employee
Employee/@xmlns:ino=http://namespaces.softwareag.com/tamino/response2
Employee/@ino:id=560
Employee/@Personnel-ID=20006900
Employee/Full-Name
Employee/Full-Name/First-Name
Employee/Full-Name/First-Name/$=JOE
Employee/Full-Name/First-Name//
Employee/Full-Name/Name
Employee/Full-Name/Name/$=ATHERTON
Employee/Full-Name/Name//
Employee/Full-Name//
Employee/Mar-Stat
Employee/Mar-Stat/$=S
Employee/Mar-Stat//
Employee/Sex
Employee/Sex/$=M
Employee/Sex//
Employee/Birth
Employee/Birth/$=1941-02-21
Employee/Birth//
Employee/Full-Address
Employee/Full-Address/Address-Line
Employee/Full-Address/Address-Line/$=11603 HUNTERS GREEN
Employee/Full-Address/Address-Line//
Employee/Full-Address/Address-Line
Employee/Full-Address/Address-Line/$=SYRACUSE
Employee/Full-Address/Address-Line//
Employee/Full-Address/Address-Line
Employee/Full-Address/Address-Line/$=NY
Employee/Full-Address/Address-Line//
Employee/Full-Address/City
Employee/Full-Address/City/$=SYRACUSE
Employee/Full-Address/City//
Employee/Full-Address/Zip
Employee/Full-Address/Zip/$=13201
Employee/Full-Address/Zip//
Employee/Full-Address/Post-Code
Employee/Full-Address/Post-Code/$=13201
Employee/Full-Address/Post-Code//
Employee/Full-Address/Country
Employee/Full-Address/Country/$=USA
Employee/Full-Address/Country//
Employee/Full-Address//
Employee/Telephone
Employee/Telephone/Phone
Employee/Telephone/Phone/$=173-9859
```

```
Employee/Telephone/Phone//
Employee/Telephone/Area-Code
Employee/Telephone/Area-Code/=$=315
Employee/Telephone/Area-Code//
Employee/Telephone//
Employee/Dept
Employee/Dept/=$=TECH10
Employee/Dept//
Employee/Job-Title
Employee/Job-Title/=$=ANALYST
Employee/Job-Title//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=43000
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=39500
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=36700
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=34400
Employee/Income/Salary//
Employee/Income//
Employee/Income
Employee/Income/Curr-Code
Employee/Income/Curr-Code/=$=USD
Employee/Income/Curr-Code//
Employee/Income/Salary
Employee/Income/Salary/=$=32600
Employee/Income/Salary//
Employee/Income//
Employee/Leave-Data
```

```
Employee/Leave-Data/Leave-Due
Employee/Leave-Data/Leave-Due/$=19
Employee/Leave-Data/Leave-Due//
Employee/Leave-Data/Leave-Taken
Employee/Leave-Data/Leave-Taken/$=4
Employee/Leave-Data/Leave-Taken//
Employee/Leave-Data//
Employee/Leave-Booked
Employee/Leave-Booked/Leave-Start
Employee/Leave-Booked/Leave-Start/$=19980112
Employee/Leave-Booked/Leave-Start//
Employee/Leave-Booked/Leave-End
Employee/Leave-Booked/Leave-End/$=19980112
Employee/Leave-Booked/Leave-End//
Employee/Leave-Booked//
Employee/Leave-Booked
Employee/Leave-Booked/Leave-Start
Employee/Leave-Booked/Leave-Start/$=19980605
Employee/Leave-Booked/Leave-Start//
Employee/Leave-Booked/Leave-End
Employee/Leave-Booked/Leave-End/$=19980605
Employee/Leave-Booked/Leave-End//
Employee/Leave-Booked//
Employee/Leave-Booked
Employee/Leave-Booked/Leave-Start
Employee/Leave-Booked/Leave-Start/$=19980916
Employee/Leave-Booked/Leave-Start//
Employee/Leave-Booked/Leave-End
Employee/Leave-Booked/Leave-End/$=19980916
Employee/Leave-Booked/Leave-End//
Employee/Leave-Booked//
Employee/Lang
Employee/Lang/$=ENG
Employee/Lang//
Employee//
```

パーサーの制限事項

パーサーでは、以下は処理されません。

- タグの合成（処理命令を含む）。開始タグのみは終了タグと一致する必要があります（処理命令を含む）。

例：

```
<.doc></.doc> <!-- invalid character in tag -->
<doc><? ?></doc> <!-- invalid whitespace -->
<doc>&#RE;</doc> <!-- invalid character in tag -->
```

■ 文字またはエンティティの参照

例：

```
<doc>& no refc</doc> <!-- missing semicolon --> <doc a1=v1></doc>
<!-- string literal expected -->
```

■ CDATA セクションの正確な処理

例：

```
<doc><![CDATA [ stuff]]></doc> <!-- must be CDATA[ -->
```

■ エンティティ／処理命令の内容

例：

```
<doc>]]></doc> <!-- ]] not allowed -->
```

■ タグ／属性の数

■ ヘッダー情報

■ Unicode 文字セット（ISO-8859-1 をサポート）

75 例

■ コピーコードのシリアライズ	244
■ 生成された Natural データエリア	246
■ Natural DTD パーサー	248
■ 生成されたタイプ定義	249
■ パーサー CALLBACK コピーコード	250

次の例が含まれています。

コピーコードのシリアライズ

XML ツールキットを使用して、Natural グループ構造を XML ドキュメントに変換するコピーコードを生成できます。

コールバックコピーコードでは、次のオペランドを使用します。

オペランド	フォーマット/長さ	説明	PARSER-X から
1	A	要素構造を表す ex-XPATH	オペランド 2
2	A1	XPATH の内容のタイプ： ? D DOCTYPE ! C CDATA セクション T 開始タグ @ 属性 / 終了タグ	オペランド 3
3	A	解析されたデータ	オペランド 4
4	L	解析されたデータが空の場合、TRUE	オペランド 5
5	I4	カウンタ変数 1 次元	
6	I4	カウンタ変数 2 次元	
7	I4	カウンタ変数 3 次元	

コピーコード **EMPL-C** の例：

```

*
----- * Parameter
Definition * * &1& 'XML' /* XML Document * &2& '#CX' /* Counter
Variable 1st Dimension * &3& '#CY' /* Counter Variable 2nd Dimension *
&4& '#CZ' /* Counter Variable 3rd Dimension *
-----
* DTD E-\SAG\nat\NATAPPS\FUSER\XMLTK\RES\empl.dtd COMPRESS &1& '<EMPLOYEE'
' PERSONNEL-ID="'EMPLOYEE.PERSONNEL-ID '' '>' INTO &1& LEAVING NO
/* now the children COMPRESS &1& '<FULL-NAME' '>' INTO &1&
LEAVING NO /* now the children COMPRESS &1& '<FIRST-NAME' '>' EMPLOYEE.FIRST-NAME
'</FIRST-NAME>' INTO &1& LEAVING NO COMPRESS &1& '<NAME'
'>' EMPLOYEE.NAME '</NAME>' INTO &1& LEAVING NO /* COMPRESS &1&
'</FULL-NAME>' INTO &1& LEAVING NO COMPRESS &1& '<FULL-ADDRESS'

```

```
'>' INTO &1& LEAVING NO /* now the children FOR &2& = 1 TO
EMPLOYEE.C@ADDRESS-LINE COMPRESS &1& '<ADDRESS-LINE' '>' EMPLOYEE.ADDRESS-LINE(&2&)
'</ADDRESS-LINE>' INTO &1& LEAVING NO END-FOR COMPRESS &1&
'<CITY' '>' EMPLOYEE.CITY '</CITY>' INTO &1& LEAVING NO COMPRESS
&1& '<ZIP' '>' EMPLOYEE.ZIP '</ZIP>' INTO &1& LEAVING
NO COMPRESS &1& '<COUNTRY' '>' EMPLOYEE.COUNTRY '</COUNTRY>'
INTO &1& LEAVING NO /* COMPRESS &1& '</FULL-ADDRESS>' INTO
&1& LEAVING NO COMPRESS &1& '<TELEPHONE' '>' INTO &1&
LEAVING NO /* now the children COMPRESS &1& '<PHONE' '>' EMPLOYEE.PHONE
'</PHONE>' INTO &1& LEAVING NO COMPRESS &1& '<AREA-CODE'
'>' EMPLOYEE.AREA-CODE '</AREA-CODE>' INTO &1& LEAVING NO /*
COMPRESS &1& '</TELEPHONE>' INTO &1& LEAVING NO COMPRESS
&1& '<JOB-TITLE' '>' EMPLOYEE.JOB-TITLE '</JOB-TITLE>' INTO
&1& LEAVING NO FOR &2& = 1 TO EMPLOYEE.C@INCOME COMPRESS &1&
'<INCOME' '>' INTO &1& LEAVING NO /* now the children COMPRESS &1&
'<SALARY' '>' EMPLOYEE.SALARY(&2&) '</SALARY>' INTO &1&
LEAVING NO FOR &3& = 1 TO EMPLOYEE.C@BONUS(&2&) COMPRESS &1&
'<BONUS' '>' EMPLOYEE.BONUS(&2&,&3&) '</BONUS>' INTO
&1& LEAVING NO END-FOR /* COMPRESS &1& '</INCOME>' INTO
&1& LEAVING NO END-FOR /* COMPRESS &1& '</EMPLOYEE>' INTO
&1& LEAVING NO
```

プログラム例：

```
*
----- * CLASS
NATURAL XML TOOLKIT * * * DESCRIPTION * Serialize a given Data structure. * *
* AUTHOR SAG 01.2006 * * VERSION 6.2. * * (c) Copyright Software AG 2006. All
rights reserved. * *
-----
* DEFINE DATA
LOCAL USING EMPL /* add generated data structure LOCAL 1
XML (A) DYNAMIC * 1 OUT (A72) 1 II (I4) * 1 OUTDYN (A) DYNAMIC 1 OBJLEN (I4) 1
OBJEND (I4) 1 OBJSTART (I4) 1 OBJLINE (I4) * 1 #CX (I4) 1 #CY (I4) 1 #CZ (I4)
END-DEFINE * EMPLOYEE.PERSONNEL-ID := 4711 * EMPLOYEE.FIRST-NAME := "ADKINSON"
EMPLOYEE.NAME := "MARTHA" * EMPLOYEE.C@ADDRESS-LINE := 2 EMPLOYEE.ADDRESS-LINE(1)
:= "8603 GARLAND COURT" EMPLOYEE.ADDRESS-LINE(2) := "FRAMINGHAM"
EMPLOYEE.ADDRESS-LINE(2)
:= "MA" EMPLOYEE.CITY := "FRAMINGHAM" EMPLOYEE.ZIP := "17010" EMPLOYEE.COUNTRY
:= "USA" * EMPLOYEE.AREA-CODE := "617" EMPLOYEE.PHONE := "210-4703" *
EMPLOYEE.JOB-TITLE
:= "MANAGER" EMPLOYEE.C@INCOME := 2 EMPLOYEE.SALARY(1) := 47000 EMPLOYEE.C@BONUS(1)
:= 2 EMPLOYEE.BONUS(1,1) := 10500 EMPLOYEE.BONUS(1,2) := 7875 * EMPLOYEE.SALARY(2)
:= 47000 EMPLOYEE.C@BONUS(2) := 1 EMPLOYEE.BONUS(2,1) := 35700 *
INCLUDE EMPL-C
"XML" "#CX" "#CY" "#CZ" /* add generated Serialize * FOR II = 1 TO *LENGTH(XML)
STEP 72 OUT := SUBSTR(XML,II) WRITE OUT END-FOR * NEWPAGE * /* WRITE COMPLETE
(A) DYNAMIC VARIABLE IF POSSIBLE USE CR AND IGNORE LF OBJSTART := 1 * EXAMINE
xml FOR "><" REPLACE WITH ">" - H'0A' - "<" EXAMINE xml FOR H'0A' GIVING
POSITION OBJEND * REPEAT WHILE OBJEND NE 0 /* IF OBJSTART GT 0 THEN ADD OBJSTART
TO OBJEND END-IF /* OBJLEN := OBJEND - OBJSTART -1 /* IF OBJLEN > 0 THEN OUTDYN
```

```
:= SUBSTRING(xml, OBJSTART, OBJLEN) /* FOR OBJLINE = 1 TO *LENGTH(OUTDYN) STEP
72 OUT := SUBSTR (OUTDYN,OBJLINE) WRITE OUT END-FOR ELSE WRITE " " END-IF /* OBJSTART
:= OBJEND IF OBJSTART GT *LENGTH(xml) ESCAPE BOTTOM END-IF /* EXAMINE
SUBSTRING(xml,OBJSTART)
FOR H'OA' GIVING POSITION OBJEND END-REPEAT * END
```

使用した **Natural PDA EMPL** :

```
DEFINE DATA PARAMETER 1 EMPLOYEE 2 ATTRIBUTES_OF_EMPLOYEE
3 PERSONNEL-ID(A8) * 2 FULL-NAME 3 FIRST-NAME(A20) 3 NAME(A20) * 2 FULL-ADDRESS
3 C@ADDRESS-LINE(I4) 3 ADDRESS-LINE(A20/1:6) 3 CITY(A20) 3 ZIP(A20) 3 COUNTRY(A3)
* 2 TELEPHONE 3 AREA-CODE(A6) 3 PHONE(A15) * 2 JOB-TITLE(A25) * 2 C@INCOME(I4)
2 INCOME(1:6) 3 SALARY(A9) 3 C@BONUS(I4) 3 BONUS(A9/1:4) END-DEFINE
```

生成された Natural データエリア

XML ツールキットを使用して、指定の文書型定義を表す Natural データエリア（より正確にはローカルデータエリア、パラメータデータエリア、またはグローバルデータエリア）を生成できます。

生成ルール :

- 属性を持たない空エレメント (<!ELEMENT br EMPTY>) は、タイプ B1 の Natural 変数として生成されます。空の Natural グループは許可されないため、このことが必要となります。
- 属性を持つ空エレメント (<!ELEMENT br EMPTY><!ATTLIST br width CDATA #IMPLIED>) は、Natural グループとして生成されます。
- 内容を持つ各エレメント (<!ELEMENT b (#PCDATA)>) は、タイプ A253 の Natural 変数として生成されます。
- エレメントの連続 (<!ELEMENT spec (front, body*, back?)>) またはエレメントの選択 (<!ELEMENT div1 (p | list | note)>) は、Natural グループとして生成されます。
- 連続または選択が組み合わされている場合 (<!ELEMENT address ((street, housenumber), (zip, city))>) は、名前の接頭辞 "##PSEUDO" が付いた特殊なグループとして生成されます。これにより、コンテキストまたは可能となる多重度を表現することができます。
- エレメントの各属性 (<!ATTLIST br width CDATA #IMPLIED>) は、エレメント名に接頭辞 "ATTRIBUTES_OF_" が付いたグループに属する、タイプ A253 の変数として生成されます。
- 複数のエレメントは、常に次元 1:v の配列として生成されます。生成される配列の上限は、手動で変更する必要があります。
- 1つのエレメントが複数回定義されている場合 (<!ELEMENT spec (front, body*)>) は、使用可能なエレメントの数を指定するために、追加のカウンタフィールド C@BODY が生成されません。

- Natural 名では大文字と小文字が区別されないため、DTD 内で使用される名前はすべて大文字に変換されます。生成されたグループ内で重複している名前には、名前が一意になるように接尾辞が付加されます。
- Natural 名として無効な特殊文字は変換されて有効な Natural 名になります。変換の設定については、XML ツールキットのオプションダイアログに関する説明を参照してください。

制限事項：

- 混在する内容データを持つエレメント（<!ELEMENT p (#PCDATA | a | ul | b | i | em)*>）はサポートされません。
- Natural は最大 3 次元のデータ構造のみをサポートするので、Natural データ構造に変換される DTD は Natural 内では使用できません。

DTD の例：

```
<!ELEMENT EMPLOYEE (FULL-NAME , FULL-ADDRESS , TELEPHONE ,JOB-TITLE, INCOME* )>
<!ATTLIST EMPLOYEE PERSONNEL-ID CDATA #REQUIRED >

<!ELEMENT FULL-NAME (FIRST-NAME , NAME )>
<!ELEMENT FIRST-NAME (#PCDATA )>
<!ELEMENT NAME (#PCDATA )>

<!ELEMENT FULL-ADDRESS (ADDRESS-LINE* , CITY , ZIP , COUNTRY )>
<!ELEMENT ADDRESS-LINE (#PCDATA )>
<!ELEMENT CITY (#PCDATA )>
<!ELEMENT ZIP (#PCDATA )>
<!ELEMENT COUNTRY (#PCDATA )>

<!ELEMENT TELEPHONE (PHONE , AREA-CODE )>
<!ELEMENT PHONE (#PCDATA )>
<!ELEMENT AREA-CODE (#PCDATA )>

<!ELEMENT JOB-TITLE (#PCDATA )>

<!ELEMENT INCOME (SALARY , BONUS* )>
<!ELEMENT SALARY (#PCDATA )>
<!ELEMENT BONUS (#PCDATA )>
```

例

生成された Natural データエリア (DTD の斜体部分。Natural に必須) :

```
DEFINE DATA PARAMETER
1 EMPLOYEE
  2 ATTRIBUTES_OF_EMPLOYEE
    3 PERSONNEL-ID(A253)
  *
  2 FULL-NAME
    3 FIRST-NAME(A253)
    3 NAME(A253)
  *
  2 FULL-ADDRESS
    3 C@ADDRESS-LINE(I4)
    3 ADDRESS-LINE(A253/1:v)
    3 CITY(A253)
    3 ZIP(A253)
    3 COUNTRY(A253)
  *
  2 TELEPHONE
    3 AREA-CODE(A253)
    3 PHONE(A253)
  *
  2 JOB-TITLE(A253)
  *
  2 C@INCOME(I4)
  2 INCOME(1:v)
    3 SALARY(A253)
    3 C@BONUS(I4)
    3 BONUS(A253/1:v)
END-DEFINE
```

Natural DTD パーサー

変換ルール :

Natural	文書型定義
1 G1 2 E1 (A&€!)	<!ELEMENT G1 (E1)> <!ELEMENT E1 (#PCDATA)>
1 G1 2 E1 (A&€!) 2 E2 (A&€!) 2 E3 (A&€!)	<!ELEMENT G1 (E1, E2, E3)> <!ELEMENT E1 (#PCDATA)> <!ELEMENT E2 (#PCDATA)> <!ELEMENT E3 (#PCDATA)>
1 C@E1_MAX (I4) CONST <10> 1 G1 2 C@E1 (I4)	<!ELEMENT G1 (E1*)> <!ELEMENT E1 (#PCDATA)>

Natural	文書型定義
2 E1 (A⟩/1:C@E1_MAX)	
1 C@E1_MAX (I4) CONST <10> 1 G1 2 C@E1 (I4) 2 E1 (A⟩/1:C@E1_MAX)	<!ELEMENT G1 (E1+)> <!ELEMENT E1 (#PCDATA)>
1 G1 2 E1 (A⟩)	<!ELEMENT G1 (E1?)> <!ELEMENT E1 (#PCDATA)>
1 G1 2 E1 (A⟩) 2 E2 (A⟩) 2 E3 (A⟩)	<!ELEMENT G1 (E1 E2 E3)> <!ELEMENT E1 (#PCDATA)> <!ELEMENT E2 (#PCDATA)> <!ELEMENT E3 (#PCDATA)>
1 G1 2 E1 (A⟩) 2 E2 (A⟩) 2 G2 2 E1_2 (A⟩) 2 E3 (A⟩)	<!ELEMENT G1 (E1, E2, G2)> <!ELEMENT E1 (#PCDATA)> <!ELEMENT E2 (#PCDATA)> <!ELEMENT G2 (E1, E3)> <!ELEMENT E3 (#PCDATA)>
1 #G1 2 #E1 (A⟩)	<!ELEMENT G1 (E1)> <!ELEMENT E1 (#PCDATA)>
2 E1 (A⟩) 3 ATTRIBUTES_OF_E1 4 A1 (A⟩) CONST <'schema'> 4 A2 (A⟩) 4 A3 (A⟩)	<!ELEMENT E1 (#PCDATA)> <!ATTLIST E1 A1 #FIXED "schema" A2 NMTOKEN #IMPLIED A3 ID #REQUIRED>

生成されたタイプ定義

XML ツールキットを使用して、Natural データエリア（より正確にはローカルデータエリア、パラメータデータエリア、またはグローバルデータエリア）で文書型定義を生成できます。

生成ルール：

- Natural 変数は、内容を持ったエレメントになります。
- Natural グループは、エレメントの連続になります。
- 複数の変数またはグループは、"0 以上" の多重度で生成されます。
- XML 名として無効な特殊文字は変換されて有効な名前になります。変換設定については、XML ツールキットの[オプション画面](#)を参照してください。

Natural データエリアの例：

```
DEFINE DATA LOCAL
1 NAT$EMPLOYEE
  2 ATTRIBUTES_OF_NAT$EMPLOYEE
    3 PERSONNEL/ID(A8)
  2 C@MAN@WORK(I4)
  2 MAN@WORK
    3 JOB(A10)
  2 A$TEST$MAKL(I4)
  2 AS/FA/SD(P7.5)
  2 #ASDFAS(F4)
  2 ASF#AS(N9)
  2 A-SF-D(A) Dynamic
  2 INC@OME(1:6)
    3 C@BONUS(I4)
    3 BONUS(A9/1:4)
END-DEFINE
```

生成された **DTD**

```
<!-- DTD XMLTOOLS BEISP -->
<!ELEMENT NATdollarEMPLOYEE ( MANatWORK , AdollarTESTdollarMAKL ,
    ASslashFAslashSD , hashASDFAS , ASFhashAS , A-SF-D , INCatOME* ) >
<!ATTLIST NATdollarEMPLOYEE PERSONNELslashID CDATA #IMPLIED >
<!ELEMENT MANatWORK ( JOB ) >
<!ELEMENT JOB (#PCDATA) >
<!ELEMENT AdollarTESTdollarMAKL (#PCDATA) >
<!ELEMENT ASslashFAslashSD (#PCDATA) >
<!ELEMENT hashASDFAS (#PCDATA) >
<!ELEMENT ASFhashAS (#PCDATA) >
<!ELEMENT A-SF-D (#PCDATA) >
<!ELEMENT INCatOME ( BONUS* ) >
<!ELEMENT BONUS (#PCDATA) >
```

パーサー **CALLBACK** コピーコード

XML ツールキットを使用して、**Natural Simple XML Parser** (Natural のシンプルな XML パーサー) で使用するコピーコードを生成できます。

コールバックコピーコードでは、次のオペランドを使用します。

オペランド	フォーマット／長さ	説明	PARSER-X から
1	A	要素構造を表す ex-XPATH	オペランド 2
2	A1	XPATH の内容のタイプ： ? 処理命令 D DOCTYPE ! コメント C CDATA セクション T 開始タグ @ 属性 / 終了タグ	オペランド 3
3	A	見つかった要素の内容	オペランド 4
4	L	解析されたデータが空の場合、TRUE	オペランド 5
5	I4	カウンタ変数 1 次元	
6	I4	カウンタ変数 2 次元	
7	I4	カウンタ変数 3 次元	

コピーコード **EMPL-P** の例：

```

* -----
* Parameter Definition
*
* &1& 'XML_PARSER_XPATH'          /* XPATH to represent element...
* &2& 'XML_PARSER_XPATH_TYPE'     /* Type of the XPATH:
*                                  ? Processing instruction
*                                  D DOCTYPE
*                                  ! Comment
*                                  C CDATA section
*                                  T Starting Tag
*                                  @ Attribute
*                                  / Close Tag
*                                  $ Parsed Data
* &3& 'XML_PARSER_CONTENT'        /* Content of found element
* &4& 'XML_PARSER_CONTENT_IS_EMPTY' /* Is TRUE if Content is empty
* &5& '#CX'                        /* Counter Variable 1st Dimension
* &6& '#CY'                        /* Counter Variable 2nd Dimension
* &7& '#CZ'                        /* Counter Variable 3rd Dimension
* -----
*
DECIDE ON FIRST &1&
VALUE 'EMPLOYEE'
RESET EMPLOYEE

```

```
VALUE 'EMPLOYEE/@PERSONNEL-ID'  
  /* #REQUIRED  
  EMPLOYEE.PERSONNEL-ID := &3&  
VALUE 'EMPLOYEE/FULL-NAME'  
  IGNORE  
VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME'  
  IGNORE  
VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME/$'  
  EMPLOYEE.FIRST-NAME := &3&  
VALUE 'EMPLOYEE/FULL-NAME/NAME'  
  IGNORE  
VALUE 'EMPLOYEE/FULL-NAME/NAME/$'  
  EMPLOYEE.NAME := &3&  
VALUE 'EMPLOYEE/FULL-ADDRESS'  
  IGNORE  
VALUE 'EMPLOYEE/FULL-ADDRESS/ADDRESS-LINE'  
  /* OPTIONAL MULTIPLE IST: 18 PARENT: FULL-ADDRESS  
  ADD 1 TO EMPLOYEE.C@ADDRESS-LINE  
VALUE 'EMPLOYEE/FULL-ADDRESS/ADDRESS-LINE/$'  
  &5& := EMPLOYEE.C@ADDRESS-LINE  
  EMPLOYEE.ADDRESS-LINE(&5&) := &3&  
VALUE 'EMPLOYEE/FULL-ADDRESS/CITY'  
  IGNORE  
VALUE 'EMPLOYEE/FULL-ADDRESS/CITY/$'  
  EMPLOYEE.CITY := &3&  
VALUE 'EMPLOYEE/FULL-ADDRESS/ZIP'  
  IGNORE  
VALUE 'EMPLOYEE/FULL-ADDRESS/ZIP/$'  
  EMPLOYEE.ZIP := &3&  
VALUE 'EMPLOYEE/FULL-ADDRESS/COUNTRY'  
  IGNORE  
VALUE 'EMPLOYEE/FULL-ADDRESS/COUNTRY/$'  
  EMPLOYEE.COUNTRY := &3&  
VALUE 'EMPLOYEE/TELEPHONE'  
  IGNORE  
VALUE 'EMPLOYEE/TELEPHONE/PHONE'  
  IGNORE  
VALUE 'EMPLOYEE/TELEPHONE/PHONE/$'  
  EMPLOYEE.PHONE := &3&  
VALUE 'EMPLOYEE/TELEPHONE/AREA-CODE'  
  IGNORE  
VALUE 'EMPLOYEE/TELEPHONE/AREA-CODE/$'  
  EMPLOYEE.AREA-CODE := &3&  
VALUE 'EMPLOYEE/JOB-TITLE'  
  IGNORE  
VALUE 'EMPLOYEE/JOB-TITLE/$'  
  EMPLOYEE.JOB-TITLE := &3&  
VALUE 'EMPLOYEE/INCOME'  
  /* OPTIONAL MULTIPLE IST: 18 PARENT: EMPLOYEE  
  ADD 1 TO EMPLOYEE.C@INCOME  
VALUE 'EMPLOYEE/INCOME/SALARY'  
  IGNORE
```

```

VALUE 'EMPLOYEE/INCOME/SALARY/$'
  &5& := EMPLOYEE.C@INCOME
  EMPLOYEE.SALARY(&5&) := &3&
VALUE 'EMPLOYEE/INCOME/BONUS'
  /* OPTIONAL MULTIPLE IST: 18 PARENT: INCOME
  &5& := EMPLOYEE.C@INCOME
  ADD 1 TO EMPLOYEE.C@BONUS(&5&)
VALUE 'EMPLOYEE/INCOME/BONUS/$'
  &5& := EMPLOYEE.C@INCOME
  &6& := EMPLOYEE.C@BONUS(&5&)
  EMPLOYEE.BONUS(&5&,&6&) := &3&
NONE
IGNORE
END-DECIDE

```

サブプログラムの例：

```

* -----
* CLASS NATURAL XML TOOLKIT - UTILITIES
*
*
* DESCRIPTION
*           Parse a given XML document.
*
*
* AUTHOR          SAG    01.2006
*
* VERSION         6.2.
*
* (c) Copyright Software AG 2006. All rights reserved.
* -----
*
DEFINE DATA PARAMETER
1 XML_PARSER_INPUT          (A) DYNAMIC
PARAMETER USING EMPL
PARAMETER
1 XML_PARSER_ERROR_TEXT    (A253)
1 XML_PARSER_RESPONSE      (I2)
*
LOCAL USING PARSER-X
LOCAL
1 XML_PARSER_XPATH         (A) DYNAMIC
1 XML_PARSER_XPATH_TYPE   (A1)
1 XML_PARSER_CONTENT       (A) DYNAMIC
1 XML_PARSER_CONTENT_IS_EMPTY (L)
*
LOCAL
1 #CX                      (I4)
1 #CY                      (I4)
1 #CZ                      (I4)

```

```

END-DEFINE
*
* ----- INCLUDE THE PARSER
INCLUDE PARSER_X 'XML_PARSER_INPUT' /* XML file to be parsed
'XML_PARSER_XPATH' /* XPATH to represent element...
'XML_PARSER_XPATH_TYPE' /* Type of callback
'XML_PARSER_CONTENT' /* Content of found element
'XML_PARSER_CONTENT_IS_EMPTY' /* Is TRUE if element is empty
'XML_PARSER_ERROR_TEXT' /* error Message
'XML_PARSER_RESPONSE' /* Error NR; 0 = OK
*
* ----- CALLBACK HANDLER
DEFINE SUBROUTINE CALLBACK
*
INCLUDE EMPL-P 'XML_PARSER_XPATH' /* XPATH to represent element...
'XML_PARSER_XPATH_TYPE' /* Type of callback
'XML_PARSER_CONTENT' /* Content of found element
'XML_PARSER_CONTENT_IS_EMPTY' /* Is TRUE if element is empty
'#CX'
'#CY'
'#CZ'
*
END-SUBROUTINE
/*
DEFINE SUBROUTINE PARSER_ERROR
IGNORE
END-SUBROUTINE
END

```

使用した **Natural PDA EMPL** :

```

DEFINE DATA PARAMETER
1 EMPLOYEE
  2 ATTRIBUTES_OF_EMPLOYEE
    3 PERSONNEL-ID(A8)
*
  2 FULL-NAME
    3 FIRST-NAME(A20)
    3 NAME(A20)
*
  2 FULL-ADDRESS
    3 C@ADDRESS-LINE(I4)
    3 ADDRESS-LINE(A20/1:6)
    3 CITY(A20)
    3 ZIP(A20)
    3 COUNTRY(A3)
*
  2 TELEPHONE
    3 AREA-CODE(A6)
    3 PHONE(A15)
*

```

```
2 JOB-TITLE(A25)
*
2 C@INCOME(I4)
2 INCOME(1:6)
  3 SALARY(A9)
  3 C@BONUS(I4)
  3 BONUS(A9/1:4)
END-DEFINE
```


76 解析エラーメッセージ

次のエラーメッセージが、パーサーによって生成されます。

レスポンス	エラーテキスト	例
00	解析はエラーなしで終了しました。	valid/*
-01	文字セットが不適切であるか、またはドキュメントの先頭が '<' ではありません。	not-wf/sa/147.xml
-02	処理命令が閉じられていません。位置：%2%	not-wf/sa/004.xml
-03	CDATA セクションが閉じられていません。位置：%2%	esi/001.xml
-04	!DOCTYPE セクションが閉じられていません。位置：%2%	not-wf/sa/055.xml
-05	コメントで不正な構文が使用されています。位置：%2%	not-wf/sa/006.xml
-06	コメントが閉じられていません。位置：%2%	not-wf/sa/027.xml
-07	CDATA セクションが閉じられていません。位置：%2%	not-wf/sa/017.xml
-08	コメントセクションが閉じられていません。位置：%2%	esi/002.xml
-09	終了タグ名の先頭に不正な文字が指定されています。位置：%2%	not-wf/sa/019.xml
-10	終了タグに開始エレメントがありません。位置：%2%	not-wf/sa/042.xml
-11	終了タグ %3% が開始タグ %1% と一致していません。位置：%2%	not-wf/sa/039.xml
-12	終了タグが閉じられていません。位置：%2%	esi/003.xml
-13	終了タグ %1% が閉じられていません。位置：%2%	例はありません
-14	開始タグ名の先頭に不正な文字が指定されています。位置：%2%	not-wf/sa/035.xml
-15	タグ %1% の属性名が見つかりません。位置：%2%	例はありません
-16	タグ %1% の属性名は不正な文字を含んでいます。位置：%2%	not-wf/sa/001.xml
-17	タグ %1% の属性値の終了引用符が見つかりません。位置：%2%	not-wf/sa/013.xml
-18	タグ %1% の属性値の終了アポストロフィが見つかりません。位置：%2%	esi/005.xml
-19	開始タグセクションが閉じられていません。位置：%2%	esi/006.xml
-20	タグ %1% セクションが閉じられていません。位置：%2%	not-wf/sa/176.xml
-21	セクションが閉じられていません。位置：%2%	not-wf/sa/025.xml

解析エラーメッセージ

レスポンス	エラーテキスト	例
< -8000	ユーザー定義のエラーメッセージです。パーサーは終了します。	例はありません
< -9000	ユーザー定義のエラーメッセージです。PARSER_ERRORが呼び出され、パーサーは終了します。	例はありません

索引
