

Natural for Windows

Web テクノロジ

バージョン 6.3.3

October 2008

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

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

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

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

目次

1 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 インターフェイスについて	6
アーキテクチャ	6
Natural Web インターフェイスモジュール	9
機能	10
機能	11
セキュリティ	12
4 Natural Web インターフェイスのインストールと構成	15
5 Natural Web インターフェイスの構成	17
サポートされる HTTP サーバー	18
RPC および RPC サーバーの構成	18
DCOM サーバーの構成	20
Web インターフェイスの構成	20
HTTP サーバーの構成	24
Natural Security との通信	25
6 Web インターフェイスのトラブルシューティング	27
7 Natural Web インターフェイスの要点	31
8 Natural Web インターフェイスでの作業	33
環境の設定	34
Natural でのサブプログラムの構築	35
9 Natural Web サーバーエクステンション	51
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	53
一般的な情報	54
インストール - RPC/DCOM	54
変換	55
変数	55
エラーログとメッセージ	55
プログラムの呼び出し	56
11 Natural Web サーバーエクステンションの概要 - SYSWEB3 を使用する場合	57
一般的な情報	58
インストール - RPC/DCOM	58
変換	59
変数	59
エラーログとメッセージ	59
プログラムの呼び出し	60
12 Natural Web サーバーエクステンション - 初期化ファイル	61
一般的な情報	62
RPC パラメータ	62
PAL パラメータ	63
DCOM パラメータ	63
Natural Web サーバーエクステンション設定	63

データ転送設定	66
HTTP サーバー変数	67
追加の変数	67
エラーテンプレート	67
13 Natural Web サーバーエクステンション - エラーメッセージ	71
14 テストユーティリティ WEB-ONL (SYSWEB)	77
前提条件	78
アプリケーションの実行	78
サポートされているコンテンツタイプ	79
入力/出力フィールド	80
15 テストユーティリティ WEB-ONL3 (SYSWEB3)	83
前提条件	84
アプリケーションの実行	84
サポートされているコンテンツタイプ	85
入力/出力フィールド	86
16 プログラミングのヒント	89
小文字での編集	90
引用符とアポストロフィ	90
値で定義された変数	91
リソースへのアクセス	91
定数値	91
新しいページの作成	92
DCOM/RPC	92
17 Web インターフェイスの管理	93
戻りページトランスポートバッファサイズの設定	94
ユーザー定義エラーページの作成	95
ユーザー定義エラーページ XML スタイルの作成	95
英数字から HTML への変換	96
英数字から URL への変換	96
18 デモアプリケーション - JavaScript なし	97
ビジネス要件	98
設計に関する決定	99
ライブラリ、モジュール、および命名規則	99
デモアプリケーションの開始	100
Natural Web インターフェイスオンラインマニュアルの開始	100
要件	100
19 デモアプリケーション - JavaScript 使用	101
ビジネス要件	102
設計に関する決定	103
デモアプリケーションの開始	103
要件	103
20 Natural Web インターフェイスのエラーメッセージ	105
エラーメッセージ	106
21 Natural Web インターフェイス SYSWEB から SYSWEB3 への移行	107
22 Web インターフェイスプラグイン	109

開始する前に	110
Web インターフェイスプラグインコマンドの呼び出し	110
Web インターフェイスプラグイン機能	112
23 Natural Web オンラインドキュメント SYSWEB	135
一般的な情報	136
基本モジュール	136
テンプレート/XSLT 処理	138
HTML エクステンション	138
ユーティリティ	140
デモアプリケーション	141
24 出力エリアのクリア	143
25 ドキュメントのコンテンツタイプの設定	145
26 出力エリアのサイズのカウンタ	147
27 エラーページの生成	149
28 ドキュメントへの書き込みおよび有効な HTML への変換	151
29 ドキュメントへの HTTP 設定の書き込み	153
30 内部値に関する情報	155
31 ドキュメントの終了および初期化	157
32 すべての環境変数のリスト	159
33 ドキュメントの場所の設定	161
34 環境変数の読み取り	163
35 環境変数グループの読み取り	165
36 環境テキストエリア変数の読み取り	167
37 ドキュメントへのテキストの書き込み	169
38 出力エリアへの改行の書き込み	171
39 テキストから HTML	173
40 テキストから XML	175
41 テキストから URL	177
42 リターンドキュメント内での置換	179
43 リソースのロード	181
44 リソースの読み取り	183
45 リソースの書き込み	185
46 リソースの確認	187
47 リソースの削除	189
48 リターンドキュメントへの XSLT 処理の適用	191
49 リソースからリターンドキュメントへの XSLT 処理の適用	193
50 リソースディレクトリからのスタイルシートのロードおよびリターンドキュメントへの XSLT 処理の適用	195
51 出力ページの読み取り	197
52 アンカー	199
53 ボタン	201
54 チェックボックスグループ	203
55 コメント行	205
56 レベル n のヘッダー	207
57 画像	209

58 入力	211
59 改行	213
60 フォーム	215
61 HTML ドキュメント	217
62 箇条書き	219
63 段落	223
64 ラジオボタングループ	225
65 横罫線	227
66 スクロールリスト	229
67 表	231
68 汎用タグ	235
69 テキストエリア	237
70 テキストから URL - デコード形式	239
71 時刻/日付文字列	241
72 すべての Natural ライブラリのリスト	243
73 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	245
74 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	247
75 Natural ライブラリのディレクトリのリスト	249
76 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	251
77 Natural ソースオブジェクトとして保存される HTML ページを返す	253
78 Natural Web インターフェイスの現在の設定のリスト	255
79 Natural オブジェクトのソースのリスト	257
80 オンラインドキュメント	259
81 Natural 以外のファイル/リソースのリスト	261
82 Natural Web オンラインドキュメント SYSWEB3	263
全般的な情報	264
基本モジュール	264
テンプレート/XSLT 処理	266
HTML エクステンション	267
ユーティリティ	268
デモアプリケーション	269
83 ドキュメントへのバイナリの書き込み	271
84 出力エリアのクリア	273
85 ドキュメントのコンテンツタイプの設定	275
86 出力エリアのサイズのカウンタ	277
87 エラーページの生成	279
88 ドキュメントへの書き込みおよび有効な HTML への変換	281
89 ドキュメントへの HTTP 設定の書き込み	283
90 内部値に関する情報	285
91 ドキュメントの終了および初期化	287
92 すべての環境変数のリスト	289
93 MIME タイプと転送/データタイプの評価	291
94 MIME タイプとファイル拡張子の評価	293
95 ドキュメントの場所の設定	295

96 環境変数の読み取り	297
97 環境変数グループの読み取り	299
98 環境テキストエリア変数の読み取り	301
99 ドキュメントへのテキストの書き込み	303
100 出力エリアへの改行の書き込み	305
101 ダイナミック X-array への Natural ソースの読み込み	307
102 テキストから HTML	309
103 テキストから XML	311
104 テキストから URL	313
105 リターンドキュメント内での置換	315
106 リソースのロード	317
107 リソースの読み取り	319
108 リソースの書き込み	321
109 リソースの確認	323
110 リソースの削除	325
111 リターンドキュメントへの XSLT 処理の適用	327
112 リソースからリターンドキュメントへの XSLT 処理の適用	329
113 リソースディレクトリからのスタイルシートのロードおよびリターンドキュメントへの XSLT 処理の適用	331
114 リソースファイルのリスト	333
115 入力ページの読み込み	335
116 出力ページの読み取り	337
117 アンカー	339
118 ボタン	341
119 チェックボックスグループ	343
120 コメント行	345
121 レベル n のヘッダー	347
122 画像	349
123 入力	351
124 改行	353
125 フォーム	355
126 HTML ドキュメント	357
127 箇条書き	359
128 段落	363
129 ラジオボタングループ	365
130 横罫線	367
131 スクロールリスト	369
132 表	371
133 汎用タグ	375
134 テキストエリア	377
135 テキストから URL - デコード形式	379
136 時刻/日付文字列	381
137 すべての Natural ライブラリのリスト	383
138 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	385
139 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	387


140 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データのリスト	389
141 Natural ライブラリのディレクトリのリスト	391
142 Natural ライブラリのリソースのリスト	393
143 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	395
144 Natural ソースオブジェクトとして保存される HTML ページを返す	397
145 Natural Web インターフェイスの現在の設定のリスト	399
146 Natural オブジェクトのソースのリスト	401
147 オンラインドキュメント	403
148 Natural 以外のファイル/リソースのリスト	405
149 XML ツールキットプラグイン	407
150 はじめに	409
XML ツールキットの特徴	410
XML ツールキットの説明	410
考慮事項と制限事項	414
151 XML ツールキットの使用	419
必要条件	420
アプリケーションの起動	420
ヘルプ	421
152 特定の生成オプションの設定	423
Natural XML オプションメニューの起動	424
Generation	424
Path	427
オプションの永続的な保存	429
153 Natural データソースの使用	431
Natural データエリアの選択	432
データタイプの選択	434
DTD 定義または XML スキーマでのファイルの生成	435
XML ドキュメントのシリアライズの生成	437
XML ドキュメントのパースの生成	438
パラメータの設定	439
ルートグループの選択	440
生成レポートの表示	440
154 外部データソースの使用	443
文書型定義または XML スキーマからの生成	444
ルートエレメントまたは文書型の選択	446
再帰レベルの選択	446
Natural データエリアの生成	448
シリアライズのためのコピーコードの生成	449
シリアライズのためのサブプログラムの生成	451
XML パーサーコールバックのためのコピーコードの生成	451
XML パーサーコールバックのためのサブプログラムの生成	453
生成結果の表示	453
155 Natural のシンプルな XML パーサー	455

パーサーの説明と例	456
パーサーの制限事項	463
156 例	465
コピーコードのシリアライズ	466
サブルーチンのシリアライズ	468
生成された Natural データエリア	476
Natural DTD パーサー	478
生成されたタイプ定義	479
パーサー CALLBACK コピーコード	480
157 解析エラーメッセージ	485
索引	487

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を説明します。

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 Natural 環境をリンクします。
 Web インターフェイスプラグイン	Web インターフェイスプラグインは、Natural スタジオのオプションのプラグインユニットです。
 XML ツールキットプラグイン	XML ツールキットプラグインにより、開発者は Natural 内で XML ドキュメントを処理できます。

Natural の Web テクノロジと併用できる Natural ステートメントの詳細については、『ステートメント』ドキュメントの「機能別ステートメント」の「インターネットと XML」を参照してください。

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 Natural 環境をリンクします。

Natural Web インターフェイスのドキュメントでは、次のトピックについて説明します。

- **Natural Web** インターフェイスについて
- **Natural Web** インターフェイスのインストールと構成
- **Natural Web** インターフェイスの要点
- **Natural Web** インターフェイス **SYSWEB** から **SYSWEB3** への移行
- **Natural Web** オンラインドキュメント **SYSWEB**
- **Natural Web** オンラインドキュメント **SYSWEB3**



Note: 互換性を確保するため、新しい Natural Web インターフェイスの SYSWEB3 だけでなく、以前のバージョンの SYSWEB も Natural に付属しています。Natural Web インターフェイスのドキュメントの内容は、特記されていない限り、両方のバージョンの Natural Web インターフェイスに適用されます。

3 Natural Web インターフェイスについて

- Natural Web インターフェイスについて 6
- アーキテクチャ 6
- Natural Web インターフェイスモジュール 9
 - 機能 10
 - 機能 11
 - セキュリティ 12

多くの組織では、情報やサービスをインターネット経由で提供する必要があります。以前は、Web ページに日々訪れる訪問者にはスタティックな HTML で十分でした。今日、Web では洗練された HTML ページの競争が激しくなっており、インターネット経由でのビジネスロジックへの完全なアクセスに対する需要は激増しています。ビジネスに不可欠な情報を含んでいるデータベース管理システムの大部分は、メインフレームのような高性能サーバーを基準にしています。

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 インターフェイスについて

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 Natural 環境をリンクします。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 に実装するための基準です。

対話式アプリケーションでは、ユーザーは情報を入力し、その入力に応じた出力を発行して反応を示すことができます。Web ベースのアプリケーションの例には、注文入力システム、旅行予約サービス、集配システムなどがあります。これは、Natural アプリケーションの範囲を大きく拡張します。企業内ユーザーではない、世界中の潜在的な顧客が、現在、同じアプリケーションを使用できます。

また、何よりも、そのようなアプリケーションを実装するために、Natural ユーザーは新しいプログラミング言語を学ぶ必要がないのです。ナビゲーションとユーザー入出力は、Natural で完全に実装されます（追加の埋め込み HTML ステートメントをいくつか使用します）。

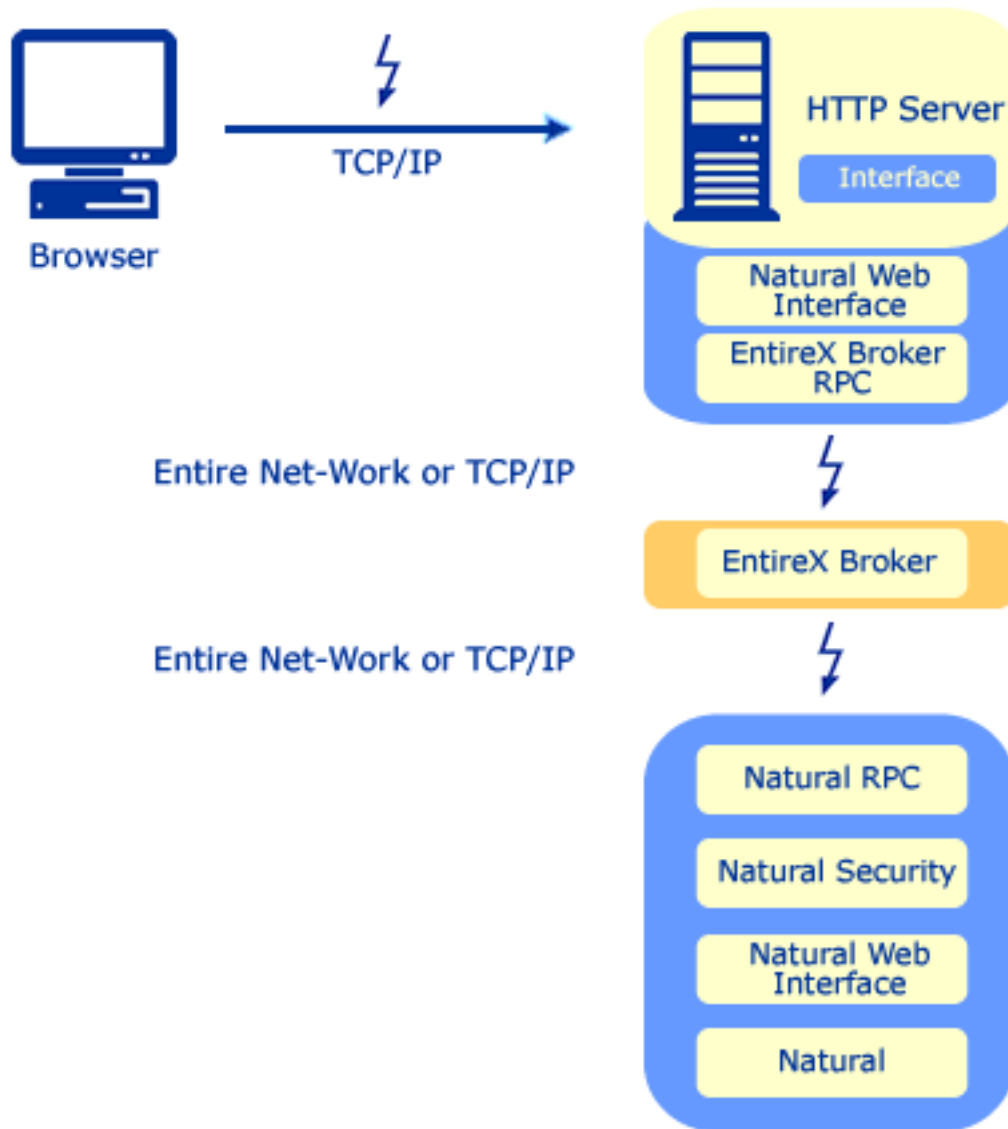
アーキテクチャ

次のトピックについて説明します。

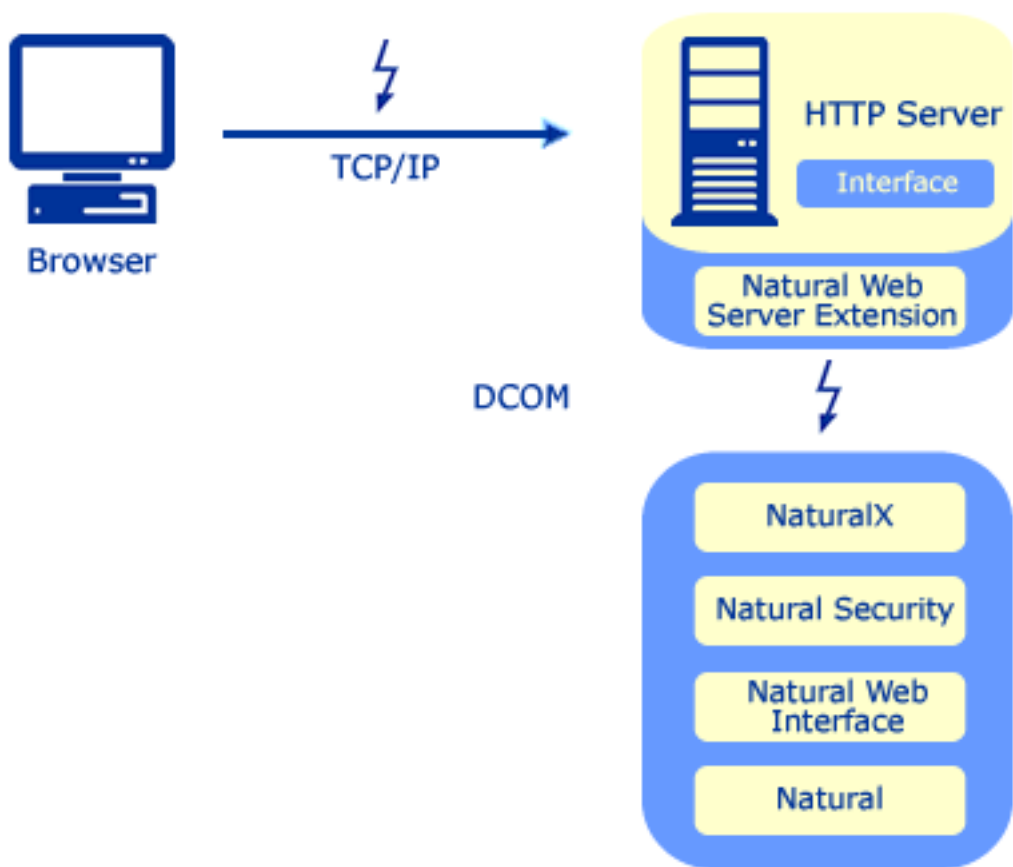
- [Natural RPC 技法を用いた通信](#)
- [DCOM 技法を用いた通信](#)

- PAL 技法を用いた通信

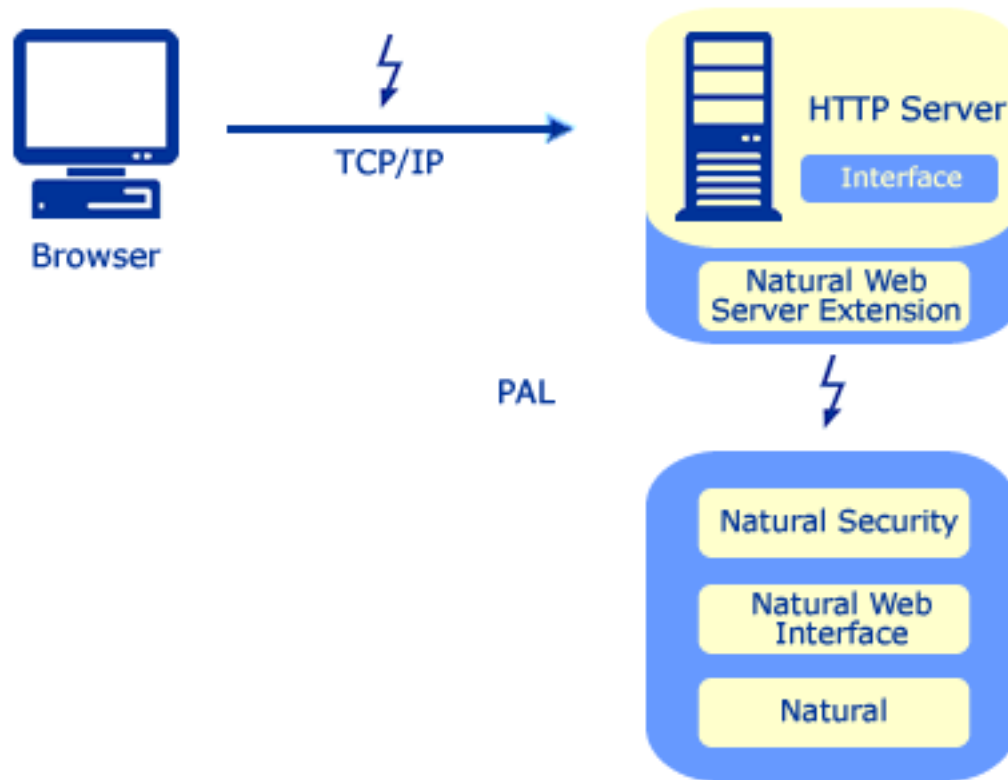
Natural RPC 技法を用いた通信



DCOM 技法を用いた通信



PAL 技法を用いた通信



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、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、PAL、または DCOM テクノロジーを使用する 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な機能の1つは、Natural サブプログラムを Web ページから呼び出せることです。これは、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サブミットすることができます。これは、ユーザーデータをパラメータとして渡す Natural サブプログラムを実行します。

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 サーバーによって提供される SQL（および ODBC ドライバ）を使用してデータを検索するための簡単なデータベースアクセスは、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。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です。

それらは操作可能なアプリケーションのために過去に実装されたので、整合性チェックなどのビジネスロジックは通常はすでに存在しています。それらが別の 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）として実装されたならば、それらは容易に再利用できるので、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はないので、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の2番目に重要な部分は、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して、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。

利点は、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ことです。

さらに、Natural は幅広いソース（Adabas、RDBMS、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ので、アプリケーションタイプは事実上無限です（Natural で構築できるあらゆるアプリケーションは 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を基準にしています。

これは既存のクライアント／サーバー環境のシームレスな統合を可能にします。Web 接続は別のクライアントであり、既存の Natural サーバーに接続できます。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、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。

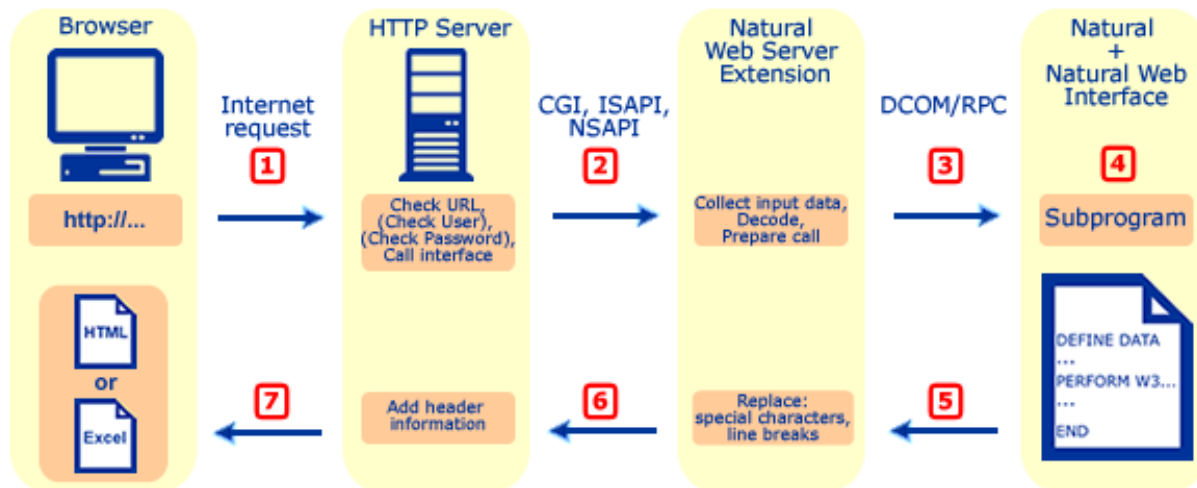
Web ページの作成

Web ページは、標準ツール (Macromedia Dreamweaver や Microsoft Frontpage など) で、または Natural 生成機能を使用した Web ページ作成ツールで作成できます。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。これが Natural サブプログラムを実行する要求形式であった場合は、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プログラム集合を発行します。

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は、URL を、元は URL で識別された Natural サーバープログラムを呼び出す Natural RPC に変換します。
4. Natural サブプログラムが実行されて、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て、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。

 **Note:** 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 とともに動作します。これは、以下のようにして行われます。

- 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したときにユーザー ID とパスワードの入力を求められます。
- HTTP サーバーは、データベースで与えられたデータを検証します。
- ユーザーが認証されると、Natural Web サーバーエクステンションがリモートユーザー名で呼び出されます。
- Natural RPC サーバーが Natural Security で開始されると、与えられた名前は *USER として設定されます。
- 承認はすでに HTTP サーバーによって行われているので、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。

2 番目のシナリオは、初期化ファイルが開始されるときに、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ことです。「[Natural Security との通信](#)」も参照してください。

4 Natural Web インターフェイスのインストールと構成

Natural Web インターフェイスは、Natural for Windows のインストール時にインストールされます。



Note: カスタムインストールを選択した場合のみ、Natural Web インターフェイスをインストールできます。このとき、インストールする Natural Web サーバーエクステンションを指定できます。

詳細については、Natural for Windows の『インストール』ドキュメントの「Setup Type」を参照してください。

このドキュメントの内容は以下のとおりです。

- **Natural Web インターフェイスの構成** Natural Web インターフェイスの構成方法について説明します。わからない製品がある場合は、詳細について対応するインストール手順を参照してください。
- **Web インターフェイスの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	18
■ RPC および RPC サーバーの構成	18
■ DCOM サーバーの構成	20
■ Web インターフェイスの構成	20
■ HTTP サーバーの構成	24
■ Natural Security との通信	25

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。わからない製品がある場合は、詳細について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。

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 Version 2.0.x■ Apache Version 2.2.x
(*)	<ul style="list-style-type: none">■ Apache Version 2.0.x■ Apache Version 2.2.x

RPC および RPC サーバーの構成

以下の構成の説明で、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構成について、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。

- [SYSWEB3 を使用する RPC の全般的な設定 \(すべてのプラットフォーム\)](#)
- [現在のバージョンの Natural for Mainframes/UNIX/Windows](#)

- EntireX Communicator/EntireX 開発者キット

SYSWEB3 を使用する RPC の全般的な設定（すべてのプラットフォーム）

ACIVERS の設定

プロファイルパラメータ ACIVERS の値は、6 以上に設定することをお勧めします。詳細については、『Natural リモートプロシージャコール (RPC)』ドキュメントの「ACI バージョンの設定」を参照してください。

MAXBUFF と MAX-MESSAGE-LENGTH の設定

Natural RPC サーバーのプロファイルパラメータ MAXBUFF の値は、1024 以上に設定することをお勧めします。このように設定した場合は、EntireX Communicator に含まれる EntireX Broker の MAX-MESSAGE-LENGTH の値を 1048576 以上に設定する必要があります。

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：

- ユーザーライブラリに 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

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)

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。

DCOM サーバーの構成

DCOM サーバーをインストールおよび構成するには、「NaturalX」の手順に従います。

以下の構成の説明で、NATWEBEXT は外部 DCOM サーバーの名前、NATWEB はローカル DCOM サーバーの名前です。

このセクションでは、次のトピックについて説明します。

- [NaturalX サーバー](#)

NaturalX サーバー

Natural Web インターフェイスをサポートするすべてのサーバーで、ライブラリ SYSWEB3 (または SYSWEB) と SYSEXT を steplib として追加する必要があります。「[現在のバージョンの Natural for Mainframes/UNIX/Windows](#)」を参照してください。

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](#)

■ MOD 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、構成は不要です。

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」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)

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。この場合は、NATWEB がグループ USER に属しているか、GUEST アカウントが有効になっている必要があります。

Windows (Apache)

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

MOD 用の Natural Web サーバーエクステンション



Note: このセクションの内容は、SYSWEB3 にのみ適用されます。

- [RPC サーバーの使用](#)
- [DCOM サーバーの使用](#)
- [SPoD サーバーの使用](#)

RPC サーバーの使用

1. Natural Web サーバーエクステンションをインストールします。
2. HTTP サーバーの ... \conf\httpd.conf ファイルを開き、RPC インターフェイス用に以下の行を追加します。

Apache 2.0.x の場合

```
...
LoadModule nww3_module modules/nww3mod2.dll
<Location /nww3/mod>
  AllowOverride None
  NWWiniFile "<yourRoot>/nww3/nww3mod.ini"
  SetHandler nww3
</Location>
...
```

Apache 2.2.x の場合

```
...
LoadModule nww3_module modules/nww3mod22.dll
<Location /nww3/mod>
  AllowOverride None
  NWWiniFile "<yourRoot>/nww3/nww3mod.ini"
  SetHandler nww3
</Location>
...
```

3. 複数のサーバーまたはブローカーを使用する場合は、/nww3 ディレクトリに追加のファイルを指定します。

4. `.ini` ファイルのスタティックな読み込みが必要な場合（パフォーマンスに影響します）、イタリック体で示している行を `obj.conf` に追加します。

DCOM サーバーの使用

1. Natural Web サーバーエクステンションをインストールします。
2. HTTP サーバーの `... \conf \httpd.conf` ファイルを開き、DCOM インターフェイス用に以下の行を追加します。

Apache 2.0.x の場合

```
...
LoadModule nww3d_module modules/nww3dmod2.d11
<Location /nww3d/mod>
    AllowOverride None
    NWWiniFile "<yourRoot>/nww3d/nww3dmod.ini"
    SetHandler nww3d
</Location>
...
```

Apache 2.2.x の場合

```
...
LoadModule nww3d_module modules/nww3dmod22.d11
<Location /nww3d/mod>
    AllowOverride None
    NWWiniFile "<yourRoot>/nww3d/nww3dmod.ini"
    SetHandler nww3d
</Location>
...
```

3. 複数のサーバーまたはブローカーを使用する場合は、`/nww3d` ディレクトリに追加のファイルを指定します。
4. `.ini` ファイルのスタティックな読み込みが必要な場合（パフォーマンスに影響します）、イタリック体で示している行を `obj.conf` に追加します。

SPoD サーバーの使用

1. Natural Web サーバーエクステンションをインストールします。
2. HTTP サーバーの ... \conf\httpd.conf ファイルを開き、SPoD (PAL) インターフェイス用に以下の行を追加します。

Apache 2.0.x の場合

```
...
LoadModule nww3p_module modules/nww3pmod2.d11
<Location /nww3p/mod>
  AllowOverride None
  NWWiniFile "<yourRoot>/nww3p/nww3pmod.ini"
  SetHandler nww3p
</Location>
...
```

Apache 2.2.x の場合

```
...
LoadModule nww3p_module modules/nww3pmod22.d11
<Location /nww3p/mod>
  AllowOverride None
  NWWiniFile "<yourRoot>/nww3p/nww3pmod.ini"
  SetHandler nww3p
</Location>
...
```

3. 複数のサーバーまたはブローカーを使用する場合は、/nww3p ディレクトリに追加のファイルを指定します。
4. .ini ファイルのスタティックな読み込みが必要な場合 (パフォーマンスに影響します)、イタリック体で示している行を obj.conf に追加します。

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。この場合は、IUSR_NATWEB がグループ USER に属しているか、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`) を設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通過するには、**Natural Security** を "AUTO=ON" で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

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 File not found.	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 error 00000000 occurred. Severity = Success</p> <p>Message:... 9999 NAT0935 Conflicting number of parameters (Subprogram...). Lib=... Pgm=D3MENU.</p>	間違ったサブプログラムが呼び出されたか、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 (ドキュメント参照) を呼び出し、インターフェイスがコンパイル時に使用されていることを確認します。
<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>Message: ERX_E_SERVICE_NOT_AVAILABLE - ETB error code 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>Message: ERX_E_SERVICE_NOT_AVAILABLE - ETB error code 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>Processing of subprogram TEST in library W3RPCDMO failed.</p> <p>Message: Status = O, Library = W3RPCDMO, Program = NATSRVD, Level = 01, Error = 00082, Line = 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、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、所定のライブラリが含まれていれば steplib または NATPARM を確認します。
<p>Natural RPC Server crash.</p> <p>Test with WEB-ONL on the same subprogram gets: WEB-ONL 1420</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
NAT0937 Conflicting array def.in parm.3 (Subprogram ..).		
Demonstration application does not work.	異なるファイル番号を使用しています。	ライブラリ SYSWEB3 (または SYSWEB) を再カタログしてください。
NAT3048 File/USERID not available at open time.	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の作業** 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。
- **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。
- **テストユーティリティ WEB-ONL (SYSWEB)** SYSWEB を使用する場合のテストユーティリティの使用方法について説明します。
- **テストユーティリティ WEB-ONL3 (SYSWEB3)** SYSWEB3 を使用する場合のテストユーティリティの使用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。
- **Web インターフェイスの管理** 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は、Java をサポートするブラウザを必要とします。
- **Natural Web インターフェイスのエラーメッセージ** 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のリストです。

Natural ライブラリ SYSWEB3（または 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 34
- Natural でのサブプログラムの構築 35

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 Web 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してください)。

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 Web インターフェイスの **SYSWEB3** 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 SYSWEB3。

Natural の `steplib` が使用可能になっているか、ライブラリ `SYSWEB3` 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する必要があります。

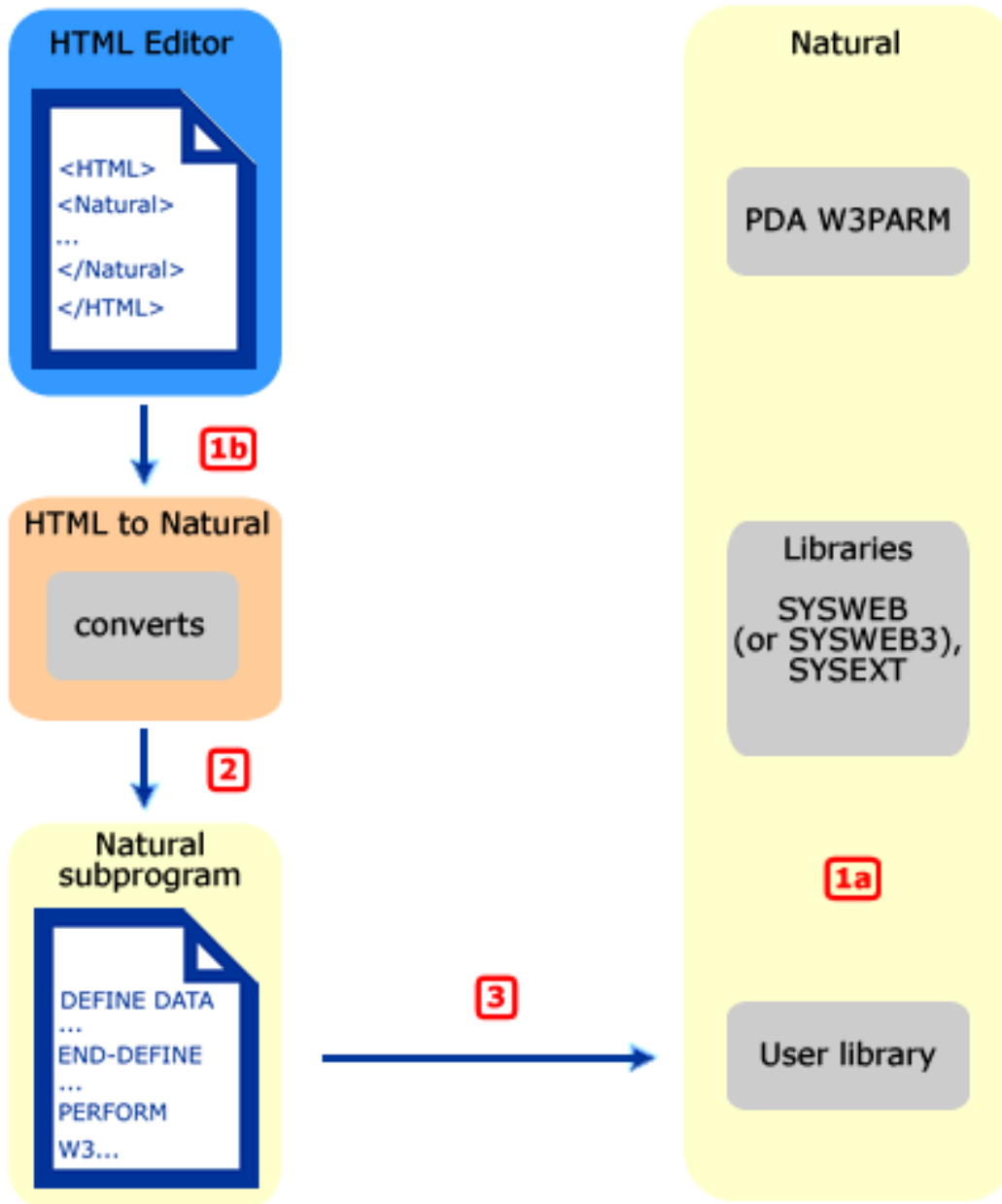
- パラメータデータエリア `W3PARM`。
- Natural `RPC` スタブまたは `NaturalX`。

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、生成されたプログラムを 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 SYSWEB3 (または SYSWEB) の Web インターフェイスプラグインを開始し、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

以下の点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 SYSWEB3 (または 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 SYSWEB3 (または 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直接入力します。ライブラリ SYSWEB3 (または 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 SYSWEB3 (または 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 SYSWEB3 (または 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、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 **SYSWEB3** (または **SYSWEB**) サブルーチンへの呼び出しの入力

```
*
* Example E3END
*
DEFINE DATA
PARAMETER USING W3PARM
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 W3PARM
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）。

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も作成できます。

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 'we'll 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り、実際の新しい行の作成にライブラリ SYSWEB3（または SYSWEB）の W3TEXT で始ま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 SYSWEB3（または 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したくない場合は、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。

SYSWEB3 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 SYSWEB3 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://.../sysweb3/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したくない場合は、Natural プログラム WEB-ONL3 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、ASCII と EBCDIC の間で必要に応じて変換されます。

初期化ファイル

HTML ページに指定された変数だけが、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。環境変数は、接頭辞 "\$" を使用して HTML エラーページに指定できます。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 および SYSWEB3 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せ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 SYSWEB3（または 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

このドキュメントの構成は以下のとおりです。

- **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。
- **SYSWEB3** を使用する場合の概要 SYSWEB3 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

▪ 全般的な情報	54
▪ インストール - RPC/DCOM	54
▪ 変換	55
▪ 変数	55
▪ エラーログとメッセージ	55
▪ プログラムの呼び出し	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パーツは、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 / DCOM

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	DCOM
CGI	nwwcgi.exe nwwcgi.ini	nwwdcgi.exe nwwdcgi.ini
ISAPI	nwwisapi.dll nwwisapi.ini	nwwdisapi.dll nwwdisapi.ini
NSAPI	nwwnsapi.dll nww/nsapi	nwwdnsapi.dll nwwd/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	NWW_INOUT_LENGTH = データ転送量



Note: 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もない実行可能ファイルも使用できます。

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 ページを使っ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 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.exe/your-library/your-program</code>	<code>http://server-name/cgi-</code>
ISAPI	<code>http://server-name/cgi-library/nwwisapi.dll/your-library/your-program</code>	<code>http://server-name/cgi-</code>
NSAPI	<code>http://server-name/nww/nsapi/your-library/your-program</code>	<code>http://server-name/nww-</code>

11 Natural Web サーバーエクステンションの概要 - SYSWEB3 を使用する場合

▪ 全般的な情報	58
▪ インストール - RPC/DCOM	58
▪ 変換	59
▪ 変数	59
▪ エラーログとメッセージ	59
▪ プログラムの呼び出し	60

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パーツは、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プログラムは、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 6.2 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ています。

- 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)。
- Apache Server 専用 Apache Module (mod)。

インストール - RPC / DCOM

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。

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	DCOM
CGI	nww3cgi.exe nww3cgi.ini	nww3dcgi.exe nww3dcgi.ini
ISAPI	nww3isapi.dll nww3isapi.ini	nww3disapi.dll nww3disapi.ini
Apache 2.0.x (UNIX)	nww3mod2.so nww3/mod	nww3dmod2.so nww3d/mod
Apache 2.0.x (PC)	nww3mod2.dll nww3/mod	nww3dmod2.dll nww3d/mod

	RPC	DCOM
パラメータ	RPC_ETB_ID_NAME = ブローカ名 RPC_SERVER_NAME = サービス名 NWW_INOUT_LENGTH = データ転送量	NWW_INOUT_LENGTH = データ転送量



Note: 一部の HTTP サーバー（Windows のみ）では、拡張子 .exe のない実行可能ファイルを使用できます。

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 ページを使っ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がある場所）の名前を含む 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 (PC)	<code>http://server-name/cgi-library/nww3cgi.exe/your-library/your-program</code>	<code>http://server-name</code>
ISAPI	<code>http://server-name/cgi-library/nww3isapi.dll/your-library/your-program</code>	<code>http://server-name</code>
mod	<code>http://server-name/nww3/mod/your-library/your-program</code>	<code>http://server-name</code>

12 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	62
▪ RPC パラメータ	62
▪ PAL パラメータ	63
▪ DCOM パラメータ	63
▪ Natural Web サーバーエクステンション設定	63
▪ データ転送設定	66
▪ HTTP サーバー変数	67
▪ 追加の変数	67
▪ エラーテンプレート	67

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 の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。 <ul style="list-style-type: none"> ■ 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。

PAL パラメータ

これらのパラメータは、SPoD (PAL) インターフェイスとの通信に必要です。

パラメータ	説明
PAL_SERVER_NODE	呼び出される PAL サーバーノードの名前です。
PAL_SERVER_PORT	呼び出される PAL サーバーポートの番号です。
PAL_SESSION_PARAMETER	サーバーに動的パラメータが必要な場合は、一重引用符 (') を使用してセッションパラメータを指定します。

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.

パラメータ	説明
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。 使用される IDL ファイル： <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> SYSWEB3 インターフェイスのデフォルト値は以下のとおりです。 <ul style="list-style-type: none"> ■ NWW_INOUT_FORMAT=0 ■ NWW_INOUT_LENGTH=0
NWW_PASSWORD	ユーザー ID のパスワードを定義します。
NWW_PATH_INFO	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。 例： NWW_PATH_INFO=/syshtml/nat-env
NWW_PATHINFO_PREFIX	このパラメータは、ISAPI インターフェイスを使用する場合のみ使用できます。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は、PATH_INFO 変数の値はディレクトリとファイル名を前置した URL になります（例：/nww/my.nww/sysweb/nat-env）。この接頭辞（斜体）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。 例： NWW_PATHINFO_PREFIX=/nww/my.nww
NWW_OUT_CSS	文字列を特定の文字に置換します。

パラメータ	説明																		
	<p>文字列 文字</p> <p>&#09; --> (タブ)</p> <p>&#64; @</p> <p>&#91; [</p> <p>&#92; \</p> <p>&#93;]</p> <p>&#123; {</p> <p>&#124; /</p> <p>&#125; }</p> <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>																		
NWW_OUT_CSS_TRANSLATE	<p>指定された文字を対応する 16 進値に置換します。 (ASCII のデフォルト値)</p> <table border="1" data-bbox="570 926 878 1325"> <thead> <tr> <th>文字</th> <th>16 進値</th> </tr> </thead> <tbody> <tr> <td>--> (タブ)</td> <td>09</td> </tr> <tr> <td>@</td> <td>40</td> </tr> <tr> <td>[</td> <td>5B</td> </tr> <tr> <td>\</td> <td>5C</td> </tr> <tr> <td>]</td> <td>5D</td> </tr> <tr> <td>{</td> <td>7B</td> </tr> <tr> <td> </td> <td>7C</td> </tr> <tr> <td>}</td> <td>7D</td> </tr> </tbody> </table> <p>英語 EBCDIC (コードページ 37) の例： #(tab), @, [, \,], {, , } NWW_OUT_CSS_TRANSLATE=05,7C,AD,61,BD,C0,4F, D0</p>	文字	16 進値	--> (タブ)	09	@	40	[5B	\	5C]	5D	{	7B		7C	}	7D
文字	16 進値																		
--> (タブ)	09																		
@	40																		
[5B																		
\	5C																		
]	5D																		
{	7B																		
	7C																		
}	7D																		
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バックアウト) 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を前置し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																		

パラメータ	説明
	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。

データ転送設定



Note: このグループのパラメータは、SYSWEB3 でのみ使用できます。

このグループのパラメータは、Natural サーバーへのデータ転送を定義します。

パラメータ	説明
SETMIMETYPE	<p>このパラメータは、受信 MIME タイプの転送タイプを定義します。定義には、MIME タイプの名前と適切な転送タイプが含まれます。MIME タイプには、1文字以上の任意の文字を表すワイルドカードとしてアスタリスクを含めることができます。アプリケーションは常に、最初に一致した MIME タイプを使用します。</p> <p>以下のいずれかの転送タイプを指定できます。</p> <p>binary alpha</p> <p>例</p> <pre>SETMIMETYPE=text/*;*charset*=*utf-16 binary SETMIMETYPE=text/* alpha SETMIMETYPE=image/svg alpha SETMIMETYPE=image/* binary SETMIMETYPE=multipart/form-data* binary SETMIMETYPE=* alpha</pre> <p>注意: 行 SETMIMETYPE=* alpha は、すべての不明な MIME タイプ用のデフォルト設定を定義します。設定しなかった場合は、alpha が使用されます。</p>

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。

よく使用される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。エラーテンプレートは、基本的に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。これを行うには、\$記号で始ま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。

次の"環境変数" はエラーテンプレートに追加できる変数です。

環境変数	説明
NWW_LOGTIME	エラーが記録される日時 (ERROR_LOG_FILE 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: $NWW_VERSION</SMALL></I></TD>
    </TR>
    <TR>
      <TD align="center">
        <B>The following error has been logged in the error log file:</B></TD>
      </TR>
    </TR>
  </TABLE>
  <TABLE border="0" width="100%" cellspacing="15" cellpadding="0">
    <TR valign="top">
      <TD align="right"><B>$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>
```


13 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	No library and subprogram specified.	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。	適切な URL を指定してください。	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。	適切な URL を指定してください。	なし	なし
NWW0003	File ... not found.	アダプタの初期化ファイルが見つかりません。	なし	なし	インストールを再実行します。
NWW0004	No subprogram specified.	指定した URL が正しくありません。	適切な URL を指定してください。	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。		
NWW0010	RPC call failed.	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。
NWW0011	ERX error ... occurred ...	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： <i>Message:</i> ... <i>Program = NATSRVD</i> ... <i>Error = 00082</i> ... 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 error register.	EntireX RPC サービスを初期化できません。	なし	なし	構成を確認してください。
NWW0013	erx.dll cannot be loaded. Subcode:	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 logon failed.	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 logoff failed.	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	File ... not found (Error: ...).	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 too long.	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。

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
NWW0035	NWW_PASSWORD too long.	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文字以下)にしてください。
NWW0036	Natural Library Name too long.	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 Subprogram Name too long.	Natural で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: ... is not supported.	CONTENT_TYPE = application/x-www-form-urlencoded のデータのみがサポートされます。	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 is greater than 30000.	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。 注意: このメッセージは SYSWEB3 には適用されません。	なし	なし	構成を変更してください。
NWW0101	Number of parameters is greater than 200.	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。 注意: このメッセージは SYSWEB3 には適用されません。	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	No Header specified.	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	Page contains no Data.	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	Interface A1(1:v) no longer supported.	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

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
					パラメータを削除します。
NWW1001	No class and method specified.	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。	適切な URL を指定してください。	なし	なし
NWW1002	No class specified.	指定した URL が正しくありません。	適切な URL を指定してください。	なし	なし
NWW1004	No method specified.	指定した URL が正しくありません。	適切な URL を指定してください。	なし	なし
NWW1005	ASCII Unicode conversion failed.	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 conversion failed.	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	Method ... not found.	指定したメソッドを呼び出せません。	適切な URL を指定してください。	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	Class ... not found.	指定したクラスを呼び出せません。	適切な URL を指定してください。	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。

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。		
NWW1009	Initialization of Class ... failed.	指定したクラスを呼び出せません。	適切な URL を指定してください。	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 call failed, error	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 error ... occurred ...	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。	適切な URL を指定してください。	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。
NWW1012	DCOM initialization failed.	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してください。
NWW1013	DCOM release failed.	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してください。
NWW1036	DCOM Class Name too long.	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 Method Name too long.	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
NWW2009	Logon to Library ... failed (Error: ...).	指定したライブラリ名が正しくありません。	適切な URL を指定してください。	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。		
NWW2010	PAL call failed (Error: ...).	PAL 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。
NWW2011	PAL error ... occurred.	内部 PAL エラー。詳細については、PAL のドキュメントを参照してください。	適切な URL を指定してください。	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。
NWW2012	PAL initialization failed (Subcode: ...).	PAL 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してください。
NWW2013	PAL Transport not initialized. (Subcode: ...).	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
NWW2014	PAL Transport failed. (Subcode: ...).	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

14 テストユーティリティ WEB-ONL (SYSWEB)

▪ 前提条件	78
▪ アプリケーションの実行	78
▪ サポートされているコンテンツタイプ	79
▪ 入力／出力フィールド	80

テストユーティリティ **Web オンライン**は、Natural Web インターフェイスのコンポーネントです。HTTP サーバーを使用せずに、サブプログラムをローカルでチェックできます。Web ページの転送パラメータはテストユーティリティに転送され、ビジネスロジックに直接ポストされます。通信プラットフォームとしては、実際のリモート通信と同様に、RPC または DCOM を選択できます。処理の結果、目的の Web ページまたはエラーメッセージが表示されます。Web ページは、選択したブラウザまたはビューアーで表示できます。エラーメッセージが表示された場合でも、追加のテストルーチンを作成する必要はなく、ビジネスロジックをローカルで簡単にデバッグできます。リモートデバッグ不要

機能：

- ローカルでのアプリケーションチェック
- リモートデバッグ不要
- 簡単なエラーチェック
- 使い勝手の良いインターフェイスによる快適な操作性
- テストルーチンの追加作成不要

このセクションでは、次のトピックについて説明します。

前提条件

- さまざまなコンテンツタイプをサポートする Web ブラウザ。例えば、Microsoft Internet Explorer バージョン 5.0 以降。
- 任意の使用可能なテキストエディタ。

アプリケーションの実行

▶手順 14.1. パス調整を定義するには

- 1 メインダイアログボックスを開きます。
- 2 [Tools] メニューの [Options...] でブラウザとビューアーを選択します。
- 3 ブラウザ、ビューアー、およびワークファイルのパスを設定します。
- 4 [OK] ボタンをクリックします。

▶手順 14.2. アプリケーションを開始するには

- 1 ダイアログ **WEB-ONL** を開始します。
- 2 ライブラリ名とサブプログラム名を選択します。

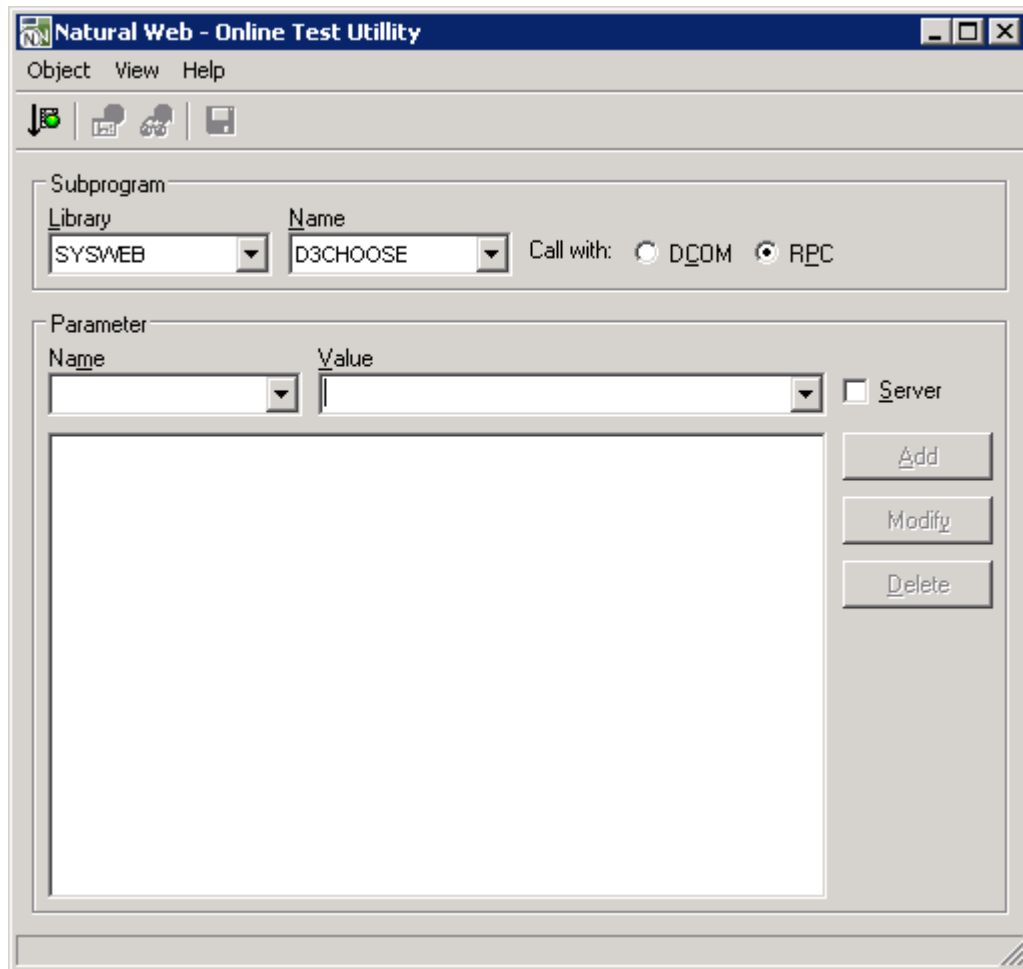
- 3 オプション：パラメータを追加します。
- 4 RPC または DCOM を選択します。
- 5 実行ボタンをクリックします。
- 6 [Result...] ボタンまたは [Browse...] ボタンをクリックして結果を表示します。

サポートされているコンテンツタイプ

次のコンテンツタイプがテストユーティリティでサポートされています。

コンテンツタイプ	拡張子
"application/rtf"	"rtf"
"application/powerpoint"	"ppt"
"application/msword"	"doc"
"application/excel"	"xls"
"text/html"	"htm"
"text/plain"	"txt"
"text/xml"	"xml"
"text/richtext"	"rtf"

上記以外のコンテンツタイプが必要な場合は、サブルーチン HTML2CONTENT-TYPE (SYSWEB/W3CO2EXT) を変更し、ニーズに合わせて変換テーブルを拡張します。



入力／出力フィールド

フィールド	説明
Subprogram : Library/Name	ライブラリおよび目的のサブプログラムの名前を指定できます。指定可能なライブラリおよびサブプログラムは、自動的にライブラリワークスペースから取得されて選択ボックスにリストされます。
Server	名前／値ペアがサーバー変数の場合は、このトグルボタンをオンにする必要があります。すべてのステータスは、再度変更するまで継続されます。
Interface	通信形式として DCOM または RPC を選択できます。DCOM の場合は、まずクラスを登録する必要があります。 デフォルト：RPC
Parameters : Name/Value	ここでは、必要な名前／値ペアをサブプログラムから入力できます。これらをパラメータリストに移動するには、[Add] ボタンをクリックします。エントリを変更するには、[Modify] ボタンをクリックします。&、=、% を置き換える必要はありません。この処理は WEB-ONL プログラムによって実行されます。サーバーパラメー

フィールド	説明
	<p>タを使用する場合は、パラメータをパラメータリストに追加する前に、 [Server] トグルボタンをオンにします。</p> <p>パラメータリストにすべての名前／値ペアが表示されます。 &、 =、 % は置き換えられます。 ペアを削除するには、項目を選択し、 [Delete] ボタンをクリックします。 選択したすべての項目が [Name] および [Value] フィールドに挿入されます。 ペアを変更するには、項目を選択し、 [Name] および [Value] フィールドで変更し、 [Modify] ボタンをクリックします。</p>

Object

Execute Subprogram

[オプション] ダイアログボックスで選択したエディタを開きます。

Save to Natural Text

返されたデータを、テキストタイプの Natural オブジェクトとして保存します。

Exit

ダイアログボックスを終了します。

View

Result

選択したサブプログラムを実行します。

Browser

[Options] ダイアログボックスで選択したブラウザを開きます。

Options...

[Options] ダイアログボックスを開きます。

Help

Contents

この HTML ベースのヘルプファイルを表示します。

About

プログラムの全般的な情報を表示します。

15 テストユーティリティ WEB-ONL3 (SYSWEB3)

▪ 前提条件	84
▪ アプリケーションの実行	84
▪ サポートされているコンテンツタイプ	85
▪ 入力／出力フィールド	86

テストユーティリティ **Web オンライン**は、Natural Web インターフェイスのコンポーネントです。HTTP サーバーを使用せずに、サブプログラムをローカルでチェックできます。Web ページの転送パラメータはテストユーティリティに転送され、ビジネスロジックに直接ポストされます。通信プラットフォームとしては、実際のリモート通信と同様に、RPC または DCOM を選択できます。処理の結果、目的の Web ページまたはエラーメッセージが表示されます。Web ページは、選択したブラウザまたはビューアーで表示できます。エラーメッセージが表示された場合でも、追加のテストルーチンを作成する必要はなく、ビジネスロジックをローカルで簡単にデバッグできます。リモートデバッグ不要

機能：

- ローカルでのアプリケーションチェック
- リモートデバッグ不要
- 簡単なエラーチェック
- 使い勝手の良いインターフェイスによる快適な操作性
- テストルーチンの追加作成不要

このセクションでは、次のトピックについて説明します。

前提条件

- さまざまなコンテンツタイプをサポートする Web ブラウザ。例えば、Microsoft Internet Explorer バージョン 5.0 以降。
- 任意の使用可能なテキストエディタ。

アプリケーションの実行

▶手順 15.1. パス調整を定義するには

- 1 メインダイアログボックスを開きます。
- 2 [Tools] メニューの [Options...] でブラウザとビューアーを選択します。
- 3 ブラウザ、ビューアー、およびワークファイルのパスを設定します。
- 4 [OK] ボタンをクリックします。

▶手順 15.2. アプリケーションを開始するには

- 1 ダイアログ **WEB-ONL3** を開始します。
- 2 ライブラリ名とサブプログラム名を選択します。

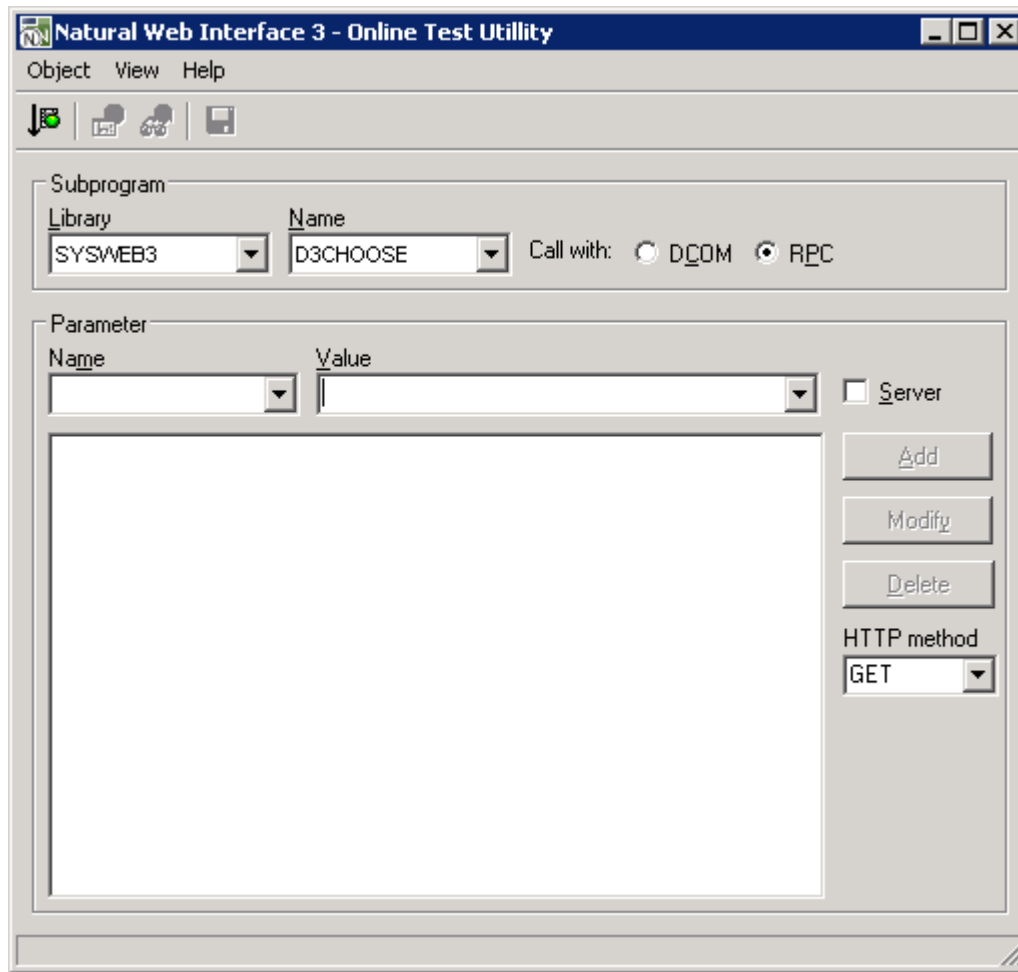
- 3 オプション：パラメータを追加します。
- 4 RPC または DCOM を選択します。
- 5 [Execute] ボタンをクリックします。
- 6 [Result...] ボタンまたは [Browse...] ボタンをクリックして結果を表示します。

サポートされているコンテンツタイプ

次のコンテンツタイプがテストユーティリティでサポートされています。

コンテンツタイプ	拡張子
"application/rtf"	"rtf"
"application/powerpoint"	"ppt"
"application/msword"	"doc"
"application/excel"	"xls"
"text/html"	"htm"
"text/plain"	"txt"
"text/xml"	"xml"
"text/richtext"	"rtf"

上記以外のコンテンツタイプが必要な場合は、サブルーチン HTML2CONTENT-TYPE (SYSWEB3/W3CO2EXT) を変更し、ニーズに合わせて変換テーブルを拡張します。



入力／出力フィールド

フィールド	説明
Subprogram : Library Name	ライブラリおよび目的のサブプログラムの名前を指定できます。指定可能なライブラリおよびサブプログラムは、自動的にライブラリワークスペースから取得されて選択ボックスにリストされます。
Interface	通信形式として DCOM または RPC を選択できます。DCOM の場合は、まずクラスを登録する必要があります。 デフォルト：RPC
Parameters: Name Value Server	サブプログラムから要求される名前／値ペアを入力できます。これらをパラメータリストに追加するには、[Add] ボタンをクリックします。エントリを変更するには、[Modify] ボタンをクリックします。&、=、% を置き換える必要はありません。この処理は WEB-ONL3 プログラムによって実行されます。サーバーパラメータを使用する場合は、パラメータをパラメータリストに追加する前に、[Server] トグルボタンをオンにします。

フィールド	説明
	<p>パラメータリストにすべての名前／値ペアが表示されます。&、=、%は置き換えられます。ペアを削除するには、項目を選択し、[Delete] ボタンをクリックします。選択したすべての項目が [Name] および [Value] フィールドに挿入されます。ペアを変更するには、項目を選択し、[Name] および [Value] フィールドで変更し、[Modify] ボタンをクリックします。</p> <p>Server : 名前／値ペアがサーバー変数の場合は、このトグルボタンをオンにする必要があります。すべてのステータスは、再度変更するまで継続されます。</p>
HTTP Method	<p>このドロップダウンリストでは、使用する HTTP 要求／送信メソッドを選択できます。</p> <p>■ HEAD GET 要求と似ていますが、応答本文がありません。これは、内容全体を取得せずに、応答ヘッダーに書かれたメタ情報を取得できるので便利です。</p> <p>■ GET 指定したリソースの表現を要求します。</p> <p>■ POST 特定したリソースからデータを送信します。データは要求の本文に組み込まれます。</p> <p>このメソッドを使用して、XML ファイルやバイナリデータ（例：グラフィック）などの異なるコンテンツタイプのデータを送信できます。</p> <p>このメソッドを指定すると、画面上で [Browse...] ボタンと [Binary] チェックボックスが追加で使用可能になります。</p> <p>[Browse...] ボタンはファイルを選択する場合に使用し、[Binary] チェックボックスはバイナリデータを送信する場合にオンにします。</p> <p>入力ファイルを指定せずにこのメソッドを指定した場合は、デフォルトの MIME タイプ "application/x-www-form-urlencoded" が設定されます。入力ファイルを使用した場合は、その入力ファイルのコンテンツタイプが使用されます。例えば、XML ファイルの場合は、コンテンツタイプは自動的に "text/xml" に設定されます。別の MIME タイプを入力フィールドに手動で指定できます。</p> <p>注意: 手動で設定した MIME タイプは、デフォルトの MIME タイプよりも常に優先されます。</p> <p>■ PUT 指定したリソースの表現をアップロードします。</p> <p>このメソッドを使用して、XML ファイルやバイナリデータ（例：グラフィック）などの異なるコンテンツタイプのデータを送信できます。</p> <p>このメソッドを指定すると、画面上で [Browse...] ボタンと [Binary] チェックボックスが追加で使用可能になります。</p> <p>[Browse...] ボタンはファイルを選択する場合に使用し、[Binary] チェックボックスはバイナリデータを送信する場合にオンにします。</p>

Object

Execute Subprogram

[Options] ダイアログボックスで選択したエディタを開きます。

Save to Natural Text

返されたデータを、テキストタイプの Natural オブジェクトとして保存します。

Exit

ダイアログボックスを終了します。

View

Result

選択したサブプログラムを実行します。

Browser

[Options] ダイアログボックスで選択したブラウザを開きます。

Options...

[Options] ダイアログボックスを開きます。

Help

Contents

この HTML ベースのヘルプファイルを表示します。

About

プログラムの全般的な情報を表示します。

16 プログラミングのヒント

▪ 小文字での編集	90
▪ 引用符とアポストロフィ	90
▪ 値で定義された変数	91
▪ リソースへのアクセス	91
▪ 定数値	91
▪ 新しいページの作成	92
▪ DCOM/RPC	92

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、考慮すべき点があいくつかあります。

- 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は、指定のNaturalライブラリへのダイナミックログオンをサポートします。
- 現在、ライブラリはクラスに同等であり、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での呼び出しは準備ができています。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必要があるだけです。
- Natural ライブラリのクラスを生成するには、Web インターフェイスプラグインの **Class Generation** を使用します。

17 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	94
▪ ユーザー定義エラーページの作成	95
▪ ユーザー定義エラーページ XML スタイルの作成	95
▪ 英数字から HTML への変換	96
▪ 英数字から URL への変換	96

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 17.1. トランスポート送信バッファの幅を変更するには

- 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 17.2. 受信データバッファの幅を変更するには

- 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 17.3. 戻りページを変更するには

- 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 または SYSWEB3/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 または SYSWEB3/W3ERRTMP) をアンカタログします。
- 2 サブルーチン (SYSWEB/W3ERXTMP または SYSWEB3/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 への変換のために、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 または SYSWEB3/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。

18 デモアプリケーション - JavaScript なし

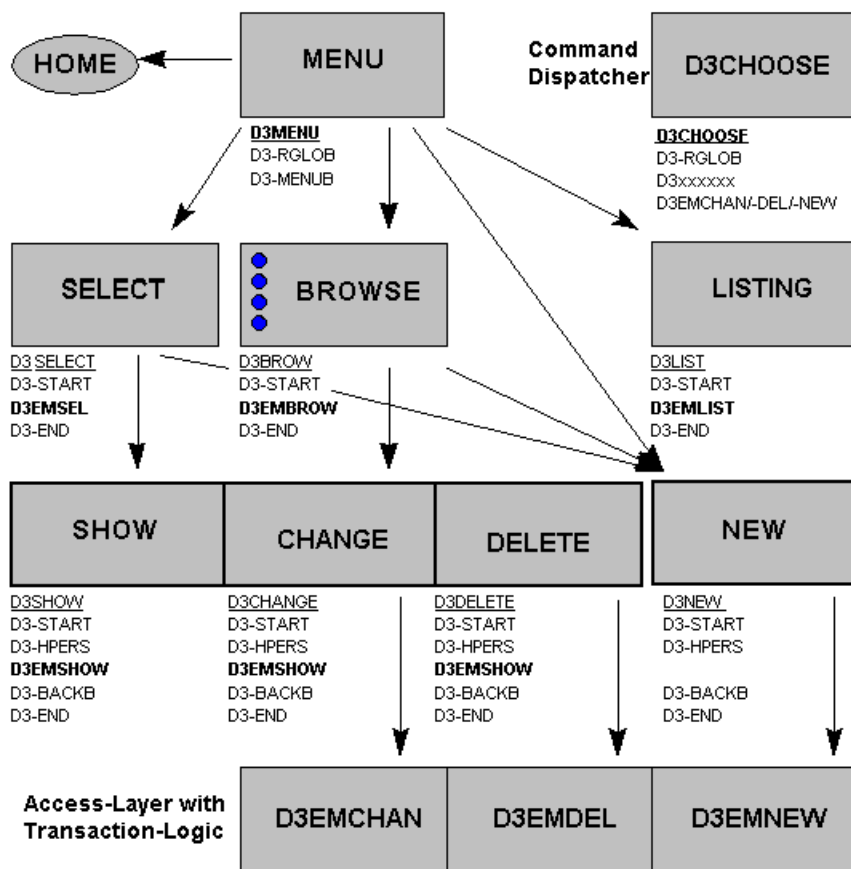
■ ビジネス要件	98
■ 設計に関する決定	99
■ ライブラリ、モジュール、および命名規則	99
■ デモアプリケーションの開始	100
■ Natural Web インターフェイスオンラインマニュアルの開始	100
■ 要件	100

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は、NaturalWeb インターフェイスの使用とプログラミングを示します。機能には、以下の図に示されている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が含まれます。

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 を基準にしています。



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く、サブミット用に事前定義されたフォントやボタンしかない（デフォルトボタンがない）などです。

そのため、デモアプリケーションでは以下を使用します。

- サブミットボタン付きのフォーム。
- 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。
- 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）の使用。
- 処理ルールの実装での VB/JAVA Scripts の不使用。
- 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）。
- 著作権上の理由によるグループ／男性／女性の標準画像。

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

SYSWEB3

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 または SYSWEB3 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`

SYSWEB3 でデモアプリケーションを呼び出す URL の例：

`http://yourserver/yourcgi/sysweb3/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 または SYSWEB3 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`

SYSWEB3 でデモアプリケーションを呼び出す URL の例：

`http://yourserver/yourcgi/sysweb3/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 EMPLOYEEES

デモアプリケーションを有効にするには、ライブラリ SYSWEB (または SYSWEB3) 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。

19 デモアプリケーション - JavaScript 使用

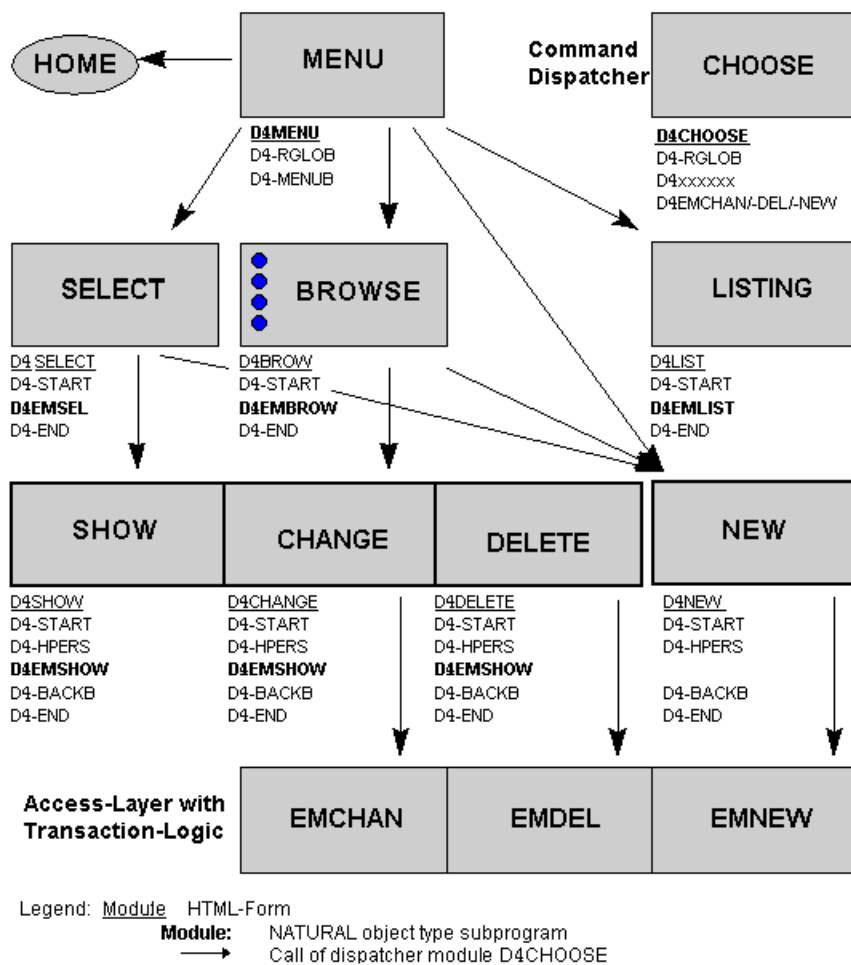
▪ ビジネス要件	102
▪ 設計に関する決定	103
▪ デモアプリケーションの開始	103
▪ 要件	103

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は、NaturalWeb インターフェイスの使用とプログラミングを示します。機能には、以下の図に示されている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が含まれます。

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 を基準にしています。



設計に関する決定

アート Web 設計の状態を使用します。

- 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。
- 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) の使用。
- 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 または SYSWEB3 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 EMPLOYEES 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 (または SYSWEB3) のプログラム D4* に対して CATALL 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。

20 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 106

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	No elements defined.	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	Can only be used inside a FORM tag.	このタグは FORM タグ内のみで使用できます。	FORM を H3-OPEN-FORM で初期化してください。
NWW9004	A FORM tag without ACTION is not allowed.	FORM ごとに 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	LI tag outside a list not allowed.	LI はリスト内に置く必要があります。	LIST を H3-OPEN-FORM で初期化してください。
NWW9006	List nested too deep: ...	リストでサポートされるレベルの数は 10 です。	レベルを減らしてください。
NWW9007	Radio Button Group has no name.	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	Element ... has no name.	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。	名前を追加します。
NWW9009	Textarea has no name.	テキストエリアを生成するには名前が必要です。	名前を追加します。

21 Natural Web インターフェイス SYSWEB から SYSWEB3 への移行

このセクションでは、Natural Web インターフェイス (SYSWEB) 用に作成されたプログラムを新しいインターフェイス (SYSWEB3) で実行できるように更新する方法について、順を追って説明します。

1. 古い nww* プログラムの代わりに、新しいインターフェイスプログラム nww3* を使用する必要があります。適切なプログラムを HTTP サーバーにコピーし、HTTP サーバーを構成します。
2. nww3*.ini ファイルの設定を修正します。
3. nww* ではなく nww3* を指すように、外部 URL を修正します。



Note: nww3* インターフェイスプログラムの名前を変更することも可能ですが、新しいインターフェイスプログラムは SYSWEB3 で格納された Natural サブプログラムでのみ動作し、古いインターフェイスプログラムは SYSWEB で格納された Natural サブプログラムでのみ動作します。

4. Natural RPC サーバーを実行している場合

- Windows または UNIX では、Natural Security を使用しない場合は、NATPARM で steplib 設定を変更します。steplib SYSWEB を steplib SYSWEB3 に置き換えます (または、新しい別の RPC サーバーを使用します)
- メインフレームシステムでは、Natural Security を使用しない場合は、サーバー起動ログオンを SYSWEB から SYSWEB3 に変更します (または別の新しい RPC サーバーを使用します)。
- Natural Security を使用する場合は (すべてのプラットフォーム)、Natural Security で steplib 設定を変更します。steplib SYSWEB を steplib SYSWEB3 に置き換えます (または、新しい別の RPC サーバーを使用します)
- EntireX Broker 設定を確認します。転送が必要な最大ページ長によっては、NUM-COMBUF、NUM-LONG-BUFFER、MAX-MSG の設定を拡張する必要があります。

5. DCOM サーバーを実行している場合 (Windows のみ)

- 最初にクラス (通常はライブラリ名に従った名前) の登録を解除します。
- クラス (通常はライブラリ名に従った名前) と関連 LDA (通常は L という名前) を削除します。
- Web インターフェイスプラグインを使用して、新しい LDA (新しい GUI) で SYSWEB3 の新しいクラスを生成します。
- 新しいクラスを登録します。

6. SYSWEB ではなく SYSWEB3 を指すように、URL、読み込むワークファイル、テンプレートなどを修正します。



Note: SYSWEB3 を SYSWEB に名前変更した場合は、アプリケーションが動作する可能性はありますが、SYSWEB3 付属の内部プログラムが正常に動作せずに不適切な結果や予期しない結果になることがあります。

7. SYSWEB の代わりに SYSWEB3 を `steplib` として使用して、Web インターフェイスプログラムを参照するすべてのプログラムを再カタログします。

8. これで、すべてのプログラムは以前に SYSWEB で実行していたときと同じように動作します。

22 Web インターフェイスプラグイン

- 開始する前に 110
- Web インターフェイスプラグインコマンドの呼び出し 110
- Web インターフェイスプラグイン機能 112

このセクションでは、次のトピックについて説明します。

開始する前に

Web インターフェイスプラグインは、Natural スタジオのオプションのプラグインユニットです。操作するには、先に Web インターフェイスプラグインをプラグインマネージャで有効にする必要があります。有効にする手順については、「プラグインマネージャ」を参照してください。

Web インターフェイスプラグインインターフェイス


Web インターフェイスプラグインを有効にすると、Natural スタジオインターフェイスは次のように変化します。

- [ツール] メニューに Web インターフェイスプラグイン関連のコマンドが表示されます。
- 頻繁に使用するコマンドがツールバーに追加されます。

使用できるコマンドは、作業コンテキストによって異なります。


Web インターフェイスプラグインコマンドの呼び出し

▶手順 22.1. メインメニューからプログラム生成コマンドを呼び出すには

- メインメニューから、[ツール] > [開発ツール] >  [Web Interface Program Generation] を選択します。


使用可能なコマンドの詳細については、「[プログラム生成](#)」を参照してください。

▶手順 22.2. メインメニューからクラス生成コマンドを呼び出すには

- メインメニューから、[ツール] > [開発ツール] >  [Web Interface Class Generation] を選択します。


使用可能なコマンドの詳細については、「[クラスの生成](#)」を参照してください。

▶手順 22.3. メインメニューからテストユーティリティコマンドを呼び出すには

- メインメニューから、[ツール] > [開発ツール] >  [Web Interface Test Utility] を選択します。

使用可能なコマンドの詳細については、「[オンラインテストユーティリティ](#)」を参照してください。



▶ **手順 22.4. メインメニューから Web インターフェイスオプションを開くには**

- メインメニューから、[ツール] > [開発ツール] >  [Web Interface Options] を選択します。

使用可能なコマンドの詳細については、「[オプション](#)」を参照してください。


▶ **手順 22.5. Web インターフェイスプラグインコマンドを呼び出し、ツールバーボタンを使用するには**

- 生成ウィザードの場合：

-  生成処理に使用する HTML ページを選択します。
-  選択した HTML ファイルを外部エディタで開きます。


Or:

テストユーティリティの場合：

-  エディタを開きます。

プログラムを実行してなくて、サブプログラムのライブラリまたは名前を変更していなければ、無効になっています。

[Options] ダイアログボックスでエディタを選択できます。

-  ブラウザを開きます。

プログラムを実行してなくて、サブプログラムのライブラリまたは名前を変更していなければ、無効になっています。

[Options] ダイアログボックスでブラウザを選択できます。

Web インターフェイスプラグイン機能

プログラム生成



Note: プログラム生成ウィザードはメインフレームシステムには適用されません。

このセクションでは、プログラム生成ウィザードの使用について説明します。このウィザードは、基本的な Web インターフェイスプログラムの生成と、Natural Web インターフェイスで HTML テンプレートを使用するプログラムの生成を可能にするプラグインです。

基本的な生成では、Web インターフェイスを使用してインターネットから呼び出されるサブプログラムに必要な部分を生成できます。その後、特定のコーディングを追加できます。

テンプレート生成では、前もって設計された HTML ページを扱います。これらの HTML ページは、リソースディレクトリからロードされます。その後、特定の部分をユーザー独自の部分に置換できます。プログラムジェネレータは、これらの HTML にページを読み込み、置換する部分（特殊文字でマークされている部分）を検索し、マップのみの出力と同等に使用できる外部サブルーチンを生成します。

このセクションでは、次のトピックについて説明します。

- [変換プログラムの使用](#)
- [置換文字列の挿入](#)
- [オプション](#)
- [View](#)
- [クラスの生成](#)
- [オンラインテストユーティリティ](#)

変換プログラムの使用

基本 Web ページが編集ツールを使用して設計されている場合は、Web から呼び出せる Natural サブプログラムにそのページを組み込むのは多少手間が掛かります。

Web ウィザードは、HTML ページを入力として使用するダイアログボックスです。ここでは、Natural Web インターフェイスを使用して Natural Web サーバーエクステンションから呼び出せる Natural サブプログラムを生成したり、出力を生成するために呼び出せるサブルーチンを生成したりします。

Web ウィザードの基本的な生成では、Natural Web インターフェイスを使用して Web サーバーエクステンションから呼び出せるスタンドアロンの Natural サブプログラムを生成できます。

The screenshot shows a dialog box titled "Natural Web Interface - Program Generation". It contains several sections for configuring the generation process:

- Type of Generation:** Radio buttons for "Basic", "Stand-alone", "Transformation" (selected), and "Template".
- Use local file:** A checkbox that is currently unchecked.
- Input File:** A section with three dropdown menus: "Library" (set to "SYSPLWEB"), "Extension" (set to "Relevant Files (*.html/*.asp)"), and "File Name" (empty).
- Generated Natural object:** A section with four dropdown menus: "Library" (set to "SYSPLWEB"), "Object type" (set to "Subroutine"), "Object name" (empty), and "Subroutine name" (empty).

At the bottom of the dialog are three buttons: "Generate", "Cancel", and "Help".

Web から直接呼び出される基本的なサブプログラムの生成

▶手順22.6. Web から直接呼び出されるサブプログラム／サブルーチンを生成するには、次の手順に従います。

- 1 生成のタイプとして [Basic] を選択します。
- 2 生成される Natural オブジェクトを選択します。
- 3 生成を開始します。
- 4 このサブプログラムを初めて生成し、生成したサブプログラムを DCOM を使用して呼び出す場合、DCOM クラスを再生成します（「[クラス](#)の生成」を参照してください）。
- 5 生成後、このページをインターネットから呼び出せます。ただし、このページにはデータが設定されていないのでページの内容は空です。

基本生成の例

生成される Natural サブプログラム（インターネットから直接呼び出し可能）：

```
0010 * ----- GENERATED BY NATURAL WEB INTERFACE
* Library .....: SYSPLWEB
* Source Name .: BASIC
* -----
DEFINE DATA
PARAMETER USING W3PARAM
LOCAL USING W3CONST
* ----- PRIVATE VARIABLES -----
* LOCAL
* 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
* --- READ ENVIRONMENT ---
* PERFORM W3READ-ENVIRONMENT-DYNAMIC 'varname' ' ' W3VALUE
* set default value
* IF *length(W3VALUE) = 0 THEN
*   W3VALUE := ??
* END-IF
* ----- HEADER FOR SERVER -----
* PERFORM W3CONTENT-TYPE 'text/html'
*
*
* Add your individual coding using W3* subroutines or
* call your own subroutines.
*
* ----- END HTTP -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
```

```
*
END
```

Web から直接呼び出されるスタンドアロンサブプログラムの生成

▶手順 22.7. Web から直接呼び出されるサブプログラムを生成するには、次の手順に従います。

- 1 生成のタイプとして [Standalone] を選択します。
- 2 生成される Natural オブジェクトを選択します。
- 3 生成を開始します。
- 4 このサブプログラムを初めて生成し、生成したサブプログラムを DCOM を使用して呼び出す場合、DCOM クラスを再生成します（「[クラスの生成](#)」を参照してください）。
- 5 生成後、Natural Web インターフェイスを呼び出してページを表示できます。

スタンドアロン生成の例

生成される Natural サブプログラム（インターネットから直接呼び出し可能）：

```
* ----- GENERATED BY NATURAL WEB INTERFACE
* Library .....: SYSPLWEB
* Source Name .: ALONE
* -----
DEFINE DATA
PARAMETER USING W3PARM
LOCAL USING W3CONST
* ----- PRIVATE VARIABLES -----
LOCAL
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'
*
* --- READ ENVIRONMENT ---
* PERFORM W3READ-ENVIRONMENT-DYNAMIC 'varname' ' ' W3VALUE
* set default value
* IF *length(W3VALUE) = 0 THEN
* W3VALUE := ??
* END-IF
```

```
* --- WRITE THE HEAD OF THE DOCUMENT ---
PERFORM W3TEXT "<!DOCTYPE 'HTML PUBLIC-//W3C//DTD HTML 3.2//EN'>"-
'<html>' -
'<head>' -
"<meta http-equiv='Content-Type' content='"-
"text/html; charset=iso-8859-1'>"-
'<title>SYSPLWEB/TEST</title>' -
'</head>'
* --- WRITE THE BODY OF THE DOCUMENT ---
PERFORM W3TEXT '<body>' -
'<h2>SYSPLWEB/TEST</h2>' -
'<hr>'
*
PERFORM W3TEXT '<p>This is your output</p>'
*
COMPRESS '<hr>generated:' *DATE *TIME INTO W3VALUE
PERFORM W3TEXT W3VALUE
* --- END THE BODY OF THE DOCUMENT ---
PERFORM W3TEXT '</body>' -
'</html>'
*
* ----- END HTTP -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
*
END
```

Natural タグを使用したサブプログラム／サブルーチンの生成

▶**手順22.8.** Web から直接呼び出されるサブプログラム／サブルーチンを生成するには、次の手順に従います。

- 1 生成のタイプとして [Transformation] を選択します。
- 2 HTML タイプの入力ファイルを選択します。
- 3 エディタまたはブラウザを使用して、選択した HTML ページを表示できます。
- 4 生成先の Natural ライブラリを選択します。
- 5 生成するオブジェクトタイプを選択します。
- 6 生成される Natural オブジェクトを選択します。
- 7 生成を開始します。
- 8 このサブプログラムを初めて生成し、生成したサブプログラムを DCOM を使用して呼び出す場合、DCOM クラスを再生成します（「[クラスの生成](#)」を参照してください）。
- 9 生成後、Natural Web インターフェイスを呼び出してページを表示できます。

Natural タグの挿入

HTML ページに Natural コーディングを直接指定できます。生成後にプログラムを追加で変更する必要はありません。

HTML2NAT ダイアログでは <NATURAL> タグが認識されます。<NATURAL> と </NATURAL> で囲まれたすべての行は、生成された Natural ソースオブジェクトにそのままコピーされます。

表示

```
<NATURAL> </NATURAL>
```

以下に参考情報を示します。

- 属性の DATA、LDA、GDA、SUB、NOT
- コメントタグ
- ASP 形式のスクリプトコマンド
- 追加のスクリプト命令
- シンプル生成の例
- シンプル生成の例 (Natural タグを使用)

属性の DATA、LDA、GDA、SUB、NOT

以下の属性を使用して、プログラム内で移動したりプログラムから除外したりするコーディングセクションを定義します。

属性	説明
DATA LDA	<NATURAL DATA> または <NATURAL LDA> は、定義されたセクションをプログラムの DEFINE DATA LOCAL 部分に移動します。
GDA	<NATURAL GDA> は、定義されたセクションをプログラムの DEFINE DATA GLOBAL 部分に移動します。
SUB	<NATURAL SUB> は、定義されたセクションをプログラムの末尾に移動します。これにより、インラインサブルーチンを指定できます。
NOT	<NATURAL NOT> は、定義されたセクションをプログラムから除外します。これにより、プログラムによって生成されたページの一部のデザインを指定できます。

コメントタグ

コーディング内の特定のセクションを非表示にするには、コメントタグ `<!-- -->` を使用します。コメントタグと `<NATURAL NOT>` を使用すれば、定義済みのページを標準のブラウザで表示できます。この方法は、ページを指定したりページの一部をダイナミックに置き換えたりするのに役立ちます。

ASP 形式のスクリプトコマンド

`<NATURAL>` と `</NATURAL>` だけでなく、テキストとの区別に `<% および %>` デリミタを使用する ASP (Active Server Page) 形式のスクリプトコマンドも使用できます。

追加のスクリプト命令

Natural のサブプログラムを作成する場合は、次の Natural 固有の命令を使用する必要があります。

出力命令：`<%= ... %>`

`<% PERFORM W3HTML ... %>` タグの短縮形です。

サブプログラム命令：`<%SUB ... %>`

`<NATURAL SUB> ... </NATURAL>` タグと同等です。

グローバルデータエリア命令：`<%GDA ... %>`

`<NATURAL GDA> ... </NATURAL>` タグと同等です。

命令：`<%LDA ... %>`

`<NATURAL LDA> ... </NATURAL>` タグと同等です。

除外命令：`<%NOT ... %>`

`<NATURAL NOT> ... </NATURAL>` タグと同等です。

処理命令：`<%@ LANGUAGE=NATURAL %>`

使用言語が Natural であることを示します。

シンプル生成の例 1

HTML ドキュメント：

```
<HTML><HEAD><TITLE>
Example1 genNat
</TITLE></HEAD><BODY><H2>
Example1 genNat
</H2><HR>
```



```
<P>This is for your output
</BODY></HTML>
```

生成される Natural サブプログラム :

```
* ----- GENERATED BY NATURAL WEB INTERFACE
* File .....:
  E:\SAG\Natural\6.2\Fnat\SYSWEB\RES\example1.html
* Library .....: SYSWEB
* Source Name ...: EXAMPLE1
* Crunch Lines...: 1
* Save Source....: 1
* Line Length....: 128
* Long Constants.: 1
* -----
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
* ----- INITIALIZE HTTP API -----
PERFORM W3INIT ##RPC
* ----- HEADER FOR SERVER -----
PERFORM W3CONTENT-TYPE 'text/html'
*
* ----- MAIN PROGRAM -----
PERFORM
  W3TEXTLINE '<HTML><HEAD><TITLE>'
PERFORM W3TEXTLINE 'Example genNat'
PERFORM
  W3TEXTLINE '</TITLE></HEAD><BODY><H2>'
PERFORM W3TEXTLINE 'Example genNat'
PERFORM W3TEXTLINE '</H2><HR>'
PERFORM W3TEXTLINE '<P>This is for your output'
PERFORM W3TEXTLINE '</BODY></HTML>'
* ----- END HTTP API -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
*
*
```

```
* ----- SUBROUTINES -----  
END
```

シンプル生成の例 2 (Natural タグを使用)

HTML ドキュメント :

```
<HTML><HEAD><TITLE>  
Example2 genNat  
</TITLE></HEAD><BODY><H2>  
Example2 genNat  
</H2><HR>  
<P>This is for your output  
<HR>  
<P>generated at:  
<NATURAL NOT>  
Time/Date  
</NATURAL>  
<NATURAL><!--  
  PERFORM DOTIME  
--></NATURAL>  
<NATURAL SUB><!--  
DEFINE SUBROUTINE DOTIME  
  COMPRESS *TIME *DATE INTO #VALUE  
  PERFORM W3TEXTLINE #VALUE  
END-SUBROUTINE  
--></NATURAL>  
<NATURAL DATA><!--  
1 #VALUE (A30)  
--></NATURAL>  
</BODY></HTML>
```

生成される Natural サブプログラム :

```
* ----- GENERATED BY NATURAL WEB INTERFACE  
* File .....: E:\SAG\Natural\6.2\Fnat\SYSWEB\RES\example2.html  
* Library .....: SYSWEB  
* Source Name ...: EXAMPLE2  
* Crunch Lines...: 1  
* Save Source....: 1  
* Line Length....: 128  
* Long Constants.: 1  
* -----  
DEFINE DATA  
PARAMETER USING W3PARM  
LOCAL USING W3CONST  
1 #VALUE (A30)  
* ----- PRIVATE VARIABLES -----
```

```
1 W3VALUE (A250)
END-DEFINE
*
* ----- ERROR HANDLER -----
ON ERROR
  PERFORM W3ERROR ##W3ERROR
  PERFORM W3END ##RPC
  ESCAPE ROUTINE
END-ERROR
* ----- INITIALIZE HTTP API -----
PERFORM W3INIT ##RPC
* ----- HEADER FOR SERVER -----
PERFORM W3CONTENT-TYPE 'text/html'
*
* ----- MAIN PROGRAM -----
PERFORM W3TEXTLINE '<HTML><HEAD><TITLE>'
PERFORM W3TEXTLINE 'Example2 genNat'
PERFORM W3TEXTLINE '</TITLE></HEAD><BODY><H2>'
PERFORM W3TEXTLINE 'Example2 genNat'
PERFORM W3TEXTLINE '</H2><HR>'
PERFORM W3TEXTLINE '<P>This is for your output'
PERFORM W3TEXTLINE '<HR>'
PERFORM W3TEXTLINE '<P>generated at:'
  PERFORM DOTIME
PERFORM W3TEXTLINE '</BODY></HTML>'
* ----- END HTTP API -----
PERFORM W3END ##RPC
* ----- END MAIN PROGRAM -----
*
*
* ----- SUBROUTINES -----
DEFINE SUBROUTINE DOTIME
  COMPRESS *TIME *DATE INTO #VALUE
  PERFORM W3TEXTLINE #VALUE
END-SUBROUTINE
END
```



Note: Natural プログラムの構文は、変換時にはチェックされません。

Web から直接呼び出されるテンプレートを使用したサブプログラム／サブルーチンの生成

▶ **手順 22.9.** Web から直接呼び出されるテンプレートを使用してサブプログラム／サブルーチンを生成するには、次の手順に従います。

- 1 生成のタイプとして [Template] を選択します。
- 2 HTML タイプの入力ファイルを選択します。
- 3 エディタまたはブラウザを使用して、選択した HTML ページを表示できます。
- 4 生成するオブジェクトタイプを選択します。

- 5 生成される Natural オブジェクトを選択します。
- 6 生成を開始します。
- 7 このサブプログラムを初めて生成し、生成したサブプログラムを DCOM を使用して呼び出す場合、DCOM クラスを再生成します（「[クラスの生成](#)」を参照してください）。
- 8 生成後、Natural Web インターフェイスを呼び出してページを表示できます。

置換文字列の挿入

HTML ページに置換文字列を直接指定する必要があります。置換文字列の先頭と末尾には特定の文字（例：\$）を指定する必要があります（「[オプション](#)」を参照してください）。文字列の名前（内容）は、変数名に関する Natural ルールに従う必要があります。従わないと、サブルーチンで格納できません。

置換文字列の名前の先頭が "HTML" の場合、"<" や ">" などの保存されていない文字がランタイムでの置換処理時に置き換えられます。

次の接頭辞が、ランタイムでの自動変換用に実装されています。

- HTML
- URL
- XML

詳細については、サブルーチン [W3REPLACE-AT-OUTPUT](#) に関するドキュメントを参照してください。

テンプレート生成の例

HTML ドキュメント：

```
<HTML>
<HEAD>
  <TITLE>Template Processing</TITLE>
</HEAD>
<BODY>
<H2>
  Template Processing
</H2>
<P>
  <HR>
<TABLE BORDER="0">
<TR><TD>Log-Time:</TD><TD>$log$<TD></TR>
<TR><TD>HTTPs Extension:</TD><TD>$html-ext$<TD></TR>
<TR><TD>Web Interface:</TD><TD>$html-ver$<TD></TR>
</TABLE>
<P>
```

```
<TABLE BORDER='0' WIDTH='100%' CELSPACING='0' CELLPADDING=5>
  <TR BGCOLOR='#00cc66'>
    <TD>$prog$ - $log$</TD>
    <TD ALIGN='RIGHT'>Natural</TD>
  </TR>
</TABLE>
</BODY></HTML>
```

生成される Natural サブルーチン（インターネットから呼び出されるサブプログラムから呼び出す必要がある）：

```
* ----- GENERATED BY NATURAL WEB INTERFACE
* File .....: E:\SAG\Natural\6.2\Fnat\SYSWEB\RES\templ.html
* Library .....: SYSWEB
* Source Name .: TEMPL
* Delimiter ...: $
* -----
DEFINE DATA PARAMETER
1 log                               (A) DYNAMIC BY VALUE
1 html-ext                          (A) DYNAMIC BY VALUE
1 html-ver                          (A) DYNAMIC BY VALUE
1 prog                              (A) DYNAMIC BY VALUE
END-DEFINE
*
*
DEFINE SUBROUTINE e3templm
*
* ----- HEADER FOR SERVER -----
PERFORM W3CLEAR
PERFORM W3CONTENT-TYPE 'text/html'
* ----- MAIN PROGRAM -----
* --- LOAD THE HTML TEMPLATE ---
PERFORM W3LOAD-RESOURCE ' ' 'e3templ.html'
*
* --- REPLACE PLACEHOLDER ---
PERFORM W3REPLACE-AT-OUTPUT ' ' '$log$' log
PERFORM W3REPLACE-AT-OUTPUT 'HTML' '$ext$' ext
PERFORM W3REPLACE-AT-OUTPUT 'HTML' '$ver$' ver
PERFORM W3REPLACE-AT-OUTPUT ' ' '$prog$' prog
* ----- END MAIN PROGRAM -----
*
END-SUBROUTINE
```

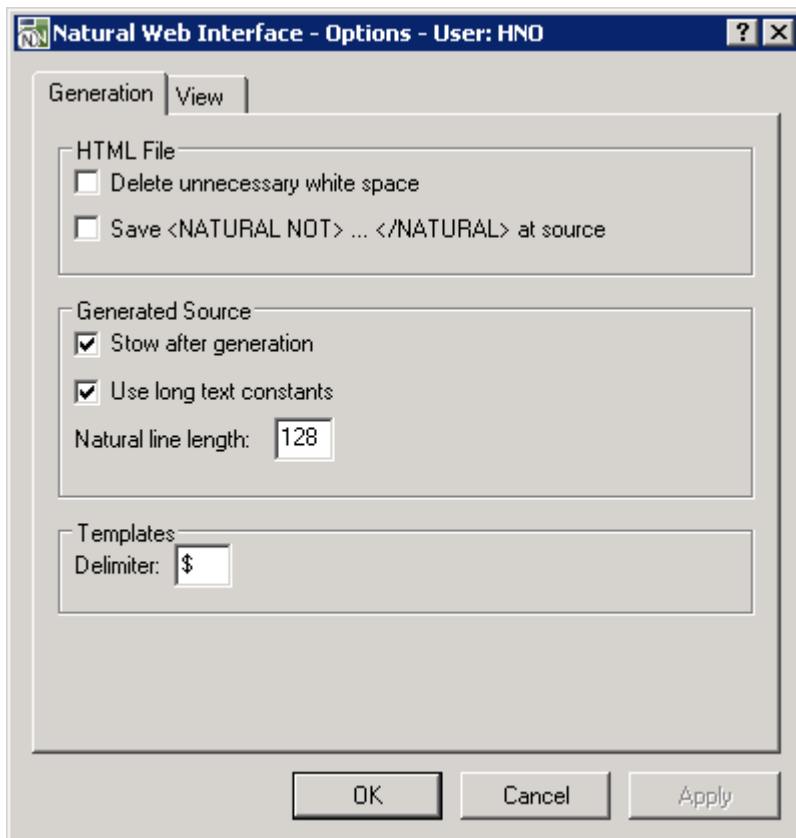
```
*  
END
```

生成される Natural サブプログラム（インターネットから直接呼び出し可能）：

```
* ----- GENERATED BY NATURAL WEB INTERFACE  
* File .....: E:\SAG\Natural\6.2\Fnat\SYSWEB\RES\templ.html  
* Library .....: SYSWEB  
* Source Name .: TEMPL  
* Delimiter ...: $  
* -----  
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'  
*  
* ----- MAIN PROGRAM -----  
* --- LOAD THE HTML TEMPLATE ---  
PERFORM W3LOAD-RESOURCE 'SYSWEB'  'e3templ.html'  
*  
* --- REPLACE PLACEHOLDER ---  
PERFORM W3REPLACE-AT-OUTPUT ' ' '$log$' 'replace-string-1'  
PERFORM W3REPLACE-AT-OUTPUT 'HTML' '$ext$' 'replace-string-2'  
PERFORM W3REPLACE-AT-OUTPUT 'HTML' '$ver$' 'replace-string-3'  
PERFORM W3REPLACE-AT-OUTPUT ' ' '$prog$' 'replace-string-4'  
* ----- END HTTP -----  
PERFORM W3END ##RPC  
* ----- END MAIN PROGRAM -----
```

*
END

オプション

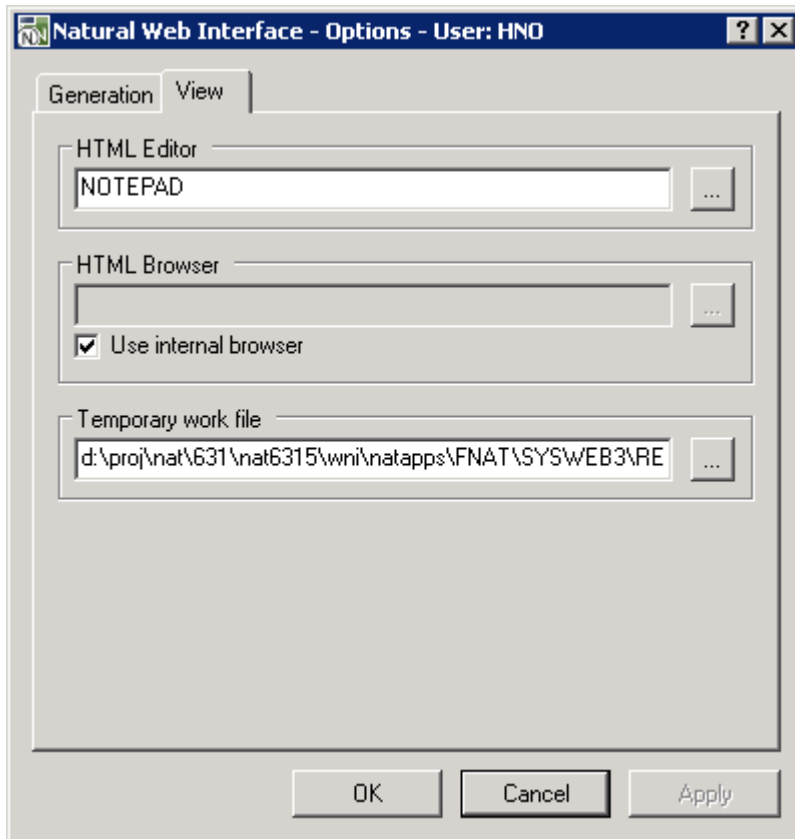


入力／出力フィールド

フィールド	説明
Delete unnecessary white space	オンにすると、空白、改行、タブなどの複数のスペース文字が減らされて単一のスペースに変換されます。<PRE>、<TEXTAREA>、<SCRIPT>などの特殊 HTML タグの場合は、スペースは減らされません。 デフォルト値：オフ
Save <NATURAL NOT> ... <NATURAL> at source	オンにすると、<NATURAL NOT> タグの内容は原則的には Natural ソースに生成されません。このオプションでは、<NATURAL NOT>の内容はコメントとして Natural ソースに生成されます。 デフォルト値：オフ
Stow after generation	オンにすると、生成が正常に完了すると生成されたプログラムは格納されます。 デフォルト値：オン

フィールド	説明
Use long text constants	より良いパフォーマンスのために 253 文字以上のテキスト定数を生成します。 デフォルト値：オン
Natural line length	生成される Natural ソース行の長さで、最小値は 20、最大値は 246 です。 デフォルト値：platform dependant
Delimiter	置換文字列のためのデリミタ文字列。 デフォルト値：\$

View



以下に参考情報を示します。

- 入力／出力フィールド
- ボタン

入力／出力フィールド

フィールド	説明
HTML Editor	HTML ページのソースを編集するために使用する外部プログラムです。 デフォルト値：NOTEPAD
HTML Browser	HTML ページを表示するために使用する外部プログラムです。 デフォルト値：Microsoft Web ブラウザ ActiveX コントロール
Use internal browser	HTML ページを表示するために使用する外部プログラムです。 独自のブラウザを選択するには、このボックスをオフにします。 デフォルト値：オン

フィールド	説明
Temporary work file	HTML ブラウザにデータを表示するために使用するデフォルトの出力ファイルです。 デフォルト値：C:\Temp\web-out

ボタン

OK	変更内容を保存してダイアログボックスを終了します。
Cancel	変更内容を保存せずにダイアログボックスを終了します。
Apply	現在の入力を保存します。

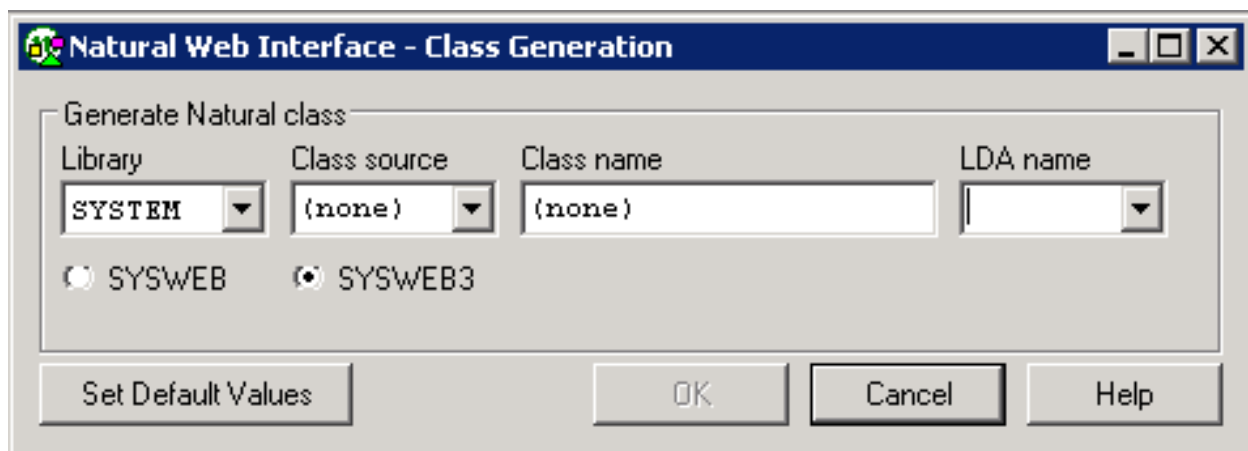
クラスの生成

RPC の代わりに DCOM を使用して Natural Web インターフェイスサブプログラムを呼び出す場合は、DCOM クラスが必要です。このクラスには、メソッドとして Natural Web インターフェイスのすべての関連 Natural サブプログラムが含まれています。

プログラム HTML to Natural は、指定されたクラスを自動的に生成します。生成されたクラスを格納するには、DCOM オブジェクトのグローバルユニーク ID (GUID) を指定するためにローカルデータエリア (LDA) が必要です。LDA の名前は L で始まり、[ライブラリ名](#)の最初の 7 文字が続きます。

以下に参考情報を示します。

- [入力／出力フィールド](#)
- [ラジオボタン](#)
- [ボタン](#)
- [ライブラリ SYSWEB の例](#)
- [ライブラリ SYSWEB3 の例](#)



入力／出力フィールド

フィールド	説明
Library	スキャンするライブラリの名前。
Class source	クラスソースの名前。 [Class source] には、ライブラリと同じ名前を選択することをお勧めします。
クラス名	後でインターネットから呼び出せるクラスの名前。 [Class name] には、クラスが生成されるライブラリと同じ名前を選択することをお勧めします。
LDA name	クラス ID と Natural Web インターフェイス ID の GUID を含む LDA の名前。 適用される命名規則については、後述の「 ライブラリ SYSWEB の例 」を参照してください。

ラジオボタン

ラジオボタン	説明
SYSWEB	ライブラリ SYSWEB で使用するクラスを生成する場合は、このオプションを選択します。
SYSWEB3	ライブラリ SYSWEB3 で使用するクラスを生成する場合は、このオプションを選択します。

ボタン

ボタン	説明
OK	クラスを生成してダイアログボックスを終了します。
Cancel	クラスを生成せずにダイアログボックスを終了します。
Set Default Values	このボタンは、ライブラリに関連するクラスが見つからない場合に使用可能になります。[Class source] と [Class name] のデフォルトが設定されます。必要な LDA は事前に生成しておく必要があります。

ライブラリ SYSWEB の例

LDA 名は LSYSWEB です。1 つ目の GUID には CLSID- にライブラリ名を後置した名前、2 つ目の GUID には IID-NATWEB という名前を付けます。


T コメント

*** データエリアの先頭 ***

X U 1 CLSID- **SYSWEB** A 36

X U 1 IID-NATWEB A 36

*** データエリアの末尾 ***

 **Caution:** クラスに新しい GUID を取得するために、LDA をコピーして名前を変更したり移動したりしないでください。LDA をコピーして名前を変更したり移動したりしても、

プリセット GUID は変更されません。これにより、大きな問題が発生する場合があります。

ライブラリ SYSWEB3 の例

LDA 名は LSYSWEB3 です。1つ目の GUID には CLSID- にライブラリ名を後置した名前、2つ目の GUID には IID-NATWEB3 という名前を付けます。


T コメント

*** データエリアの先頭 ***

X U 1 CLSID- SYSWEB3 A 36

X U 1 IID-NATWEB3 A 36

*** データエリアの末尾 ***

 **Caution:** クラスに新しい GUID を取得するために、LDA をコピーして名前を変更したり移動したりしないでください。LDA をコピーして名前を変更したり移動したりしても、プリセット GUID は変更されません。これにより、大きな問題が発生する場合があります。

オンラインテストユーティリティ

このテストユーティリティは NaturalWeb インターフェイスのコンポーネントです。HTTP サーバーを使用せずに、サブプログラムをローカルでチェックできます。Web ページの転送パラメータはテストユーティリティに転送され、ビジネスロジックに直接ポストされます。

通信プラットフォームとしては、実際のリモート通信と同様に、RPC または DCOM を選択できます。処理の結果、目的の Web ページまたはエラーメッセージが表示されます。Web ページは、選択したブラウザまたはビューアーで表示できます。エラーメッセージが表示された場合でも、追加のテストルーチンを作成する必要はなく、ビジネスロジックをローカルで簡単にデバッグできます。リモートデバッグは不要です。

機能：

- ローカルでのアプリケーションチェック
- リモートデバッグ不要
- 簡単なエラーチェック
- 使い勝手の良いインターフェイスによる快適な操作性
- テストルーチンの追加作成不要

Natural Web Interface - Test Utility

Subprogram

Library: SYSTEM Name: (none) Interface: DCOM RPC

Parameters

Name	Value
------	-------

Server

Add

Modify

Delete

HTTP method: POST

Browse...

Binary

Execute Cancel Help

以下に参考情報を示します。

- 前提条件
- アプリケーションの実行
- サポートされているコンテンツタイプ
- 入力／出力フィールド

■ ボタン

前提条件

- さまざまなコンテンツタイプをサポートする Web ブラウザ。例えば、Microsoft Internet Explorer バージョン 5.0 以降。
- 任意の使用可能なテキストエディタ。

アプリケーションの実行

▶手順 22.10. パス調整を定義するには

- 1 メインダイアログボックスを開きます。
- 2 [ツール] > [開発ツール] > [Web Interface Options] を選択し、ブラウザとビューアを指定します。
- 3 ブラウザ、ビューアー、およびワークファイルのパスを設定します。
- 4 [OK] ボタンをクリックします。

▶手順 22.11. アプリケーションを開始するには

- 1 SYSWEB 用のダイアログ WEB-ONL または SYSWEB3 用のダイアログ WEB-ONL3 を開始します。
- 2 ライブラリ名とサブプログラム名を選択します。
- 3 オプション：パラメータを追加します。
- 4 RPC または DCOM を選択します。
- 5 [Execute] ボタンをクリックします。
- 6 [Result] ボタンまたは [Browse] ボタンをクリックして結果を表示します。

サポートされているコンテンツタイプ

次のコンテンツタイプがテストユーティリティでサポートされています。

コンテンツタイプ	拡張子
"application/rtf"	"rtf"
"application/powerpoint"	"ppt"
"application/msword"	"doc"
"application/excel"	"xls"
"text/html"	"htm"
"text/plain"	"txt"

コンテンツタイプ	拡張子
"text/xml"	"xml"
"text/richtext"	"rtf"

上記以外のコンテンツタイプが必要な場合は、サブルーチン HTML2CONTENT-TYPE (SYSWEB/W3CO2EXT または SYSWEB3/W3CO2EXT) を変更し、ニーズに合わせて変換テーブルを拡張します。

入力／出力フィールド

フィールド	説明
Subprogram : Library Name	ライブラリおよび目的のサブプログラムの名前を指定できます。指定可能なライブラリおよびサブプログラムは、自動的にライブラリワークスペースから取得されて選択ボックスにリストされます。
Interface	通信形式として DCOM または RPC を選択できます。DCOM の場合は、まずクラスを登録する必要があります。 デフォルト：RPC
Parameters: Name Value Server	ここでは、必要な名前／値ペアをサブプログラムから入力できます。これらをパラメータリストに移動するには、[Add] ボタンをクリックします。エントリを変更するには、[Modify] ボタンをクリックします。&、=、% を置き換える必要はありません。この処理は WEB-ONL プログラムによって実行されます。サーバーパラメータを使用する場合は、パラメータをパラメータリストに追加する前に、[Server] トグルボタンをオンにします。 パラメータリストにすべての名前／値ペアが表示されます。&、=、% は置き換えられます。ペアを削除するには、項目を選択し、[Delete] ボタンをクリックします。選択したすべての項目が [Name] および [Value] フィールドに挿入されます。ペアを変更するには、項目を選択し、[Name] および [Value] フィールドで変更し、[Modify] ボタンをクリックします。 Server : 名前／値ペアがサーバー変数の場合は、このトグルボタンをオンにする必要があります。すべてのステータスは、再度変更するまで継続されます。
HTTP Method	このドロップダウンリストでは、使用する HTTP 要求／送信メソッドを選択できます。 ■ HEAD GET 要求と似ていますが、応答本文がありません。これは、内容全体を取得せずに、応答ヘッダーに書かれたメタ情報を取得できるので便利です。 ■ GET 指定したリソースの表現を要求します。 ■ POST 特定したリソースからデータを送信します。データは要求の本文に組み込まれます。 このメソッドを使用して、XML ファイルやバイナリデータ（例：グラフィック）などの異なるコンテンツタイプのデータを送信できます。

フィールド	説明
	<p>このメソッドを指定すると、画面上で [Browse] ボタンと [Binary] チェックボックスが追加で使用可能になります。</p> <p>[Browse...] ボタンはファイルを選択する場合に使用し、[Binary] チェックボックスはバイナリデータを送信する場合にオンにします。</p> <p>入力ファイルを指定せずにこのメソッドを指定した場合は、デフォルトの MIME タイプ "application/x-www-form-urlencoded" が設定されます。入力ファイルを使用した場合は、その入力ファイルのコンテンツタイプが使用されます。例えば、XML ファイルの場合は、コンテンツタイプは自動的に "text/xml" に設定されます。別の MIME タイプを入力フィールドに手動で指定できます。</p> <p>注意: 手動で設定した MIME タイプは、デフォルトの MIME タイプよりも常に優先されます。</p> <p>■ PUT 指定したリソースの表現をアップロードします。</p> <p>このメソッドを使用して、XML ファイルやバイナリデータ（例：グラフィック）などの異なるコンテンツタイプのデータを送信できます。</p> <p>このメソッドを指定すると、画面上で [Browse...] ボタンと [Binary] チェックボックスが追加で使用可能になります。</p> <p>[Browse...] ボタンはファイルを選択する場合に使用し、[Binary] チェックボックスはバイナリデータを送信する場合にオンにします。</p>

ボタン



要求したサブプログラムの出力を受け取る処理を実行します。WEB-ONL または WEB-ONL3 ダイアログウィンドウの下部にあるステータスバーに処理のステータスが表示されます。



エディタを開きます。プログラムを実行していなくて、サブプログラムのライブラリまたは名前を変更していなければ、無効になっています。 **[Options]** ダイアログボックスでエディタを選択できます。



[Options] ダイアログボックスで選択したブラウザを開きます。プログラムを実行していない場合は、無効になっています。

23 Natural Web オンラインドキュメント SYSWEB

■ 一般的な情報	136
■ 基本モジュール	136
■ テンプレート/XSLT 処理	138
■ HTML エクステンション	138
■ ユーティリティ	140
■ デモアプリケーション	141

このセクションでは、以下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* および 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サンプルプログラムが含まれています。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`



Note: 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 サーバーが開始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、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。未指定の場合、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で、接頭辞は W3 です。

Natural サブプログラムと HTTP サーバーエクステンションとの通信を可能にします。Natural Web インターフェイスの他のすべてのプログラムはこれらのプログラムを使用します。

転送データの量とフォーマットを定義するため、変換テーブルを変更するため、およびエラーページを変更するために、管理用の変更を加えることができます。

 **Note:** 使用可能な新しいプログラムは、下表に *italics* 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	最大バイト数、および出力エリア内の現在書き込まれているバイト数または空き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	HTML 文字列を出力ページに書き込み、特殊文字を HTML 対応表現に変換します。
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、ドキュメントを HTTP サーバーに戻すための準備をします。
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	同じ名前の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 (W3SPACE)	改行を出力ページに書き込みます。 W3TEXTDYNAMIC " " に置き換え可能です。
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-TO-HTML W3-ASCII-HTML-TABLE	ASCII を HTML の特定のエンコードに変換します。
W3TEXT-TO-XML W3-ASCII-XML-TABLE	ASCII を XML の特定のエンコードに変換します。
W3TEXT-TO-URL W3-ASCII-URL-TABLE	ASCII を URL の特定のエンコードに変換します。

テンプレート/XSLT 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、新しい文字列で置き換えます。
W3LOAD-RESOURCE	結果ドキュメントとしてファイルを特定の Natural ライブラリの リソースディレクトリ からロードします。
W3READ-RESOURCE	ファイルを特定の Natural ライブラリの リソースディレクトリ からダイナミック変数に読み込みます。
W3WRITE-RESOURCE	ダイナミック変数を特定の Natural ライブラリの リソースディレクトリ のファイルに書き込みます。
W3CHECK-RESOURCE	リソース の存在を確認します。存在しない場合は、新しいリソースを作成できます。
W3DELETE-RESOURCE	リソースディレクトリから リソース を削除します。
W3APPLY-XSLT-XML	出力ページの変換に スタイルシート を適用します。
W3APPLY-XSLT-RESOURCE	出力ページにリソースとして保存される スタイルシート を適用します。
W3APPLY-XSLT-XML-TO-DYNAMIC	ダイナミック変数への変換に スタイルシート を適用します。
W3READ-OUTPUT	作成済み の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（ソースコードとともに提供される）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 のすべての構文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ない場合があります。拡張機能が必要な場合は、ソースコードで提供されているプログラムを自由に拡張してかまいません。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（場合によっては後続の追加テキストがあります）。	
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 ドキュメントを開始および終了します。	<HTML>...</HTML>
H3-OPEN-LIST H3-LIST-ITEM H3-CLOSE-LIST	順序付けされたリスト、順序付けされていないリスト、メニューリスト、またはディレクトリリストを生成します。 <DIR>......</DIR> <MENU>......</MENU>
H3-PARAGRAPH	追加テキストが入力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>

プログラム	説明	HTML タグ
H3-TEXT-T0-HTML	Natural 文字列の内容を HTML に変換します。 W3TEXT-T0-HTML に置き換え可能です。	
H3-TEXT-T0-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-T0-URL に置き換え可能です。	
H3-TIME_DATE H3-TIME-DATE	時刻/日付文字列を生成します。	生成内容：2005年1月17日 (月) 15:35:18 GMT

ユーティリティ

Web インターフェイスプラグイン

プラグイン	説明
プログラム生成	Natural のドキュメントを参照してください。
クラスの生成	Natural のドキュメントを参照してください。
オンラインテストユーティリティ WEB-ONL	Natural のドキュメントを参照してください。

オンライン

ダイアログ	説明
WEB-ONL	Natural のドキュメントを参照してください。

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。
NAT-RES	Natural 以外のファイル/リソースを表示します (プラットフォーム共有リソースのみが有効です)。

デモアプリケーション

付属のデモアプリケーションは、選択機能を備えたシンプルなファイルメンテナンスを示します。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。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。

デモアプリケーションの 3 つの実装が提供されます。

1. テンプレート、xml および xslt、名前の接頭辞 D6* の使用。
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 D6INDEX 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/d6index`

2. JavaScript、名前の接頭辞 D4* の使用。
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`

3. 標準 HTML 3.2、名前の接頭辞 D3* の使用。
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ピクチャは、Natural Web インターフェイスとともに提供されます。これらは、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されます。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 を、そのリモートディレクトリ名に設定します。

D4* の例のための JavaScript ファイルは、Natural Web インターフェイスとともに提供されます。これは、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されます。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 を、そのリモートディレクトリ名に設定します。

24 出力エリアのクリア

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

説明

出力エリアにすで書き込まれた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
```


25 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、コンテンツがどのように表示されるかを把握するために、ブラウザプログラムによって使用されます。

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

旧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ので、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
```


26 出力エリアのサイズのカウンタ

サブルーチン名	実行ファイルの例	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出力エリアに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
```


27 エラーページの生成

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

説明

Natural ランタイムで生成されたエラーは、画面に出力されないように処理される必要があります。そのため、NaturalWeb 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 も追加する必要があります。

エラーが発生した場合は、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 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アクティブにするには、W3ERROR-TEMPLATE をアンカタログし、サブルーチンの名前を W3ERROR-TEMPLATE-XML から W3ERROR-TEMPLATE に変更し、サブルーチンを格納します。

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
```

エラーページの生成

```
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
```


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

W3HTML および W3HTMLLINE により、指定した文字列の末尾の空白を削除できます。

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。

旧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ので、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
```

29 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先頭に書き込みます。このテキスト行設定で、COOKIES、EXPIRE-DATES、またはその他の HTTP 互換ドキュメント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、出力の物理的改行を作成できます。

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

W3HTTP により、指定した文字列の末尾の空白を削除できます。

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。

旧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ので、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
```

30 内部値に関する情報

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

説明

このプログラムにより、Web インターフェイスの一部の内部値を設定および読み取りできます。

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、適切なクローズ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 タグ内部のみでしか使用できないタグを生成できます。FORM が開いているかどうかを確認するフラグを呼び出せます。フラグは H3-OPEN-FORM および H3-CLOSE-FORM によって変更されます。

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
```

31 ドキュメントの終了および初期化

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

説明

各プログラムでは、専用の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使用することを考慮し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。初期化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。

W3* 呼び出しが W3END の後に実行された場合、作成された出力は 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
```


32 すべての環境変数のリスト

サブルーチン名	実行ファイルの例	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変数をすべてリストします。

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
```

33 ドキュメントの場所の設定

サブルーチン名	実行ファイルの例	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して、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せます。

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

旧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ので、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
```


34 環境変数の読み取り

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

説明

最初に検出される 1 件の変数を読み取ります。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、同じ変数の複数の検出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
```

35 環境変数グループの読み取り

サブルーチン名	実行ファイルの例	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、同じ名前の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
```


36 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
```

環境テキストエリア変数の読み取り

```
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
```

37 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。

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

W3TEXT および W3TEXTLINE により、指定した文字列の末尾の空白を削除できます。

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。

旧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ので、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
```

38 出力エリアへの改行の書き込み

サブルーチン名

W3NEWLINE

説明

1つの改行 (##HTTP_NEWLINE) を出力エリアに追加します。

このサブルーチンは、次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 を使用します。

パラメータ

*/ NONE

呼び出し方法

PERFORM W3NEWLINE

39 テキストから HTML

サブプログラム名	実行ファイルの例	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 が、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ます。このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


40 テキストから XML

サブプログラム名	実行ファイルの例	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 が、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ます。このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


41 テキストから URL

サブプログラム名	実行ファイルの例	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 が、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ます。
このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


42 リターンドキュメント内での置換

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3REPLACE	E3TEMPL	E3TEMPL	E3TEMPL.HTML

説明

すでに作成さ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新しい文字列にすべて置換します。テンプレート処理の場合は、W3LOAD-RESOURCE と併用します。次の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所定の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"
```

43 リソースのロード

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3LOAD-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

ファイルを所定のライブラリのリソースディレクトリから出力ページにロードします。

ライブラリが指定されていない場合は、現在のライブラリを使用します。

テンプレート処理の場合は、[W3REPLACE](#) と併用します。

パラメータ

```
1 W3LIBRARY      (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3RESOURCE-FILE (A)  DYNAMIC BY VALUE /* i /m : File Name
1 W3RESPONSE     (I4) OPTIONAL           /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。

W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3LOAD-RESOURCE W3LIBRARY W3RESOURCE-FILE
PERFORM W3LOAD-RESOURCE " " "templ.html"
```

制限事項

リソースのロードは、Natural 以外のファイル/リソースをサポートしている Natural for Windows および UNIX でのみ使用可能です。内部的に、ワークファイル 12 を使用します。

44 リソースの読み取り

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3READ-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

リソースファイルを所定のライブラリから動的変数に読み取ります。ライブラリが指定されていない場合は、現在のライブラリを使用します。テンプレート処理の場合は、[W3APPLY-XSLT](#) と併用します。

パラメータ

```
1 W3LIBRARY (A8) BY VALUE optional /* i /o : Natural Library ID
1 W3FILE (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3DYN (A) DYNAMIC /* o/m : Read Resource
1 W3RESPONSE (I4) OPTIONAL /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。

W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3READ-RESOURCE W3LIBRARY W3FILE W3DYN
PERFORM W3READ-RESOURCE W3LIBRARY W3FILE W3DYN W3RESPONSE
```

制限事項

リソースのロードは、Natural 以外のファイル/リソースをサポートしている Natural for Windows および UNIX でのみ使用可能です。内部的に、ワークファイル 12 を使用します。

45 リソースの書き込み

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3WRITE-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

リソースファイルを所定のライブラリに書き込み、所定のダイナミック変数を組み込みます。ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3LIBRARY (A8) BY VALUE OPTIONAL /* i /o : Natural Library
1 W3FILE (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3DYN (A) DYNAMIC BY VALUE /* i /m : Read Resource
1 W3RESPONSE (I4) OPTIONAL /* o/o : Response ID
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。

W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3WRITE-RESOURCE W3LIBRARY W3FILE W3DYN
PERFORM W3WRITE-RESOURCE W3LIBRARY "Test.txt" W3DYN
PERFORM W3WRITE-RESOURCE W3LIBRARY W3FILE W3DYN W3RESPONSE
```

制限事項

リソースのロードは、Natural 以外のファイル/リソースをサポートしている Natural for Windows および UNIX でのみ使用可能です。内部的に、ワークファイル 12 を使用します。

46 リソースの確認

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3CHECK-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

所定のライブラリにリソースファイルの読み取りが存在するかどうかを確認します。

リソースが存在していない場合は、作成フラグを使用して新しい空のリソースを作成できます。

ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3LIBRARY (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3FILE (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3CREATE (L) BY VALUE /* i /m : If true, create new file
1 W3PATH (A) DYNAMIC /* o/m : Resource path
1 W3RESPONSE (I4) OPTIONAL /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。

W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3CHECK-RESOURCE W3LIBRARY W3FILE W3CREATE W3PATH  
PERFORM W3CHECK-RESOURCE " " 'test.html" TRUE W3PATH  
PERFORM W3CHECK-RESOURCE 1x 'test.html" TRUE W3PATH  
PERFORM W3CHECK-RESOURCE W3LIBRARY W3FILE W3CREATE W3PATH W3RESOURCE
```

制限事項

リソースの確認は、Natural 以外のファイル／リソースをサポートしている Natural for Windows および UNIX でのみ使用可能です。

47 リソースの削除

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3DELETE-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

リソースファイルを所定のライブラリから削除します。ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3LIBRARY (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3FILE (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3RESPONSE (I4) OPTIONAL /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。

W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3DELETE-RESOURCE W3LIBRARY W3FILE
PERFORM W3DELETE-RESOURCE " " 'test.html'
PERFORM W3DELETE-RESOURCE 1x 'test.html'
PERFORM W3DELETE-RESOURCE W3LIBRARY W3FILE W3RESPONSE
```

制限事項

リソースの確認は、Natural 以外のファイル/リソースをサポートしている Natural for Windows および UNIX でのみ使用可能です。

48 リターンドキュメントへの XSLT 処理の適用

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3APPLY-XSLT-XML	E3XSLT	E3XSLT

説明

所定の XML を使用して XSLT プロセッサを呼び出し、生成されたデータで出力ページを置き換えます。

パラメータ

```
1 W3XSL      (A) DYNAMIC BY VALUE /* i /m : xsl template
1 W3XML      (A) DYNAMIC BY VALUE /* i /m : xml data
1 W3RESPONSE (I4) OPTIONAL          /* o/o : result
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。

W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3APPLY-XSLT-XML W3XSL W3XML
PERFORM W3APPLY-XSLT-XML W3XSL W3XML W3RESPONSE
```

制限事項

XSLT 処理の適用は、ユーザー出口 USR 6001P を含む Natural for Windows および UNIX でのみ使用可能です。

49 リソースからリターンドキュメントへの XSLT 処理の適用

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3APPLY-XSLT-RESOURCE	E3XSLT2	E3XSLT2	E3XSLT2.XML E3XSLT2.XSL

説明

所定の XSL リソースを使用して XSLT プロセッサを呼び出し、生成されたデータで出力ページを置き換えます。

パラメータ

```
1 W3LIBRARY      (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3RESOURCE-FILE (A)  DYNAMIC BY VALUE /* i /m : File Name
1 W3RESPONSE     (I4) OPTIONAL          /* o/o : result
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。

W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3APPLY-XSLT-RESOURCE W3XSL W3XML
```

制限事項

XSLT 処理の適用は、ユーザー出口 USR6001P を含む Natural for Windows および UNIX のみ使用可能です。

50 リソースディレクトリからのスタイルシートのロードおよびリターンドキュメントへの XSLT 処理の適用

サブルーチン名	実行ファイルの例	表示可能ファイルの例	使用されるリソース
W3APPLY-XSLT-XML-TO-DYNAMIC	E3XSLT3	E3XSLT3	E3XSLT3.HTML E3XSLT3.XSL E3XSLT3.XML

説明

XSLT プロセッサを呼び出し、所定の Natural ライブラリのリソースディレクトリからロードされたスタイルシートで出力ページを変換します。ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3XSLT      (A) DYNAMIC BY VALUE /* i /m : xslt
1 W3XML       (A) DYNAMIC BY VALUE /* i /m : xml
1 W3RESULT    (A) DYNAMIC           /* o/m : result
1 W3RESPONSE (I4) OPTIONAL         /* o/o : result
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

リソースディレクトリからのスタイルシートのロードおよびリターンドキュメントへの XSLT 処理の適用

呼び出し方法

```
PERFORM W3APPLY-XSLT-XML-TO-DYNAMIC W3XSLT W3XML W3RESULT  
PERFORM W3APPLY-XSLT-XML-TO-DYNAMIC W3XSLT W3XML W3RESULT W3RESPONSE
```

制限事項

スタイルシートのロードおよび XSLT 処理の適用は、Natural 以外のファイル/リソースをサポートしている Natural for Windows および UNIX にユーザー出口 USR 6001P が含まれる場合のみ使用可能です。内部的に、ワークファイル 12 を使用します。

51 出力ページの読み取り

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3READ-OUTPUT	E3RESOUR	E3RESOUR

説明

作成済みの出力ページをダイナミック変数に読み取ります。

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
```

52 アンカー

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

説明

ハイパーリンクを作成します。

旧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ので、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.
```

アンカー

```
1 H3STRING      (A) DYNAMIC /* i /MH: String to be displayed
*                /*                as anchor text.
```

呼び出し方法

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

53 ボタン

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```

ボタン

```
1 H3VALUE (A) DYNAMIC /* i /M : Value of the input variable
1 H3URL   (A) DYNAMIC /* i /M : URL of the picture to be used
```

呼び出し方法

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

54 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。グループはテーブル内にフォーマットできます。

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
```

チェックボックスグループ

```
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
```

55 コメント行

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```


56 レベル n のヘッダー

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```

57 画像

サブプログラム名	ソース名	実行ファイルの例	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。そのため、すべてのピクチャは HTTP サーバーを使用して保存する必要があります。

旧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ので、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
```

画像

```
*          /*      'B' Bottom  
*          /*      'M' Middle
```

呼び出し方法

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

58 入力

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```

入力

```
1 H3SIZE (N4) /* i /M : Size of the input box  
1 H3MAX (N4) /* i /M : Maximum length of the input text
```

呼び出し方法

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

59 改行

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

説明

改行を強制設定します（場合によっては後続の追加 HTML テキストがあります）。

旧バージョンからの変更点

パラメータにはオプションとマークされているので、どちらのバージョンも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ので、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
```


60 フォーム

サブプログラム名	ソース名	実行ファイルの例	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。事前に 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 を実行しないと、開いているフォームがすべて H3-CLOSE-HTML によって閉じられます。

旧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ので、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
```

61 HTML ドキュメント

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```

62 箇条書き

サブプログラム名	ソース名	実行ファイルの例	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。

- 順序付けされていないリスト
- 順序付けされたリスト
- 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ます。一度に複数のレベルを閉じることもできます。

旧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ので、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
```


63 段落

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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"> または <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
```

段落

```
*           /*           'C' Center
1 H3HTML (A) DYNAMIC /* i /mh: HTML text after the paragraph
```

呼び出し方法

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

64 ラジオボタングループ

サブプログラム名	ソース名	実行ファイルの例	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。グループはテーブル内に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
```

ラジオボタングループ

```
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
```

65 横罫線

サブプログラム名	ソース名	実行ファイルの例	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
```


66 スクロールリスト

サブプログラム名	ソース名	実行ファイルの例	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。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。旧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ので、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
```


67 表

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

説明

所定の配列のシンプル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。行は交互に色付けできます。

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
```


68 汎用タグ

サブプログラム名	ソース名	実行ファイルの例	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は、コードを記述できるフレームワークを作成します。

旧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ので、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
```

汎用タグ

```
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
```

69 テキストエリア

サブプログラム名	ソース名	実行ファイルの例	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。旧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ので、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
```

テキストエリア

```
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
```


70 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 デコード形式の構文に変換します。特殊文字が含まれている場合に便利です。H3-TEXT-TO-URL ではなく [W3-TEXT-TO-URL](#) を使用します。H3-ASCII-URL-TABLE ではなく [W3-ASCII-URL-TABLE](#) を使用します。

サブプログラム W3-ASCII-URL-TABLE が、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ます。このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


71 時刻／日付文字列

サブプログラム名	ソース名	実行ファイルの例	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: ..."文字列を作成するか、現在の時刻／日付を使用してオフセットを含むHTTP互換の時刻／日付文字列を作成します（GMTまたはGMTへのオフセットがNaturalに認識されないため）。

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
```

72 すべての Natural ライブラリのリスト

サブプログラム名	実行ファイルの例	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足し、その日付を有効期限日に設定します。
START=	表示されるオブジェクトセットのワイルドカードを選択します。

呼び出し方法

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


73 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する場合に備えて、Natural Web インターフェイスサブプログラムをオンラインで実行しておくのが便利です。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 文字より長い行は折り返されます。環境変数を設定できます。変数をサーバー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

Natural の *next* プロンプトからプログラム WEB-ONL を実行します。

74 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、次のバージョンのいずれかで削除されます。Windows では **Web インターフェイスプラグイン** 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

Natural の *next* プロンプトからプログラム WEB-WIZ を実行します。

75 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足し、その日付を有効期限日に設定します。
START=	表示されるオブジェクトセットのワイルドカードを選択します。

呼び出し方法

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


76 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
```


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

サブプログラム名	実行ファイルの例	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る 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
```


78 Natural Web インターフェイスの現在の設定のリスト

サブプログラム名	実行ファイルの例	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
```


79 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足し、その日付を有効期限日に設定します。
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
```


80 オンラインドキュメント

サブプログラム名	実行ファイルの例
NAT-DOCU	NAT-DOCU

説明

Natural ソースオブジェクトとして保存される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足し、その日付を有効期限日に設定します。

呼び出し方法

```
nat-docu
```


81 Natural 以外のファイル／リソースのリスト

サブプログラム名	実行ファイルの例	表示可能ファイルの例
NAT-RES	NAT-RES?source=e3xslt2.xml&lib=sysweb	NAT-RES

制限事項

共有リソースを備えたプラットフォームでのみ使用可能です。

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=	リソース名 (拡張子付き)

呼び出し方法

```
nat-res?source=e3xslt2.xml&lib=sysweb
```


82

Natural Web オンラインドキュメント SYSWEB3

■ 全般的な情報	264
■ 基本モジュール	264
■ テンプレート/XSLT 処理	266
■ HTML エクステンション	267
■ ユーティリティ	268
■ デモアプリケーション	269

このセクションでは、以下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* および 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サンプルプログラムが含まれています。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 SYSWEB3 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/yourcgi3/sysweb3/nat-docu`



Note: 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 サーバーが開始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、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。未指定の場合、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で、接頭辞は W3 です。

Natural サブプログラムと HTTP サーバーエクステンションとの通信を可能にします。Natural Web インターフェイスの他のすべてのプログラムはこれらのプログラムを使用します。

転送データの量とフォーマットを定義するため、変換テーブルを変更するため、およびエラーページを変更するために、管理用の変更を加えることができます。

プログラム	説明
W3BINARY	バイナリデータを内部バイナリバッファに書き込みます。
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	最大バイト数、および出力エリア内の現在書き込まれているバイト数または空き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 W3HTMLDYNAMIC W3HTMLLINEDYNAMIC W3HTMLARRAY	HTML 文字列を出力ページに書き込み、特殊文字を HTML 対応表現に変換します。
W3HTTP W3HTTPDYNAMIC W3HTTP-HEADER W3HTTPARRAY	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	SYSWEB3 を初期化し、ドキュメントを HTTP サーバーに返すための準備をします。
W3LIMIT	戻りページの最大サイズを設定します。上記の W3COUNTER を参照してください。
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。
W3MIME-DATA	指定された MIME タイプに必要なデータタイプ (バイナリまたは英数字) を評価します。
W3MIME-TYPE	指定された MIME タイプのファイル拡張子を評価します。 指定されたファイル拡張子の MIME タイプを評価します。
W3READ-ENVIRONMENT W3READ-ENVIRONMENT-ARRAY W3READ-ENVIRONMENT-TO-DYNAMIC	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 W3READ-ENVIRONMENT-TEXTAREA-DYN	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	同じ名前の環境変数をすべて読み取ります。
W3READ-INPUT	HTTP サーバーから渡されたすべてのデータを読み込みます。
W3TEXT	テキスト文字列を出力ページに書き込みます。

プログラム	説明
W3TEXTLINE W3TEXTDYNAMIC W3TEXTLINEDYNAMIC W3TEXTARRAY	
W3NEWLINE	改行を出力ページに書き込みます。
W3SOURCE-TO-XARRAY	完全な Natural ソースを単一の X-array に読み込みます。
W3TEXT-TO-HTML W3-ASCII-HTML-TABLE	ASCII を HTML の特定のエンコードに変換します。
W3TEXT-TO-XML W3-ASCII-XML-TABLE	ASCII を XML の特定のエンコードに変換します。
W3TEXT-TO-URL W3-ASCII-URL-TABLE	ASCII を URL の特定のエンコードに変換します。

テンプレート / XSLT 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、新しい文字列で置き換えます。
W3READ-OUTPUT	作成済みの出力ページを読み取ります。

プログラム	説明
W3REPLACE	出力ページで特定の文字列を検索し、新しい文字列で置き換えます。
W3LOAD-RESOURCE	結果ドキュメントとしてファイルを特定の Natural ライブラリの リソースディレクトリ からロードします。
W3READ-RESOURCE	ファイルを特定の Natural ライブラリの リソースディレクトリ からダイナミック変数に読み込みます。
W3WRITE-RESOURCE	ダイナミック変数を特定の Natural ライブラリの リソースディレクトリ のファイルに書き込みます。
W3CHECK-RESOURCE	リソース の存在を確認します。存在しない場合は、新しい リソース を作成できます。
W3DELETE-RESOURCE	リソースディレクトリ から リソース を削除します。
W3APPLY-XSLT-XML	出力ページの変換に スタイルシート を適用します。
W3APPLY-XSLT-RESOURCE	出力ページに リソース として保存される スタイルシート を適用します。
W3APPLY-XSLT-XML-TO-DYNAMIC	ダイナミック変数への変換に スタイルシート を適用します。
W3LIST-RESOURCE	特定の Natural ライブラリの リソースファイル をすべてリストします。
W3READ-OUTPUT	作成済みの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（ソースコードとともに提供される）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のすべての構文に対応していません。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ない場合があります。拡張機能が必要な場合は、ソースコードで提供されているプログラムを自由に拡張してかまいません。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（場合によっては後続の追加テキストがあります）。	
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 ドキュメントを開始および終了します。	<HTML>... </HTML>
H3-OPEN-LIST H3-LIST-ITEM H3-CLOSE-LIST	順序付けされたリスト、順序付けされていないリスト、メニューリスト、またはディレクトリリストを生成します。 <DIR>... ... </DIR> <MENU>... ... </MENU>
H3-PARAGRAPH	追加テキストが入力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>

プログラム	説明	HTML タグ
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 に置き換え可能です。	
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 に置き換え可能です。	
H3-TIME_DATE H3-TIME-DATE	時刻／日付文字列を生成します。	生成内容：2005年1月17日 (月) 15:35:18 GMT

ユーティリティ

Web インターフェイスプラグイン

プラグイン	説明
プログラム生成	Natural のドキュメントを参照してください。
クラスの生成	Natural のドキュメントを参照してください。
オンラインテストユーティリティ	Natural のドキュメントを参照してください。

オンライン

ダイアログ	説明
WEB-ONL または WEB-ONL3	Natural のドキュメントを参照してください。

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-DATA	HTTPサーバーから渡されたすべてのヘッダー／データ（バイナリ、英数字、Unicode?）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-DIRR	特定の Natural ライブラリのリソース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。
NAT-RES	Natural 以外のファイル／リソースを表示します（プラットフォーム共有リソースのみが有効です）。

デモアプリケーション

付属のデモアプリケーションは、選択機能を備えたシンプルなファイルメンテナンスを示します。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。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。

デモアプリケーションの実装では、テンプレート、XML および XSLT、名前の接頭辞 D6* を使用します。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 SYSWEB3 からサブプログラム D6INDEX 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/sysweb3/d6index>

使用されるすべてのピクチャは、Natural Web インターフェイスとともに提供されます。これらは、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されます。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 を、そのリモートディレクトリ名に設定します。

83 ドキュメントへのバイナリの書き込み

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3BINARY	E3BINARY	E3BINARY

説明

バイナリをドキュメントに書き込みます。ドキュメントに含めることができるのは、バイナリまたは英数字の戻り値のいずれかです。両方を含めることはできません。

パラメータ

W3BINARY

```
1 H3BINARY (A) DYNAMIC BY VALUE /* i /mH: Output string
```

呼び出し方法

```
PERFORM W3BINARY H3BINARYDYNAMIC
```


84 出力エリアのクリア

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

説明

出力エリアにすで書き込まれた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
```


85 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、コンテンツがどのように表示されるかを把握するために、ブラウザプログラムによって使用されます。

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

旧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ので、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
```


86 出力エリアのサイズのカウント

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3COUNT W3LIMIT	E3COUNT	E3COUNT

説明

すで書き込まれたバイト数を返します。

旧バージョンからの変更点

Natural バージョン 6.2 以降では、出力エリアのサイズがコンパイル時の内部設定の制限を受けなくなりました。したがって、最大サイズは、Natural ダイナミック変数の最大サイズと、RPC または DCOM サーバーに使用されるパラメータ設定によってのみ制限されます。W3MAXPAGE の戻り値は、プログラム W3LIMIT 内で設定できるようになりましたが、内部で生成され保存される最大出力サイズには影響しません。

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
```


87 エラーページの生成

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

説明

Natural ランタイムで生成されたエラーは、画面に出力されないように処理される必要があります。そのため、NaturalWeb 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 も追加する必要があります。

エラーが発生した場合は、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 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アクティブにするには、W3ERROR-TEMPLATE をアンカタログし、サブルーチンの名前を W3ERROR-TEMPLATE-XML から W3ERROR-TEMPLATE に変更し、サブルーチンを格納します。

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
```

エラーページの生成

```
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
```

88 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。

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

W3HTML および W3HTMLLINE により、指定した文字列の末尾の空白を削除できます。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。

旧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ので、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 VALUE /* 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
```

89 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先頭に書き込みます。このテキスト行設定で、COOKIES、EXPIRE-DATES、またはその他の HTTP 互換ドキュメント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、出力の物理的改行を作成できます。

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

W3HTTP により、指定した文字列の末尾の空白を削除できます。

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。

旧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ので、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
```

90 内部値に関する情報

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

説明

このプログラムにより、Web インターフェイスの一部の内部値を設定および読み取りできます。

読み取り (アクション 'V')

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、適切なクローズ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 タグ内部のみでしか使用できないタグを生成できません。FORM が開いているかどうかを確認するフラグを呼び出せます。フラグは H3-OPEN-FORM および H3-CLOSE-FORM によって変更されます。

旧バージョンからの変更点

LDA W3PINFO が変更されました。また、変数 LOG-TIME、VERSION、および WEBAPI が (A) DYNAMIC に変更されました。

パラメータ

LOCAL USING W3PINFO

PDA W3PINFO

```
1 ##W3INFO
2 ACTION (A1) /* i /m : Action to be called
2 LOG-TIME (A) DYNAMIC /* o/m : Log time set by the Natural Web Interface
2 VERSION (A) DYNAMIC /* o/m : Version set by the HTTP Server Extension
2 WEBAPI (A) DYNAMIC /* 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 (A1/1:10) /* io/m : Saves the type of LIST
2 LIST_MAX (N2) /* io/m : Current number of nested LISTS
```

呼び出し方法

```
PERFORM W3INFO ##W3INFO
```


91 ドキュメントの終了および初期化

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

説明

各プログラムでは、専用の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使用することを考慮して初期化するために、PDA W3PARM 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。初期化プログラムで定義された PDA W3PARM は W3END に渡す必要があります。W3* 呼び出しが W3END の後に実行された場合、作成された出力は HTTP サーバーに転送されません。

SYSWEB3 では、W3PARM が変更されて、HTTP サーバーへのインターフェイスを改善されました。この新しいインターフェイスでは、バイナリまたは英数字データの転送、受信データへの他の MIME タイプ ("application/x-www-form-urlencoded" など) の使用、HTTP PUT 要求の使用が可能です。

互換性を確保するため、新機能の HTTP PUT とデータ入力用の他の MIME タイプは、W3INIT を次の追加パラメータで呼び出した場合のみ使用できます。

第 2 パラメータ := TRUE - 他の MIME タイプを有効にしてから、受信データ用に "application/x-www-form-urlencoded" を有効にします。

第 3 パラメータ := TRUE - HTTP PUT 要求を有効にします。

旧バージョンからの変更点

新しいオプションのパラメータが追加されました。

パラメータ

1. W3INIT

```
###RPC /* i /m : Parameter of Subprogram
TRUE /* io/ : TRUE to activate mime-types
TRUE /* io/ : TRUE to activate HTTP PUT
```

2. W3END

```
###RPC /* o/m : Parameter of Subprogram
```

3. W3PARAM

```
1 ###RPC /* use only with SYSWEB3
2 LOG-TIME (A30) /* i /m : Timestamp
2 VERSION (A) DYNAMIC /* i /m : Interface version
2 HTTP_HEADER (A) DYNAMIC /* io/m : HTTP header
2 C_HTTP_HEADER (I4) /* io/m : Bytes sent
2 HTTP_BODY (A) DYNAMIC /* io/m : HTTP body alphanumeric
2 HTTP_BINARY (B) DYNAMIC /* io/m : HTTP body binary
2 C_HTTP_DATA (I4) /* io/m : Bytes sent
2 MIME-TYPE (A) DYNAMIC /* io/m : Mime-type sent
2 ERROR-NR (I4) /* io/m : Generation result
```

呼び出し方法

```
PERFORM W3INIT ###RPC
PERFORM W3INIT ###RPC TRUE /* activate mime-types ne
'application/x-www-form-urlencoded'
PERFORM W3INIT ###RPC TRUE TRUE /* activate mime-types and HTTP PUT
PERFORM W3END ###RPC
```

92 すべての環境変数のリスト

サブルーチン名	実行ファイルの例	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変数をすべてリストします。

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
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
```

93

MIME タイプと転送／データタイプの評価

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3MIME-DATA MIMEDATA	NAT-RES?source=e3xslt2.xml&lib=sysweb3 NAT-RES?source=composing_natural_logo.jpg&lib=sysweb3	NAT-RES

説明

各プログラムでは、専用の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使用することを考慮し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。初期化プログラムで定義された W3PARMPDA は W3END に渡す必要があります。W3* 呼び出しが W3END の後に実行された場合、作成された出力は 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 W3PARM /* 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
```

94 MIME タイプとファイル拡張子の評価

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3MIME-TYPE MIMETYPE	NAT-RES?source=e3xslt2.xml&lib=sysweb3 NAT-RES?source=composing_natural_logo.jpg&lib=sysweb3	NAT-RES

説明

Natural Web インターフェイスには、MIME タイプ／ファイル拡張子設定用の独自のテーブルがあります。このテーブルは、Natural テキストメンバ MIMETYPE に保存されます。このテキストメンバには、以下の MIME タイプ名とファイル拡張子のペアが含まれています。

```
text/html      html htm
text/plain     txt
text/xml       xml
text/richtext  rtf
```

W3MIME-TYPE は、内部テーブルを読み込み、以下の変換を行います。

- 指定されたファイル拡張子の MIME タイプを評価します。
- 指定された MIME タイプのファイル拡張子を評価します。

この機能は、W3LOAD-RESOURCE サブプログラムで内部的に使用されます。

パラメータ

1. W3MIME-TYPE

```
W3MIME-TYPE      (A) DYNAMIC /* i /m : requested mime-type
W3FILE-EXTENTION (A) DYNAMIC /* o/m : found file extension
```

2. W3MIME-TYPE

```
W3MIME-TYPE      (A) DYNAMIC /* o/m : found mime-type
W3FILE-EXTENTION (A) DYNAMIC /* i /m : requested file extension
```

呼び出し方法

```
PERFORM W3MIME-TYPE W3FILE-EXTENTION
```


95 ドキュメントの場所の設定

サブルーチン名	実行ファイルの例	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して、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せます。

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

旧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ので、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
```


96 環境変数の読み取り

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

説明

最初に検出される 1 件の変数を読み取ります。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、同じ変数の複数の検出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
```

97 環境変数グループの読み取り

サブルーチン名	実行ファイルの例	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、同じ名前の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
```


98

環境テキストエリア変数の読み取り

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3READ-ENVIRONMENT-TEXTAREA W3READ-ENVIRONMENT-TEXTAREA-DYN	E3ENVTX E3ENVTXD	E3ENVTX E3ENVTX1 E3ENVTXD E3ENVTX2

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. W3READ-ENVIRONMENT-TEXTAREA

```
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
```

2. W3READ-ENVIRONMENT-TEXTAREA-DYN

```
1 W3START          (I4)          /* io/m : Offset to be started at
*                  /*          out: 0 no occurrences
*                  /*          out: >0 more occurrences
1 W3NAME-DYN       (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 W3ARRAYVALUE-DYN (A/1:V) DYNAMIC /* o/m : Value of variables
1 W3ARRAYVALUELENGTH (I4/1:V)    /* o/m : Length of variables
1 W3ARRAYMAXIMUM   (I4)          /* o/m : Total number of variables
```

呼び出し方法

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

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


99 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。

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

W3TEXT および W3TEXTLINE により、指定した文字列の末尾の空白を削除できます。

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。

旧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ので、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
```

100

出力エリアへの改行の書き込み

サブルーチン名

W3NEWLINE

説明

1つの改行 (##HTTP_NEWLINE) を出力エリアに追加します。このサブルーチンは、次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 を使用します。

パラメータ

*/ NONE

呼び出し方法

PERFORM W3NEWLINE

101 ダイナミック X-array への Natural ソースの読み込み

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3SOURCE-TO-XARRAY	E3NAT2XA	E3NAT2XA

説明

指定されたライブラリから Natural ソースを単一の X-array に読み込みます。

パラメータ

```
1 W3LIBRARY      (A8) DYNAMIC BY VALUE /* i /o : Natural Library ID
1 W3SOURCE       (A8) DYNAMIC BY VALUE /* i /m : Source Name
1 W3OUTSRC-XA    (A/1:*) DYNAMIC      /* o/o : Response Array
1 W3RESPONSE     (I4)                  /* o/o : Response
```

呼び出し方法

```
PERFORM W3SOURCE-TO-XARRAY W3LIBRARY W3SOURCE W3OUTSRC-XA(*) W3RESPONSE
```


102 テキストから HTML

サブプログラム名	実行ファイルの例	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 が、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ます。このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


103 テキストから XML

サブプログラム名	実行ファイルの例	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 が、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ます。このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


104 テキストから URL

サブプログラム名	実行ファイルの例	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 が、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ます。このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


105 リターンドキュメント内での置換

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3REPLACE	E3TEMPL	E3TEMPL	E3TEMPL.HTML

説明

すでに作成さ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新しい文字列にすべて置換します。テンプレート処理の場合は、W3LOAD-RESOURCE と併用します。

次の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所定の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"
```


106 リソースのロード

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3LOAD-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

ファイルを所定のライブラリのリソースディレクトリから出力ページにロードします。
ライブラリが指定されていない場合は、現在のライブラリを使用します。
テンプレート処理の場合は、[W3REPLACE](#) と併用します。

データは、"MIMEDATA" および "MIMETYPE" の設定に応じてバイナリまたは英数字としてロードされます。

ファイル拡張子の MIME タイプを評価するには、[W3MIME-TYPE](#) を使用します。

パラメータ

```
1 W3LIBRARY      (A8) DYNAMIC BY VALUE OPTIONAL      /* i /o : Natural Library ID
1 W3RESOURCE-FILE (A) DYNAMIC BY VALUE              /* i /m : File Name
1 W3RESPONSE    (I4) OPTIONAL                       /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3LOAD-RESOURCE W3LIBRARY W3RESOURCE-FILE  
PERFORM W3LOAD-RESOURCE " " "templ.html"  
PERFORM W3LOAD-RESOURCE 1X "templ.html"
```

制限事項

リソースのロードは、Natural 以外のファイル／リソースをサポートしている Natural for Windows、および UNIX でのみ使用可能です。内部的に、ワークファイル 12 を使用します。

107 リソースの読み取り

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3READ-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

リソースファイルを所定のライブラリから動的変数に読み取ります。ライブラリが指定されていない場合は、現在のライブラリを使用します。

テンプレート処理の場合は、[W3APPLY-XSLT](#) と併用します。

パラメータ

```
1 W3LIBRARY (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3FILE (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3DYN (A) DYNAMIC /* o/m : Read Resource
1 W3RESPONSE (I4) OPTIONAL /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3READ-RESOURCE W3LIBRARY W3FILE W3DYN
PERFORM W3READ-RESOURCE W3LIBRARY W3FILE W3DYN W3RESPONSE
```

制限事項

リソースのロードは、Natural 以外のファイル/リソースをサポートしている Natural for Windows および UNIX でのみ使用可能です。内部的に、ワークファイル 12 を使用します。

108 リソースの書き込み

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3WRITE-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

リソースファイルを所定のライブラリに書き込み、所定のダイナミック変数を組み込みます。ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3LIBRARY (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3FILE (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3DYN (A) DYNAMIC BY VALUE /* i /m : Read Resource
1 W3RESPONSE (I4)OPTIONAL /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3WRITE-RESOURCE W3LIBRARY W3FILE W3DYN
PERFORM W3WRITE-RESOURCE W3LIBRARY "Test.txt" W3DYN
PERFORM W3WRITE-RESOURCE W3LIBRARY W3FILE W3DYN W3RESPONSE
```

制限事項

リソースのロードは、Natural 以外のファイル/リソースをサポートしている Natural for Windows、および UNIX でのみ使用可能です。内部的に、ワークファイル 12 を使用します。

109 リソースの確認

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3CHECK-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

所定のライブラリにリソースファイルの読み取りが存在するかどうかを確認します。リソースが存在していない場合は、作成フラグを使用して新しい空のリソースを作成できます。ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3LIBRARY      (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3FILE         (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3CREATE       (L) BY VALUE /* i /m : If true, create new file
1 W3PATH         (A) DYNAMIC /* o/m : Resource path
1 W3RESPONSE     (I4) OPTIONAL /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3CHECK-RESOURCE W3LIBRARY W3FILE W3CREATE W3PATH
PERFORM W3CHECK-RESOURCE " " 'test.html' TRUE W3PATH
PERFORM W3CHECK-RESOURCE 1x 'test.html' TRUE W3PATH
PERFORM W3CHECK-RESOURCE W3LIBRARY W3FILE W3CREATE W3PATH W3RESOURCE
```

制限事項

リソースの確認は、Natural 以外のファイル/リソースをサポートしている Natural for Windows、および UNIX でのみ使用可能です。

110 リソースの削除

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3DELETE-RESOURCE	E3RESOUR	E3RESOUR	E3SAVE.HTML

説明

リソースファイルを所定のライブラリから削除します。ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3LIBRARY      (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3FILE         (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3RESPONSE    (I4)OPTIONAL /* o/o : Response
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3DELETE-RESOURCE W3LIBRARY W3FILE
PERFORM W3DELETE-RESOURCE " " 'test.html'
PERFORM W3DELETE-RESOURCE 1x 'test.html'
PERFORM W3DELETE-RESOURCE W3LIBRARY W3FILE W3RESPONSE
```

制限事項

リソースの確認は、Natural 以外のファイル/リソースをサポートしている Natural for Windows、および UNIX でのみ使用可能です。

111 リターンドキュメントへの XSLT 処理の適用

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3APPLY-XSLT-XML	E3XSLT	E3XSLT

説明

所定の XML を使用して XSLT プロセッサを呼び出し、生成されたデータで出力ページを置き換えます。

パラメータ

```
1 W3XSL      (A) DYNAMIC BY VALUE /* i /m : xsl template
1 W3XML      (A) DYNAMIC BY VALUE /* i /m : xml data
1 W3RESPONSE (I4) OPTIONAL      /* o/o : result
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3APPLY-XSLT-XML W3XSL W3XML
PERFORM W3APPLY-XSLT-XML W3XSL W3XML W3RESPONSE
```

制限事項

XSLT 処理の適用は、ユーザー出口 USR 6001P を含むバージョンの Natural for Windows/UNIX でのみ使用可能です。

112 リソースからリターンドキュメントへの XSLT 処理の適用

サブルーチン名	実行ファイルの例	表示可能ファイルの例	リソース
W3APPLY-XSLT-RESOURCE	E3XSLT2	E3XSLT2	E3XSLT2.XML E3XSLT2.XSL

説明

所定の XSL リソースを使用して XSLT プロセッサを呼び出し、生成されたデータで出力ページを置き換えます。

パラメータ

```
1 W3LIBRARY      (A8) BY VALUE OPTIONAL /* i /o : Natural Library ID
1 W3RESOURCE-FILE (A) DYNAMIC BY VALUE /* i /m : File Name
1 W3RESPONSE     (I4) OPTIONAL          /* o/o : result
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

呼び出し方法

```
PERFORM W3APPLY-XSLT-RESOURCE W3XSL W3XML
```

制限事項

XSLT 処理の適用は、ユーザー出口 USR6001P を含むバージョンの Natural for Windows/UNIX でのみ使用可能です。

113 リソースディレクトリからのスタイルシートのロードおよびリターンドキュメントへの XSLT 処理の適用

サブルーチン名	実行ファイルの例	表示可能ファイルの例	使用されるリソース
W3APPLY-XSLT-XML-TO-DYNAMIC	E3XSLT3	E3XSLT3	E3XSLT3.HTML E3XSLT3.XSL E3XSLT3.XML

説明

XSLT プロセッサを呼び出し、所定の Natural ライブラリのリソースディレクトリからロードされたスタイルシートで出力ページを変換します。ライブラリが指定されていない場合は、現在のライブラリを使用します。

パラメータ

```
1 W3XSLT      (A) DYNAMIC BY VALUE /* i /m : xslt
1 W3XML       (A) DYNAMIC BY VALUE /* i /m : xml
1 W3RESULT    (A) DYNAMIC          /* o/m : result
1 W3RESPONSE (I4) OPTIONAL         /* o/o : result
```

W3RESPONSE パラメータを指定しない場合は、エラー発生時に Web インターフェイスエラー処理がトリガされます。W3RESPONSE パラメータを指定した場合は、パラメータに、エラー発生時は Natural メッセージ番号、成功時はゼロが含まれます。

リソースディレクトリからのスタイルシートのロードおよびリターンドキュメントへの XSLT 処理の適用

呼び出し方法

```
PERFORM W3APPLY-XSLT-XML-TO-DYNAMIC W3XSLT W3XML W3RESULT  
PERFORM W3APPLY-XSLT-XML-TO-DYNAMIC W3XSLT W3XML W3RESULT W3RESPONSE
```

制限事項

スタイルシートのロードおよび XSLT 処理の適用は、Natural 以外のファイル/リソースをサポートしているバージョンの Natural for Windows/UNIX にユーザー出口 USR 6001P が含まれる場合のみ使用可能です。内部的に、ワークファイル 12 を使用します。

114 リソースファイルのリスト

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3LIST-RESOURCE	nat-dirr	nat-dirr

説明

特定のNaturalライブラリのリソースファイルをすべてリストします。ワイルドカード選択にはW3PATTERNを使用します。

パラメータ

```
1 W3LIBRARY      (A) DYNAMIC BY VALUE OPTIONAL /* i / : library
1 W3PATTERN      (A) DYNAMIC BY VALUE      /* i /m : selcection pattern
1 W3FILES        (A/1:*) DYNAMIC           /* o/m : array with all file names
1 W3RESPONSE     (I4) OPTIONAL             /* o/ : response code
```

呼び出し方法

```
PERFORM W3LIST-RESOURCE 1x '*' W3FILES(*)
PERFORM W3LIST-RESOURCE 'SYSWEB' '*.HTM' W3FILES(*)
PERFORM W3LIST-RESOURCE 'SYSTEM' '*.BMP' W3FILES(*) W3RESPONSE
```


115 入力ページの読み込み

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3READ-INPUT	E3MULTIP	E3MULTIP NAT-DATA

説明

指定された入力データを HTTP サーバーからダイナミック変数に読み込みます。

W3BODY には、"POST/PUT" で渡された英数字データが含まれています。

W3BINARY には、"POST/PUT" で渡されたバイナリデータが含まれています。

W3HEADER には、渡されたすべての HTTP ヘッダーが含まれています。

W3DATA には、QUERY_STRING のデータと、POST で渡されたデータ（コンテンツタイプが application/x-www-form-urlencoded の場合）が含まれています。

パラメータ

```
1 W3BODY      (A) DYNAMIC          /* o/m : given body
1 W3BINARY    (A) DYNAMIC OPTIONAL /* o/  : given binary
1 W3HEADER    (A) DYNAMIC OPTIONAL /* o/  : given header
1 W3DATA      (A) DYNAMIC OPTIONAL /* o/  : given parameter data
```

呼び出し方法

```
PERFORM W3READ-INPUT W3BODY

PERFORM W3READ-INPUT W3BODY 2X W3DATA

PERFORM W3READ-INPUT W3BODY W3BINARY W3HEADER W3DATA
```


116 出力ページの読み取り

サブルーチン名	実行ファイルの例	表示可能ファイルの例
W3READ-OUTPUT	E3RESOUR	E3RESOUR

説明

作成済みの出力ページをダイナミック変数に読み取ります。

旧バージョンからの変更点

Natural バージョン 6.2 では、オプションのパラメータ W3BINARY が追加されています。

パラメータ

```
1 W3HEADER (A) DYNAMIC /* o/m : written header
1 W3BODY (A) DYNAMIC /* o/m : written body
1 W3BINARY (A) DYNAMIC OPTIONAL /* o/ : written binary
```

呼び出し方法

```
PERFORM W3READ-OUTPUT W3HEADER W3BODY
PERFORM W3READ-OUTPUT W3HEADER W3BODY W3BINARY
```


117 アンカー

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

説明

ハイパーリンクを作成します。

旧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ので、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.
```

アンカー

```
1 H3STRING          (A) DYNAMIC /* i /MH: String to be displayed
*                   /*           as anchor text.
```

呼び出し方法

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

118 ボタン

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```

ボタン

```
1 H3VALUE (A) DYNAMIC /* i /M : Value of the input variable
1 H3URL (A) DYNAMIC /* i /M : URL of the picture to be used
```

呼び出し方法

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


119 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。グループはテーブル内にフォーマットできます。

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
* /* the elements
1 H3ROW (N4) /* i /m : Set number of rows for
* /* tables
```

チェックボックスグループ

```
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
```

120 コメント行

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```


121 レベル n のヘッダー

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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>
```

パラメータ

レベルnのヘッダー

```
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
```

122 画像

サブプログラム名	ソース名	実行ファイルの例	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。そのため、すべてのピクチャは HTTP サーバーを使用して保存する必要があります。

旧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ので、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 of 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
```

画像

```
*          /*      'B' Bottom  
*          /*      'M' Middle
```

呼び出し方法

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


123 入力

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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
```

入力

```
1 H3SIZE (N4) /* i /M : Size of the input box
1 H3MAX (N4) /* i /M : Maximum length of the input text
```

呼び出し方法

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

124 改行

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

説明

改行を強制設定します（場合によっては後続の追加 HTML テキストがあります）。

旧バージョンからの変更点

パラメータにはオプションとマークされているので、どちらのバージョンも同等に使用できます。バージョン H3-LINE-BREAK のみを使用することをお勧めします。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ので、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
```

125 フォーム

サブプログラム名	ソース名	実行ファイルの例	表示可能ファイルの例
H3-OPEN-FORM	H3OFORM	E3FORM	E3FORM
H3-CLOSE-FORM	H3CFORM	E3FORM2 (UTF-8)	E3FORM2

説明

フォームを作成します。事前に 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 を実行しないと、開いているフォームがすべて H3-CLOSE-HTML によって閉じられます。

旧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ので、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
```

126 HTML ドキュメント

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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="#RPG">
```

3. H3-CLOSE-HTML

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

サポートされている属性

```
BACKGROUND="URL", BGCOLOR="#RPG", 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
```


127 箇条書き

サブプログラム名	ソース名	実行ファイルの例	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。

- 順序付けされていないリスト
- 順序付けされたリスト
- 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ます。一度に複数のレベルを閉じることもできます。

旧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ので、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
```


128 段落

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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"> または <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
```

段落

```
*           /*           'C' Center
1 H3HTML    (A) DYNAMIC /* i /mh: HTML text after the paragraph
```

呼び出し方法

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

129 ラジオボタングループ

サブプログラム名	ソース名	実行ファイルの例	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。グループはテーブル内に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 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
```

ラジオボタングループ

```
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
```


130 横罫線

サブプログラム名	ソース名	実行ファイルの例	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
```


131 スクロールリスト

サブプログラム名	ソース名	実行ファイルの例	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。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。

旧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ので、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"> </SELECT> <OPTION> </OPTION>
```

サポートされている属性

```
SIZE="n", MULTIPLE, VALUE="string", SELECTED
```

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
```

スクロールリスト

```
*                               /*      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
```

132 表

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

説明

所定の配列のシンプル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。行は交互に色付けできます。

表示

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

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
```


133 汎用タグ

サブプログラム名	ソース名	実行ファイルの例	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は、コードを記述できるフレームワークを作成します。

旧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ので、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
```

汎用タグ

```
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
```

134 テキストエリア

サブプログラム名	ソース名	実行ファイルの例	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。

旧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ので、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 lines
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
```

テキストエリア

```
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
```

135 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 デコード形式の構文に変換します。特殊文字が含まれている場合に便利です。H3-TEXT-TO-URL ではなく [W3-TEXT-TO-URL](#) を使用します。H3-ASCII-URL-TABLE ではなく [W3-ASCII-URL-TABLE](#) を使用します。

サブプログラム W3-ASCII-URL-TABLE が、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ます。このプログラムは、必要に応じて変更したり拡張したりできます。

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
```


136 時刻／日付文字列

サブプログラム名	ソース名	実行ファイルの例	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: ..." 文字列を作成するか、現在の時刻／日付を使用してオフセットを含む HTTP 互換の時刻／日付文字列を作成します（GMT または GMT へのオフセットが Natural に認識されないため）。

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
```


137

すべての Natural ライブラリのリスト

サブプログラム名	実行ファイルの例	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足し、その日付を有効期限日に設定します。
START=	表示されるオブジェクトセットのワイルドカードを選択します。

呼び出し方法

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


138 オンライン NaturalWeb 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する場合に備えて、Natural Web インターフェイスサブプログラムをオンラインで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 文字より長い行は折り返されます。

環境変数を設定できます。変数をサーバー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

Natural の *next* プロンプトからプログラム WEB-ONL を実行します。

139 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、次のバージョンのいずれかで削除されます。Windows では **Web インターフェイスプラグイン** を使用します。

入力マップ

```
12:12:40          ***** Natural Web Subprogram Wizard *****          2005-11-15
                                     - 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

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

140 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データのリスト

サブプログラム名	実行ファイルの例	表示可能ファイルの例
NAT-DATA	NAT-DATA NAT-DATA?parm1=test1&parm2=test2 E3MULTIP	NAT-DATA E3MULTIP

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データが記載された HTML ページを生成します。MIME タイプ、HTTP ヘッダー、パラメータ、英数字データ、およびバイナリデータが含まれます。

パラメータ

```
/* none
```

呼び出し方法

```
NAT-DATA
```


141 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=sysweb3&version=no&start=E3* 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。

Natural バージョン 6.2 の新機能

サブプログラムモジュールのバージョンチェックが追加されました。ライブラリ SYSWEB でカタログされている場合は SYSWEB、ライブラリ SYSWEB3 でカタログされている場合は SYSWEB3 が表示されます。NAT-DIR が SYSWEB インターフェイスで呼び出された場合は、実行リンクは表示されません。

パラメータ

LIB=	Natural ライブラリ
EXPIRE=	現在の日付に日数を足し、その日付を有効期限日に設定します。
START=	表示されるオブジェクトセットのワイルドカードを選択します。
VERSION=	NO に設定すると、ランタイムに使用されるサブプログラムの SYSWEB バージョン (SYSWEB または SYSWEB3) がチェックされません。デフォルトは YES です。

呼び出し方法

```
nat-dir?lib=sysweb3
```

142

Natural ライブラリのリソースのリスト

サブプログラム名	実行ファイルの例	表示可能ファイルの例
NAT-DIRR	NAT-DIRR?lib=sysweb3 NAT-DIRR?lib=sysweb3&start=*.jpg NAT-DIRR?lib=sysweb&start=*.htm	NAT-DIRR

説明

Natural ライブラリのリソースファイル情報が記載された 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足し、その日付を有効期限日に設定します。
START=	表示されるオブジェクトセットのワイルドカードを選択します。

呼び出し方法

```
nat-dirr?lib=sysweb3
```


143 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
```


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

サブプログラム名	実行ファイルの例	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る 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
```


145 Natural Web インターフェイスの現在の設定のリスト

サブプログラム名	実行ファイルの例	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
```


146

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足し、その日付を有効期限日に設定します。
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
```


147 オンラインドキュメント

サブプログラム名	実行ファイルの例
NAT-DOCU	NAT-DOCU

説明

Natural ソースオブジェクトとして保存される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足し、その日付を有効期限日に設定します。

呼び出し方法

```
nat-docu
```


148

Natural 以外のファイル／リソースのリスト

サブプログラム名	実行ファイルの例	表示可能ファイルの例
NAT-RES	NAT-RES?source=e3xslt2.xml&lib=sysweb NAT-RES?source=composing_natural_logo.jpg&lib=sysweb3 NAT-RES/sysweb3/e3put.html	NAT-RES

制限事項

共有リソースを備えたプラットフォームでのみ使用可能です。

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。データは、テキストメンバ **MIMEDATA** の定義に応じてバイナリまたは英数字データとして転送されます。

パラメータ

LIB=	Natural ライブラリ
SOURCE=	リソース名 (拡張子付き)

呼び出し方法

```
nat-res?source=e3xslt2.xml&lib=sysweb3
```

拡張 URL 構文による呼び出し方法

パラメータ LIB= および SOURCE= を指定する代わりに、プログラム名の直後にライブラリおよびソース名を追加できます。 nat-res/<ライブラリ>/<リソース>

拡張機能

HTTP サーバーが PUT 対応の場合は、NAT-RES を使用して、HTTP PUT 要求で HTTP サーバーに書き戻すことができます。

PUT 機能を有効にするには、サブプログラム NAT-RES の変数 F_PUT を TRUE に設定し、サブプログラム NAT-RES を再カタログします。

149 XML ツールキットプラグイン

XML ツールキットプラグインにより、開発者は Natural 内で XML ドキュメントを処理できます。

ツールキットには、Natural ソースコードを生成し、次の機能を提供するウィザードが用意されています。

- DTD または XML スキーマへの Natural データ定義のマッピング
- Natural データ構造のシリアライズおよび XML ファイルへのその内容の割り当て
- Natural データ定義への DTD のマッピング
- XML ファイルの解析および Natural データ構造へのその内容の割り当て

ウィザードはライブラリ SYSXTK に含まれています。

このドキュメントでは、外部プログラム部分のない Natural for Windows 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。

150 はじめに

■ XML ツールキットの特徴	410
■ XML ツールキットの説明	410
■ 考慮事項と制限事項	414

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。
 - DTD 定義への Natural データ構造の変換
 - 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 ツールキットの目的は、Natural に追加の XML 機能を提供し、Natural アプリケーションと XML の親和性を改善することです。

一般的なアーキテクチャ

Natural XML ツールキットは、Natural プラグインとして実装されています。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して、XML データへのアクセスを可能にしたり、Natural からデータを XML フォーマットで提供したりすることができます。

Natural XML ツールキットは、次の機能を呼び出します。

XML ツールキット機能

1. Natural データ定義と DTD または XML スキーマの間のマッピング
2. XML トークンから NAT 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データ構造に保存する必要があります。所定のデータを Natural データ構造に保存できる Natural 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り出し、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です。

Natural ダイアログは XML ツールキット機能へのユーザーインターフェイスを実装しています。DTD または XML スキーマ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。

DTD への Natural データ定義のマップ

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、XML タグとして Natural データを表現するために必要です。以下の例は、Natural と DTD の間のマッピングと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: SYSXTK  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ライズで使用されます（下記参照）。

XML へのデータのシリアライズ

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、シリアライズ中に XML フォーマットでダイナミック変数に書き込まれ、以前に生成された DTD を入力として使用します。

XML ツールキットは、データをシリアライズするプログラムを生成します。

```
<?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>
...
```

Natural データ定義への DTD のマップ

Natural データ構造への DTD のマッピングについて、もう一度違いを示します。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ので、ツールキットで最大 "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が正確な数を把握すれば、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 には、各レコードの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 (A) DYNAMIC（現在の最大長）のフィールドを作成します。

```

* DTD E:\SAG\nat\6.3\fnat\SYSXTK\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\6.3\fnat\SYSXTK\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&

```

...

考慮事項と制限事項

XML ツールキットは、完全にアSEMBルされたXMLスキーマ（レイヤ1）のみをサポートします。詳細については、[XML スキーマ（レイヤ1）に関する W3C 勧告（英語）](#)を参照してください。

XML ツールキットを使用する場合は、その他に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
- 多次元配列
- XML スキーマ：アクセスと構成
- DTD：外部解析データの追加
- 条件付きの DTD
- ワイルドカード

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、以下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てください。

多次元配列

次の制限は、ソースデータに配列が含まれている場合に、[Natural データエリア](#)からXMLドキュメントを生成するときに適用されます。

- 各配列では厳密に1次元配列のみ使用できます。
- 各レベルでは厳密に1次元のみ追加できます。
- 各配列にはカウンタ変数が必要です。
- カウンタ変数には以下の制限があります。
 - カウンタ変数は配列の前に配置する必要があります。

- カウンタ変数名は、先頭が文字 C で、それに **カウンタセパレータフィールド** を続けた名前にする必要があります。
- 配列の *LBOUND (下限) は 1 にする必要があります。

XML スキーマ：アクセスと構成

<include>

ターゲット名前空間が同じ複数のスキーマをドキュメントに追加します。ドキュメントは一切の変更なしにインクルードされる必要があります。

<import>

ターゲット名前空間が異なる複数のスキーマをドキュメントに追加します。まず、インポートするドキュメントで名前空間接の頭辞を変換する必要があります。その後、ドキュメントをインクルードできます。

<redefine>

特定の単一または複合タイプ、グループ、および属性グループを外部スキーマから選択することで、元の仕様を必要に応じて変更できます。



Note: 上記の要素を使用する場合は、相対 URI のみ使用できます。絶対 URI (例：<http://www.yourdomain.com/your/path> または <file://your/path>) は使用できません。

DTD：外部解析データの追加

外部データをドキュメントにインクルードする必要があります。変換は必要ありません。

条件付きの DTD

<![INCLUDE] が見つかった場合は、そこに含まれる定義が生成に使用されます。

<![IGNORE] が見つかった場合は、そこに含まれる定義は生成に使用されません。

ワイルドカード

XML ツールキットでは、次の 2 種類のワイルドカード表現がサポートされています。


- 後続の要素をすべて保存する
- 指定されていないすべての属性を保存する

ワイルドカードサポートには、次のルールと制限が適用されます。

XML スキーマ <anyAttribute>

属性の場合は、`attributes_of_<entity-name>` グループが生成されます。このグループに接続されているすべての属性が追加されます。属性の名前は変数名として保存され、内容は変数の内容になります。

<any>属性を追加するには、指定されていないすべての属性を含む変数を追加する必要があります。

 **Note:** <any> 属性は、"実際の" 属性ではありません。解析されないデータのコンテナとして使用され、属性/値のペアが含まれます。<any> 属性は、タイプ (A) DYNAMIC の ##ANY 変数によって表されます。

このデータにアクセスする必要がある場合があるため、##ANY の後に一般的な数字を付けるだけでなく、より具体的な名前を使用する必要があります。親エンティティの名前とキーワード ATTR または ATTRIBUTE を追加することをお勧めします。以下の例を参照してください。


```
1 HTML
2 BODY
3 ATTRIBUTES_OF_BODY
4 BGCOLOR (A) DYNAMIC
4 ##ANY_ATTR_BODY (A) DYNAMIC
```

解析中に XML スキーマ内に名前のない属性が見つかった場合は、変数名と値は all_attributes_of_<element-name> グループにそのまま (標準 XML 構文で) 保存されます。

<attribute-name1>=<attribute-value1> <attribute-name2>=<attribute-value2> など
シリアル化時に上記の文字列が追加されます。

XML スキーマ <any> または DTD <!ELEMENT element-name ANY>

<any> データタイプを追加するには、エンティティの名前と値に関係なく、エンティティの後続のすべてのデータを保存する必要があります。

 **Note:** <any> エンティティは "実際の" エンティティを表してはいません。解析されないデータのコンテナとして使用され、エンティティとその内容全体 (属性など) が含まれます。<any> エンティティは、タイプ (A) DYNAMIC の ##ANY 変数によって表されます。

このデータにアクセスする必要がある場合があるため、##ANY の後に一般的な数字を付けるだけでなく、より具体的な名前を使用する必要があります。親エンティティの名前を追加することをお勧めします。以下の例を参照してください。

```
1 HTML
2 BODY
3 ATTRIBUTES_OF_BODY
4 BGCOLOR (A) DYNAMIC
4 ##ANY_ATTR_BODY (A) DYNAMIC
3 ##ANY_BODY (A) DYNAMIC
```

解析中にタイプ <any> の要素が見つかった場合は、後続のすべてのデータが収集されます。

シリアライズ時に、すべてのデータは変更なしに取り出され、結果のXMLドキュメントに追加されます。

xs:any の制限

xs:any の属性 "maxOccurs" または "minOccurs" が指定された場合でも、xs:any を実装する Natural 変数は常にスカラーになります。この Natural 変数には複数のエンティティのデータが含まれることがあります。

xs:any の属性 "namespace" は評価されず、xs:any を実装する Natural 変数に複数の名前空間のエンティティが含まれることがあります。

使用されるパーサーは検証を行わないため、xs:any の属性 "processContents" は評価されません。

解析中に複数の <any> コンテナを認識することはできないため、xs:choice または xs:sequence に xs:any の定義が複数含まれていると生成に失敗します。

XML スキーマで指定されていないエンティティがドキュメントに含まれており、同じレベルに xs:any が定義されている場合は、xs:any を実装する Natural 変数にこの "無指定の" エンティティデータが含まれることがあります。

Natural : ##ANY ワイルドカードを含む XML スキーマ または DTD の生成

外部データ構造の生成中、##ANY が前置された各変数は特定のワイルドカード機能に変換されません。

- ##ANY_ : any エンティティタイプ。DTD と XML スキーマに適用されます。
- ##ANY_ATTR_ : any 属性タイプ。XML スキーマのみに適用されます。

151 XML ツールキットの使用

■ 必要条件	420
■ アプリケーションの起動	420
■ ヘルプ	421

次のトピックについて説明します。

必要条件

Natural バージョン

Natural for Windows バージョン 6 が必要です。

アプリケーションの起動

XML ツールキットはライブラリ SYSXTK に含まれています。

▶手順 151.1. XML ツールキットを使用するには

- ライブラリワークスペースからライブラリ SYSXTK を選択します。

Dialogs フォルダでダイアログ Menu を選択します。

Or:

Natural コマンド行で、「LOGON SYSXTK」と入力します。

「Menu」と入力します。

ダイアログウィザードの最初の画面が表示されます。

次の機能を使用できます。

- **Natural** データ構造から生成 Natural データエリアをデータソースとして使用します。
- 文書型定義/XML スキーマから生成 文書型定義 (.dtd)、XML スキーマ (.xsd)、または Tamino スキーマ 2 (.tsd) 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

ヘルプ

▶手順 151.2. XML ツールキットでヘルプを呼び出すには

- XML ツールキットのメインダイアログボックスで、[Help] メニューの [XML Toolkit Help] を選択します。

Or:

XML ツールキットのメインダイアログボックスで、F1 キーを押します。

152 特定の生成オプションの設定

■ Natural XML オプションメニューの起動	424
■ Generation	424
■ Path	427
■ オプションの永続的な保存	429

生成オプションは、3つのタブに配置され、生成、ビュー、および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。

Natural XML オプションメニューの起動

▶手順 152.1. Natural XML オプションメニューを起動するには

- [Generate] メニューの [Options] を選択します。

[NaturalXMLOptions] の [Generation] タブが表示されます。メニュータブのフィールドについては以下で説明します。

Generation

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。以下のメニューで、デフォルトの変換設定を必要に応じて変更できます。



フィールドの説明

Counter Separator Character

所属グループ:	Additional Fields
デフォルト値:	@

Namespace Separator Character ":" WITH

所属グループ:	XML Name Replacements
デフォルト値:	\$

Dot Sign '.' WITH

所属グループ:	XML Name Replacements
デフォルト値:	/

Plus Sign '+' WITH

所属グループ:	XML Variable Name Replacements
デフォルト値:	plus

Hash / Number Sign '#' WITH

所属グループ:	Natural Variable Name Replacements
デフォルト値:	hash

Slash Sign '/' WITH

所属グループ:	Natural Variable Name Replacements
デフォルト値:	slash

At Sign '@' WITH

所属グループ:	Natural Variable Name Replacements
デフォルト値:	at

Paragraph Sign '\$' WITH

所属グループ:	Natural Variable Name Replacements
デフォルト値:	para

Ampersand Sign '&' WITH

所属グループ:	Natural Variable Name Replacements
デフォルト値:	amp

Dollar Sign '\$' WITH

所属グループ:	Natural Variable Name Replacements
デフォルト値:	dollar

Parse Statement

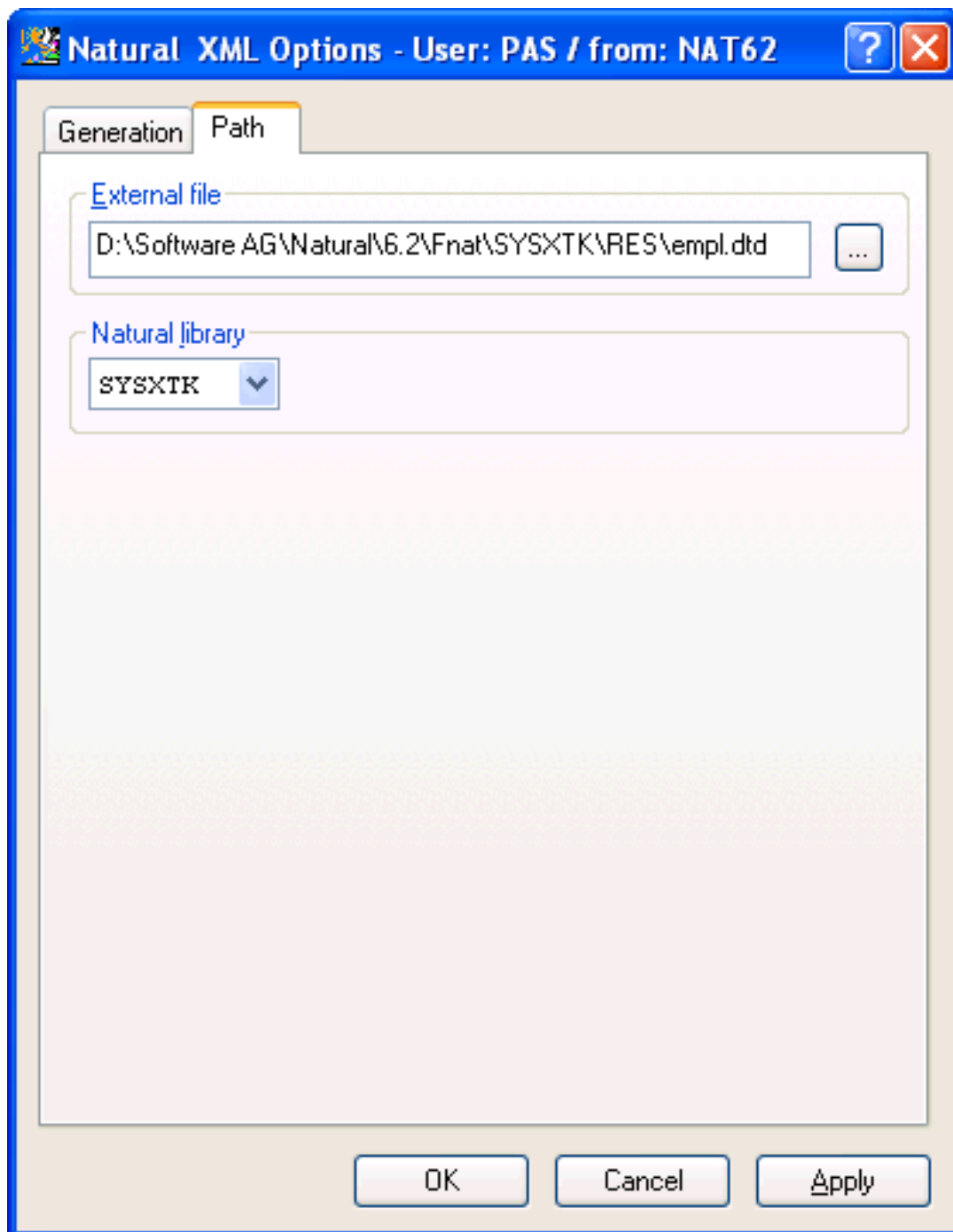
所属グループ:	Code Generation
デフォルト値:	選択 (はい)

X-Arrays

所属グループ:	Code Generation
デフォルト値:	選択 (はい)

Path

[Natural XML Options] の [Path] タブでは、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。



フィールドの説明

External file

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

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

[Natural XML Options] 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、[Natural XML Options] の設定はすべて上書きされます。

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 に保存します。

153 Natural データソースの使用

■ Natural データエリアの選択	432
■ データタイプの選択	434
■ DTD 定義または XML スキーマでのファイルの生成	435
■ XML ドキュメントのシリアライズの生成	437
■ XML ドキュメントのパースの生成	438
■ パラメータの設定	439
■ ルートグループの選択	440
■ 生成レポートの表示	440

この機能により、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](#)

Natural データエリアの選択

このダイアログボックスでは、Natural データ構造からの生成または XML スキーマ／文書型定義からの生成を選択します。

▶ **手順 153.1.** 以下に示されているダイアログを呼び出すには、次の手順に従います。

- 1 [ツール] > [コンフィグレーションツール] > [プラグインマネージャ] > [XML Toolkit] を選択して、XML プラグインを起動します。

Or:

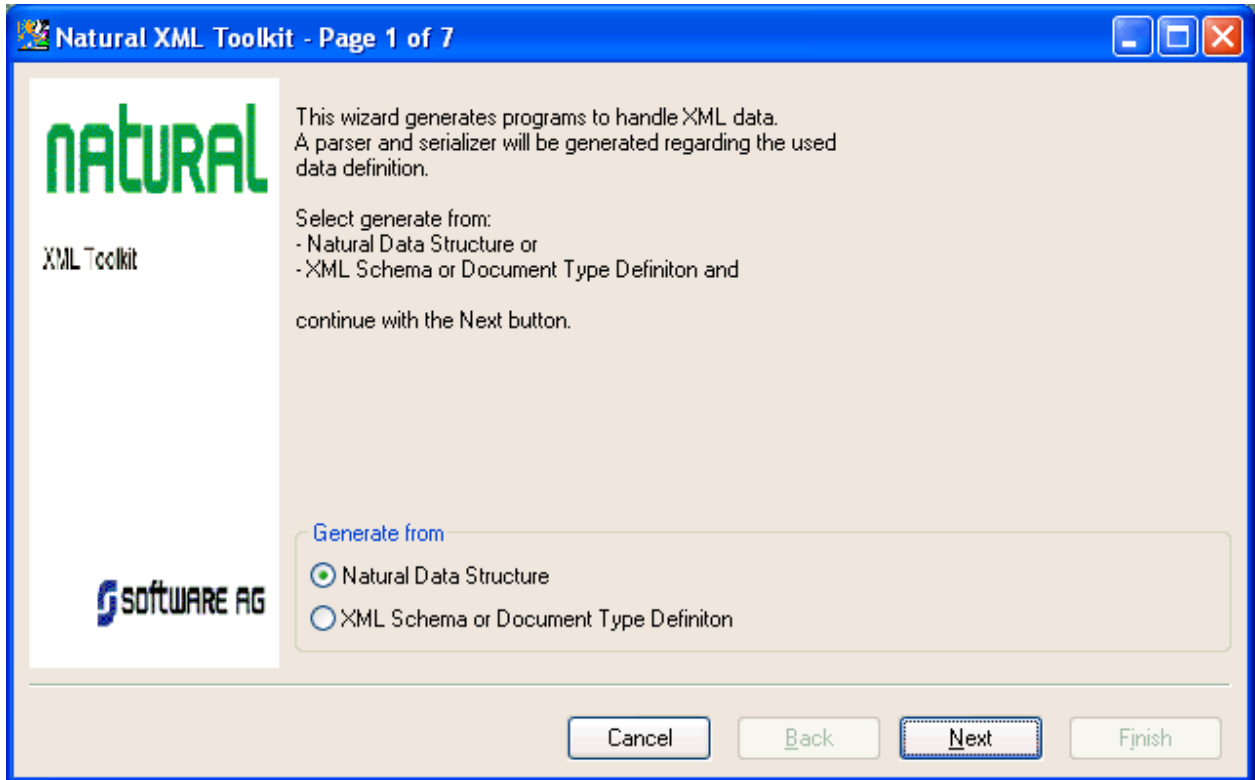


ボタンをクリックして、プラグインマネージャを開きます。

- 2 [ツール] > [開発ツール] > [XML Toolkit] を選択します。



Note: 以下のダイアログボックスに表示されているエントリは、デフォルトまたは例の値です。



生成を、Natural データソースから行うか、XML スキーマまたは DTD から行うかを選択します。

[Next] を選択して次に進みます。

フィールドの説明

Library

所属グループ：	Select Input Data Area
デフォルト値：	(すべてのライブラリ)

タイプ

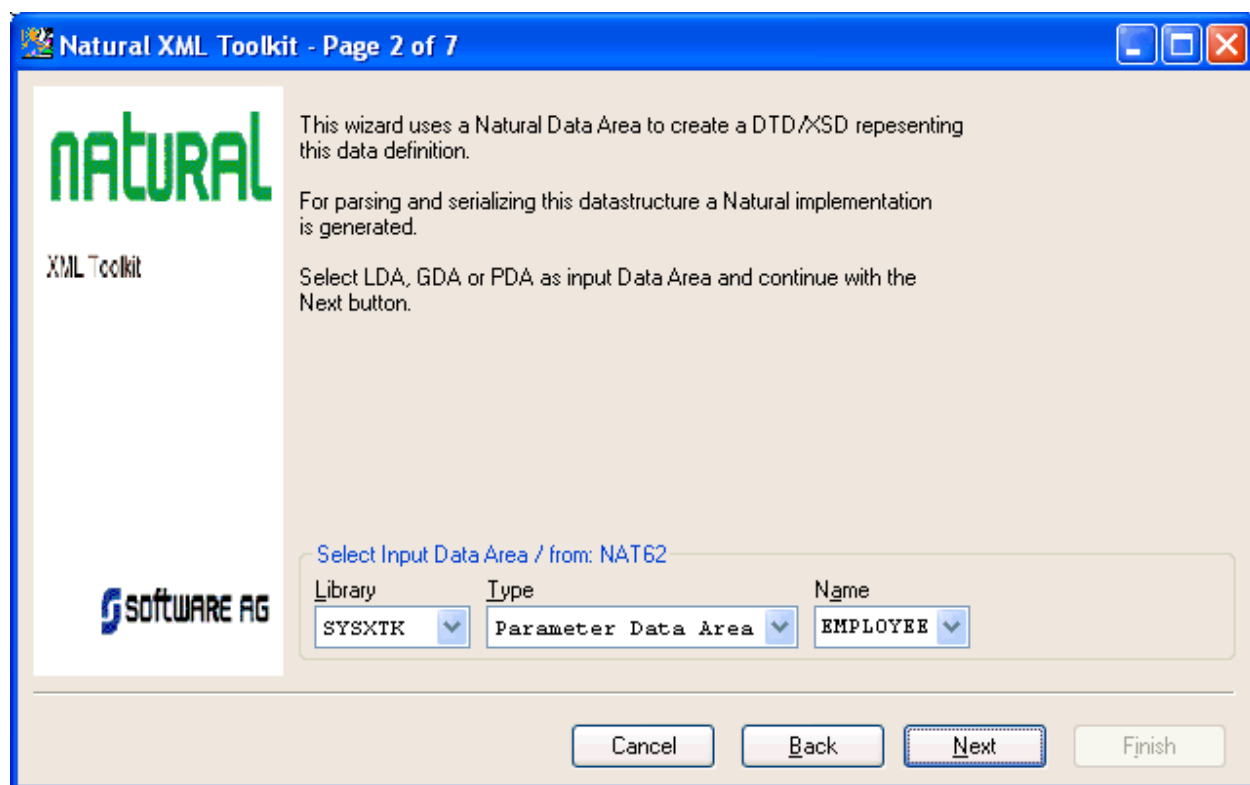
所属グループ：	Select Input Data Area
---------	------------------------

名前

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)

データタイプの選択

このダイアログボックスでは、データタイプを選択します。



フィールドの説明

Library

所属グループ:	Library
デフォルト値:	(すべてのライブラリ)

タイプ

所属グループ:	Type
設定可能値:	ローカルデータエリア パラメータデータエリア グローバルデータエリア

名前

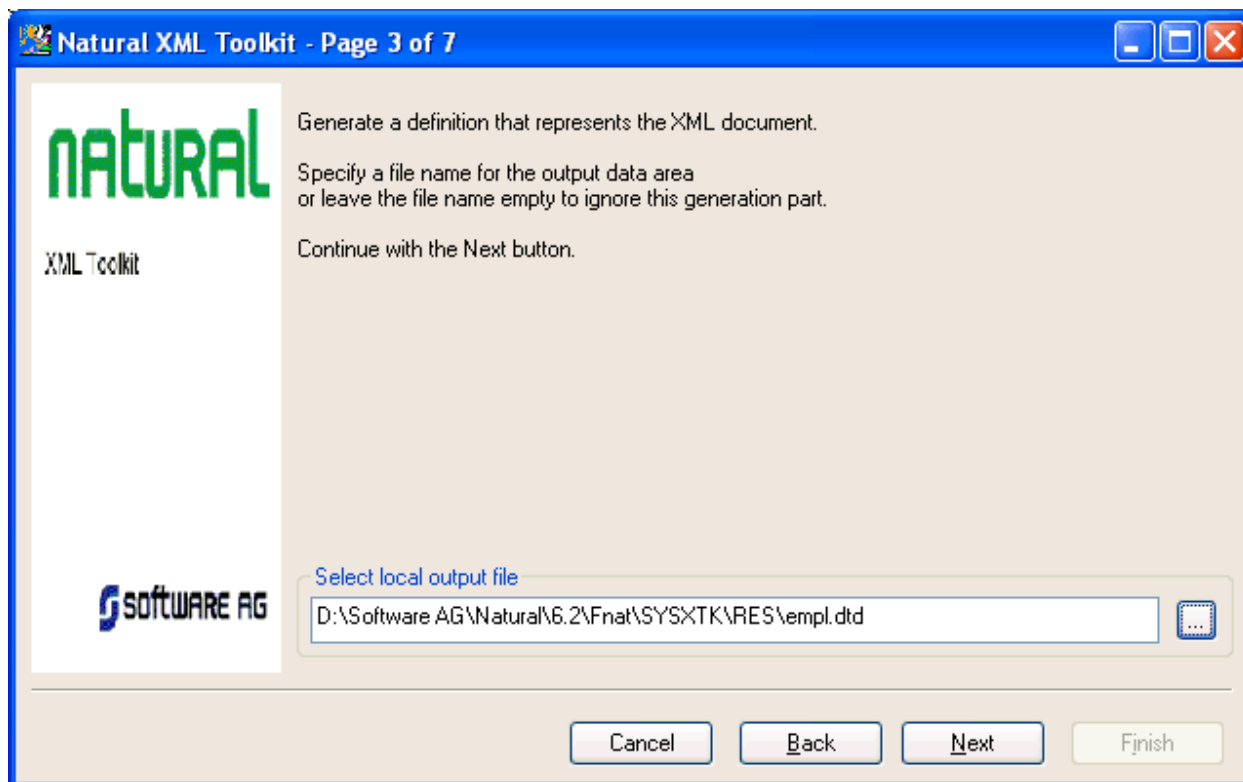
所属グループ:	Name
デフォルト値:	(選択されたライブラリおよびタイプのすべてのオブジェクト)

目的の要素 (EMPLOYEE など) を選択します。

[Next] を選択して次に進みます。

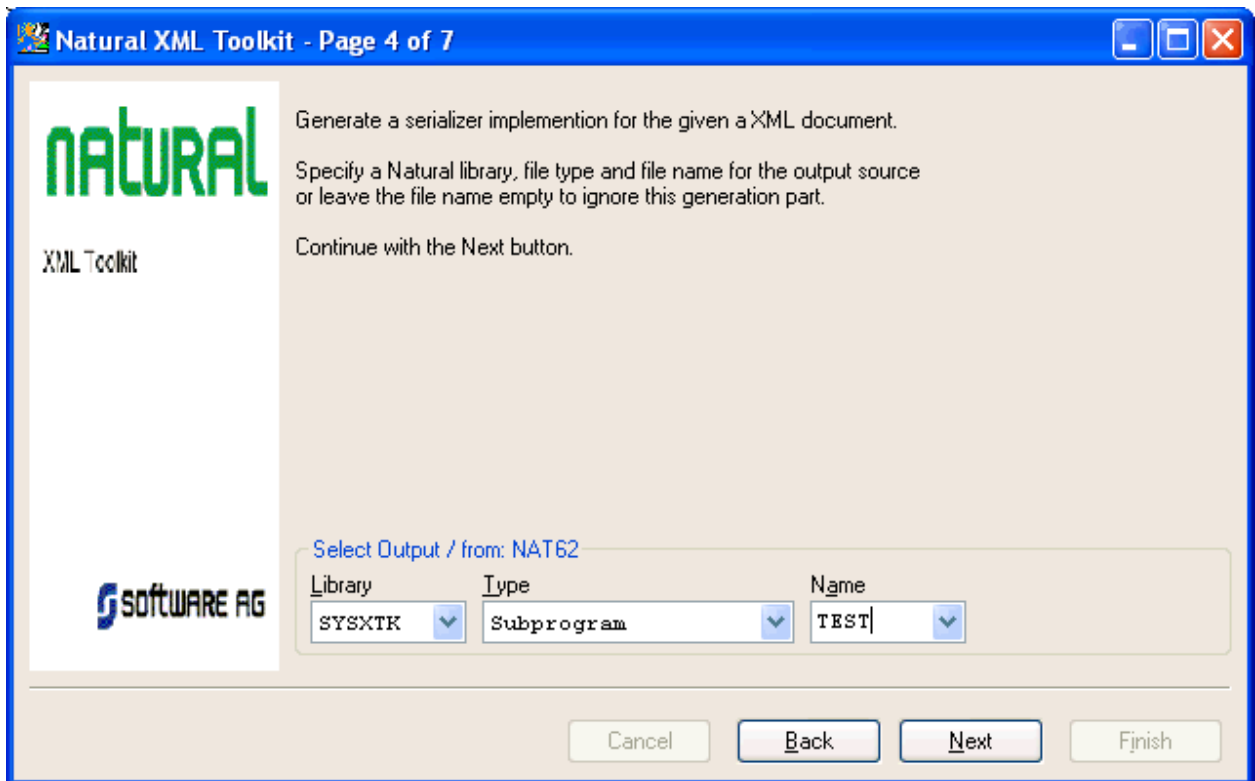
DTD 定義または XML スキーマでのファイルの生成

このダイアログボックスでは、出力データエリアのファイル名を指定できます。



[Next] を選択して次に進みます。

XML ドキュメントのシリアライザの生成



このダイアログボックスでは、出力ソースのNaturalライブラリ、ファイルタイプ、およびファイル名を指定します。

フィールドの説明

Library

所属グループ:	Library
デフォルト値:	(すべてのライブラリ)

タイプ

所属グループ:	Type
設定可能値:	Copycode Subprogram

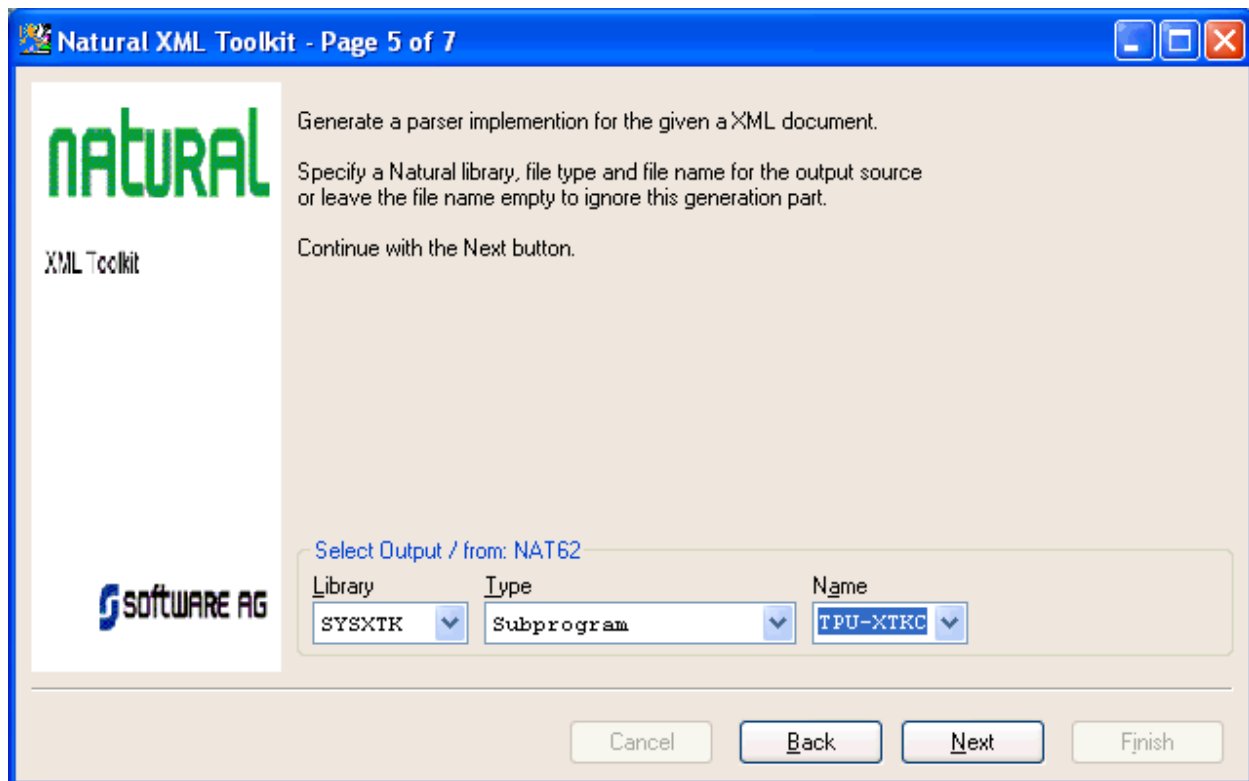
名前

所属グループ:	Name
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)

[Next] を選択して次に進みます。

XML ドキュメントのパーサーの生成

このダイアログは、特定のグループのシリアライゼーションの実装としてコピーコードを XML ドキュメントに生成するために使用します。



フィールドの説明

Library

所属グループ:	Library
デフォルト値:	(すべてのライブラリ)

タイプ

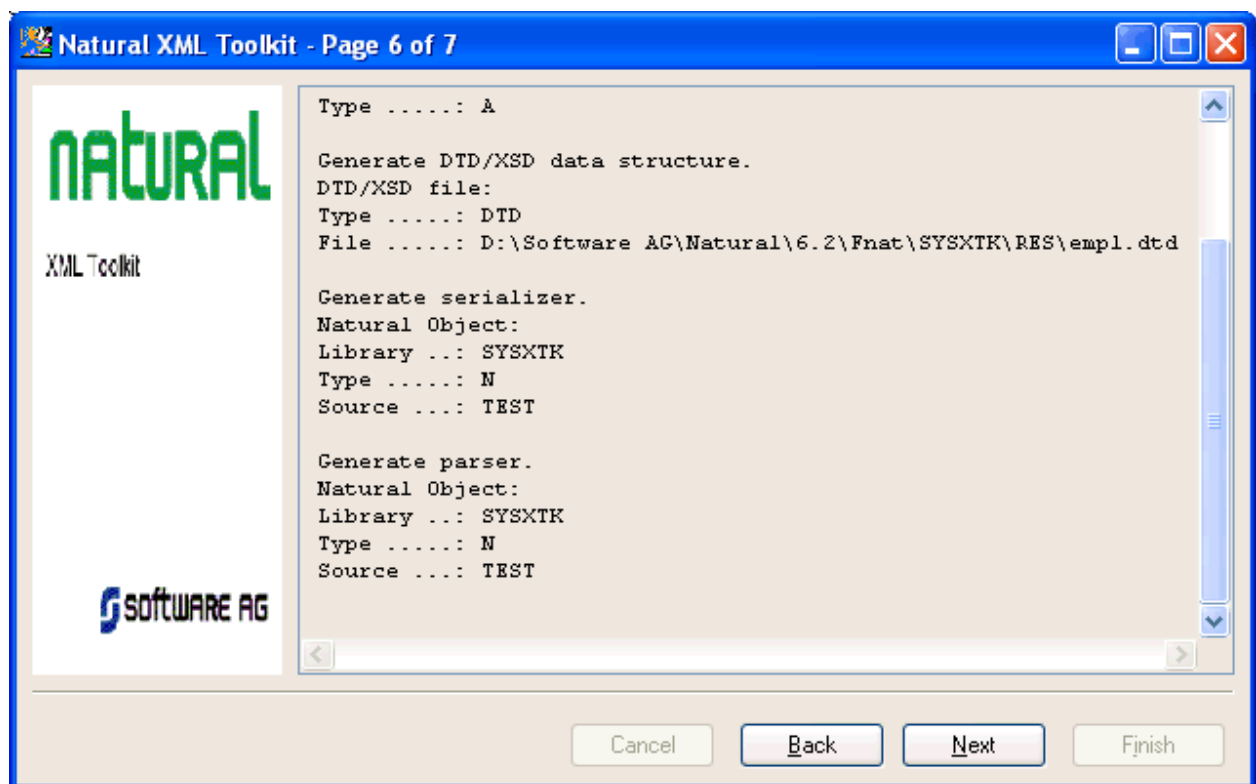
所属グループ:	Type
設定可能値:	Copycode Subprogram

名前

所属グループ:	Name
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)

[Next] を選択して次に進みます。

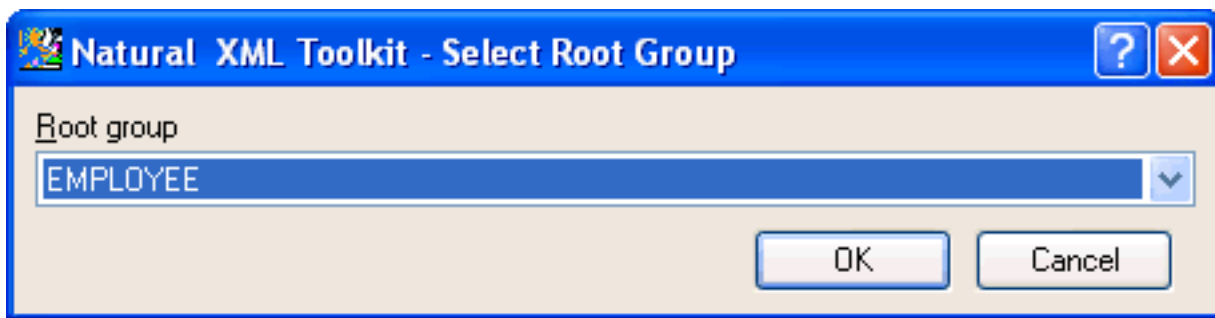
パラメータの設定



この画面には、生成処理に使用される設定が表示されます。

[Next] を選択して次に進みます。

ルートグループの選択

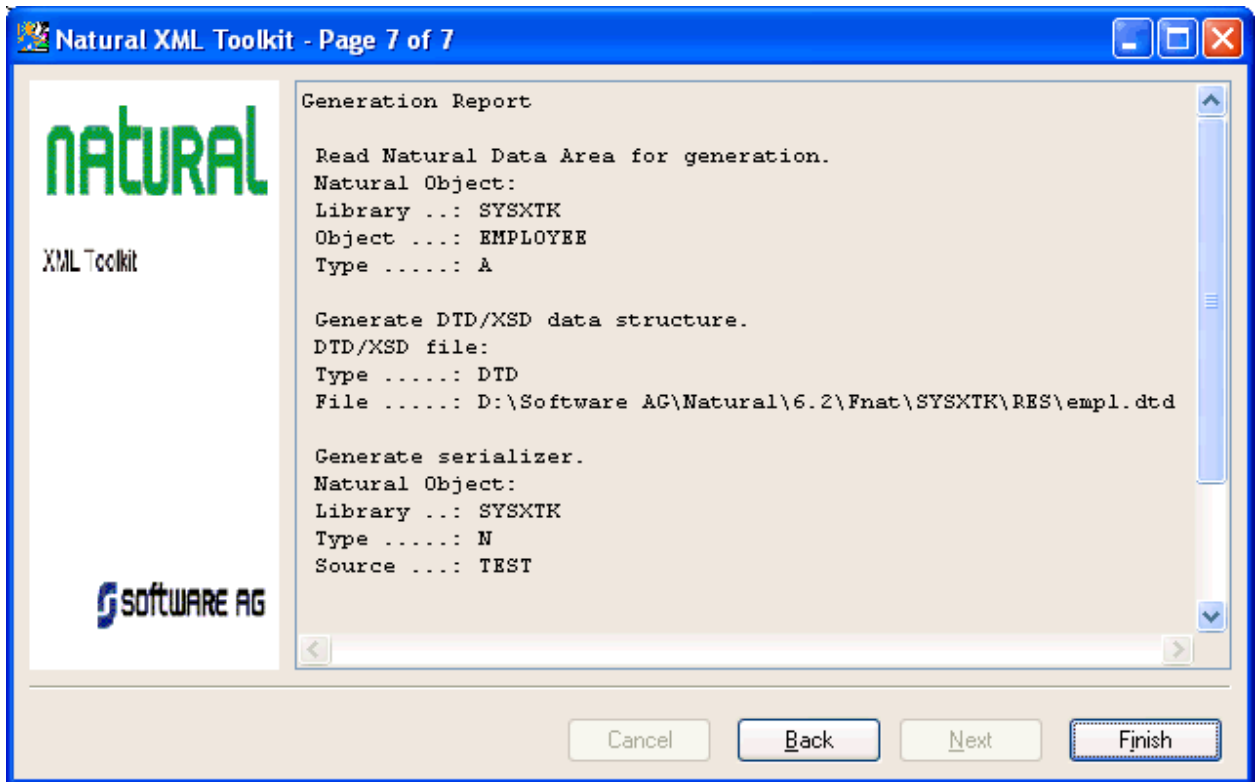


このダイアログボックスでは、ルートグループを選択できます。

[OK] を選択して次に進みます。

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。



[Finish] をクリックして生成処理を終了します。

154 外部データソースの使用

▪ 文書型定義または XML スキーマからの生成	444
▪ ルートエレメントまたは文書型の選択	446
▪ 再帰レベルの選択	446
▪ Natural データエリアの生成	448
▪ シリアライズのためのコピーコードの生成	449
▪ シリアライズのためのサブプログラムの生成	451
▪ XML パーサーコールバックのためのコピーコードの生成	451
▪ XML パーサーコールバックのためのサブプログラムの生成	453
▪ 生成結果の表示	453

この機能により、XML ドキュメントをローカル、グローバル、またはパラメータデータエリアに定義されている Natural 変数に解析できます。

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](#)



Note: 入力文書型として XML スキーマ (XSD) を使用する場合は、最初の XSD 要素がルートエレメントとして使用されます。

文書型定義または XML スキーマからの生成

このダイアログボックスでは、入力文書型として文書型定義 (DTD)、XML スキーマ (XSD)、または Tamino スキーマ 2 (TSD) を選択します。

▶ **手順 154.1.** 以下に示されているダイアログを呼び出すには、次の手順に従います。

- [Generate] メニューの [DTD/XSD] を選択します。

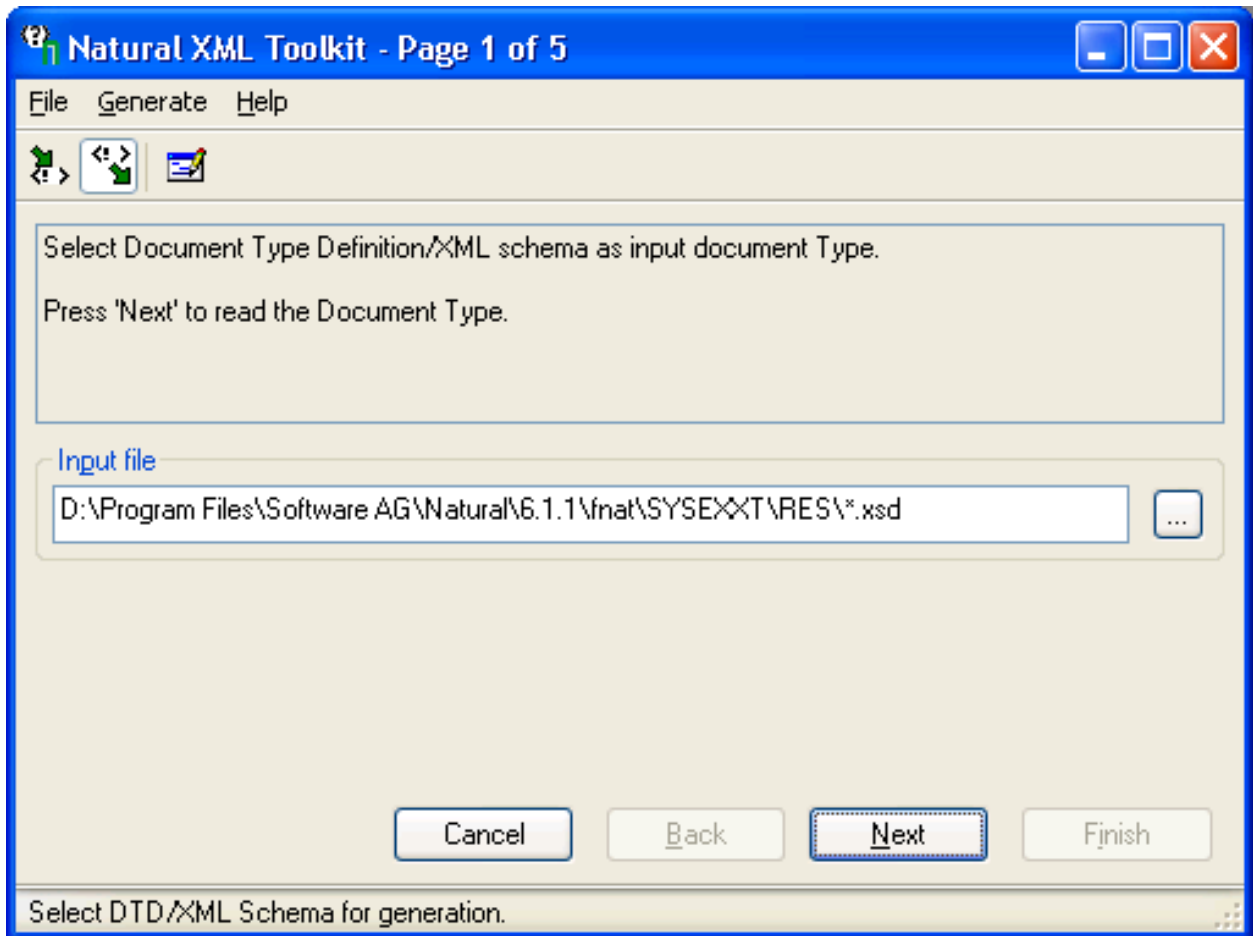
Or:



ボタンをクリックします。




Note: 以下のダイアログボックスに表示されているフィールドエントリは、デフォルトまたは例の値です。



フィールドの説明

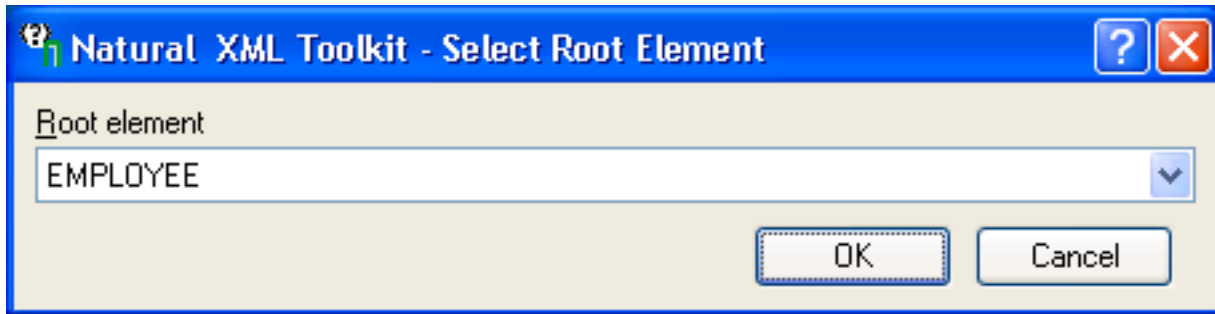
Input File

DTD、XSD、または TSD ファイルを選択します。参照ボタン  を使用して、既存の DTD、XSD、または TSD ファイルを検索できます。

[Next] を選択して次に進みます。

ルートエレメントまたは文書型の選択

このダイアログボックスでは、XML ドキュメントのルートである要素または文書型を選択します。



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) を選択し、[OK] をクリックします。

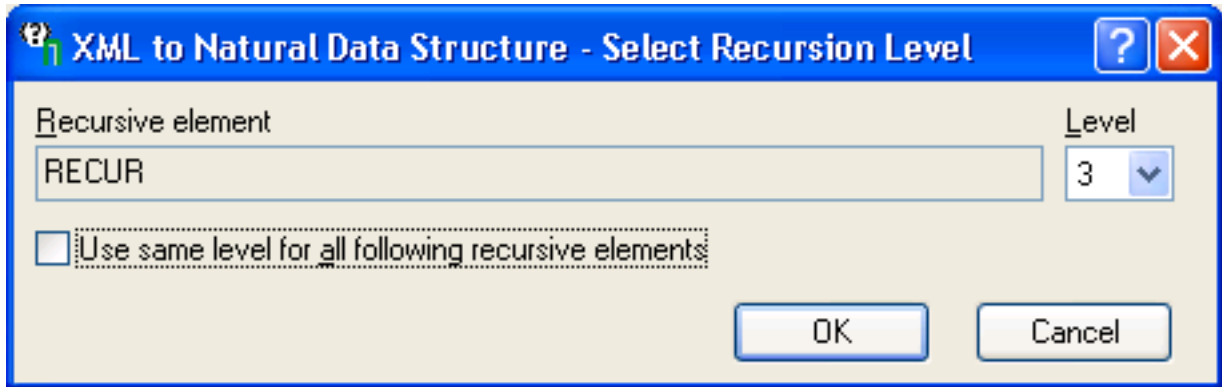
Document Type (Tamino スキーマの場合)

デフォルト値: (すべての要素)

目的の要素 (EMPLOYEE など) を選択し、[OK] をクリックします。

再帰レベルの選択

このダイアログボックスは、最初のダイアログボックスで選択した DTD、XSD、または TSD に再帰エレメントが含まれる場合にのみ表示されます。



フィールドの説明

Recursive Element

再帰的に使用されるエレメントの名前。

デフォルト値： (すべてのライブラリ)

レベル

生成する再帰レベルの数。

デフォルト値： 3

Use same level for all following recursive elements

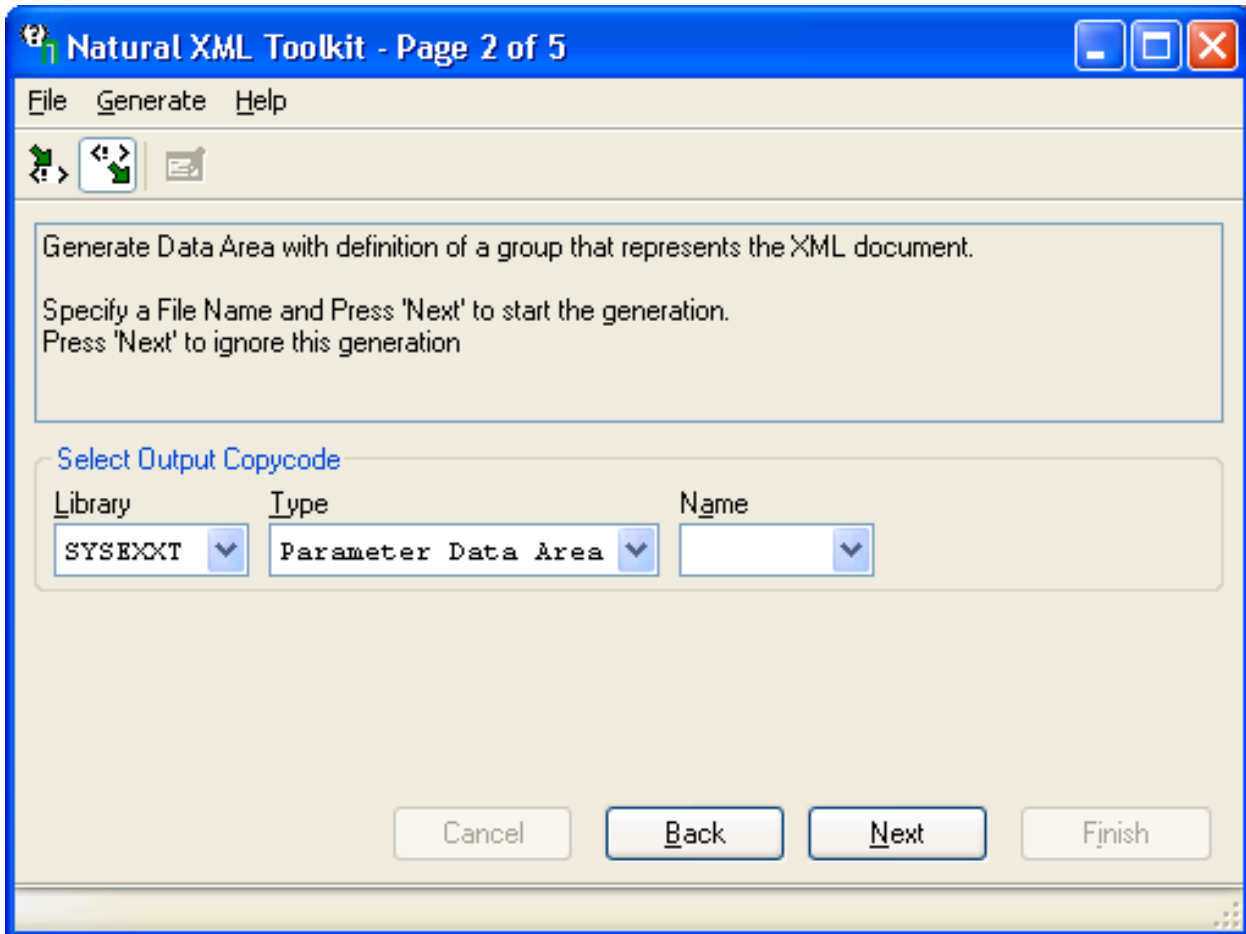
別の再帰エレメントが見つかった場合に、同じ再帰レベルを使用します。

デフォルト値： オフ

[OK] を選択して次に進みます。

Natural データエリアの生成

このダイアログボックス画面では、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。



フィールドの説明

Library

所属グループ:	Select Output Copycode
デフォルト値:	(すべてのライブラリ)

タイプ

所属グループ：	Select Output Copycode
デフォルト値：	L - ローカルデータエリア

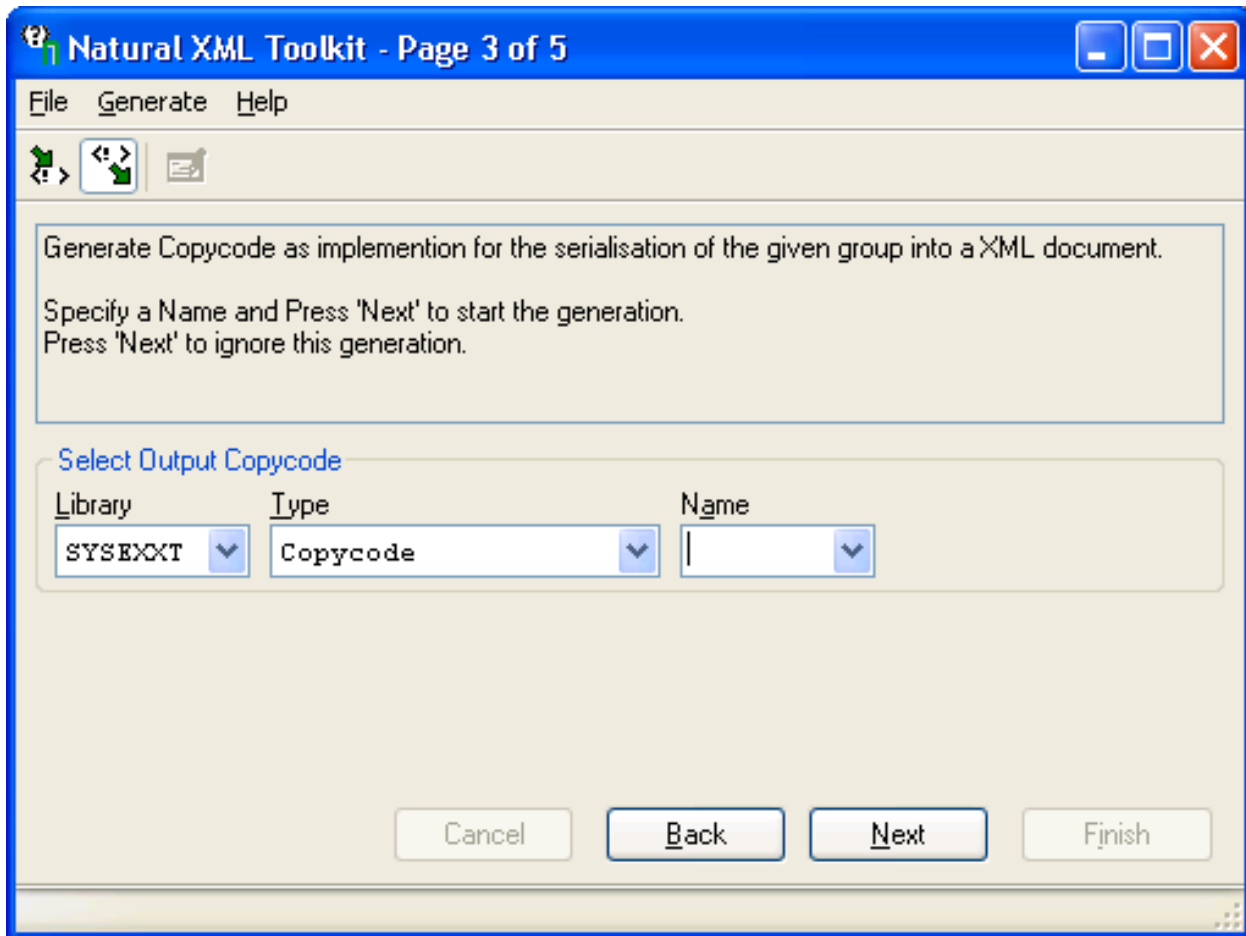
名前

所属グループ：	Select Output Copycode
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)

[Next] を選択して次に進みます。

シリアルイズのためのコピーコードの生成

このダイアログは、特定のグループのシリアルイゼーションの実装としてコピーコードを XML ドキュメントに生成するために使用します。



「例」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
デフォルト値：	(すべてのライブラリ)

タイプ

所属グループ:	Select Output Copycode
デフォルト値:	Copycode

名前

所属グループ:	Select Output Copycode
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)

[Next] を選択して次に進みます。

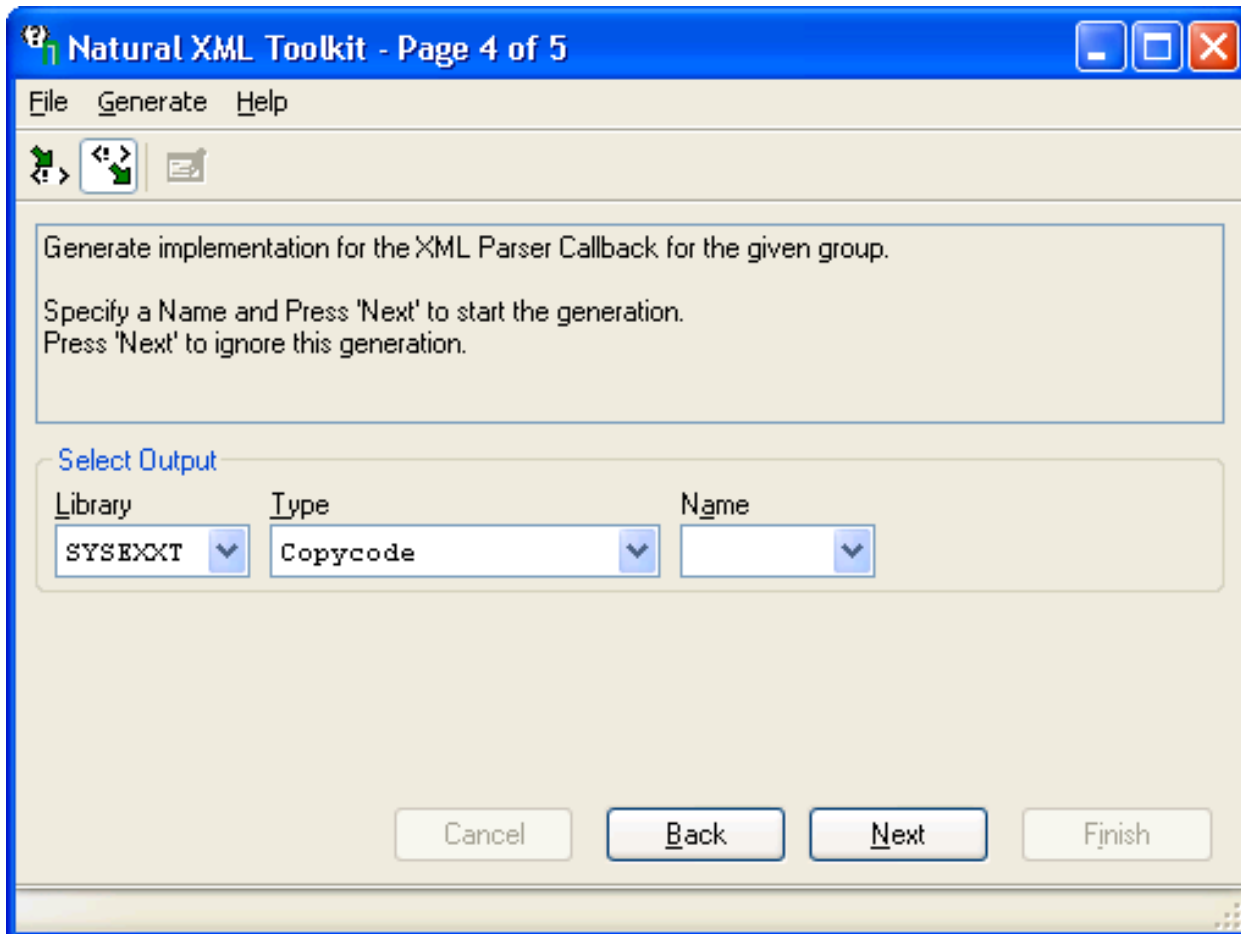
シリアルイズのためのサブプログラムの生成

このダイアログボックスでは、XML ドキュメントへの指定グループのシリアルイズの実装としてサブプログラムを生成します。

上述したダイアログと同じエントリをコピーコードに使用しますが、[Type] フィールドには [Subprogram] というエントリが含まれています。

XML パーサーコールバックのためのコピーコードの生成

このダイアログボックスでは、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。



パーサーの CALLBACK コピーコードを生成します。サンプル文書にある「[パーサーの 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
デフォルト値:	(すべてのライブラリ)

タイプ

所属グループ:	Select Output
デフォルト値:	Copycode

名前

所属グループ:	Select Output
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)

[Next] を選択して次に進みます。

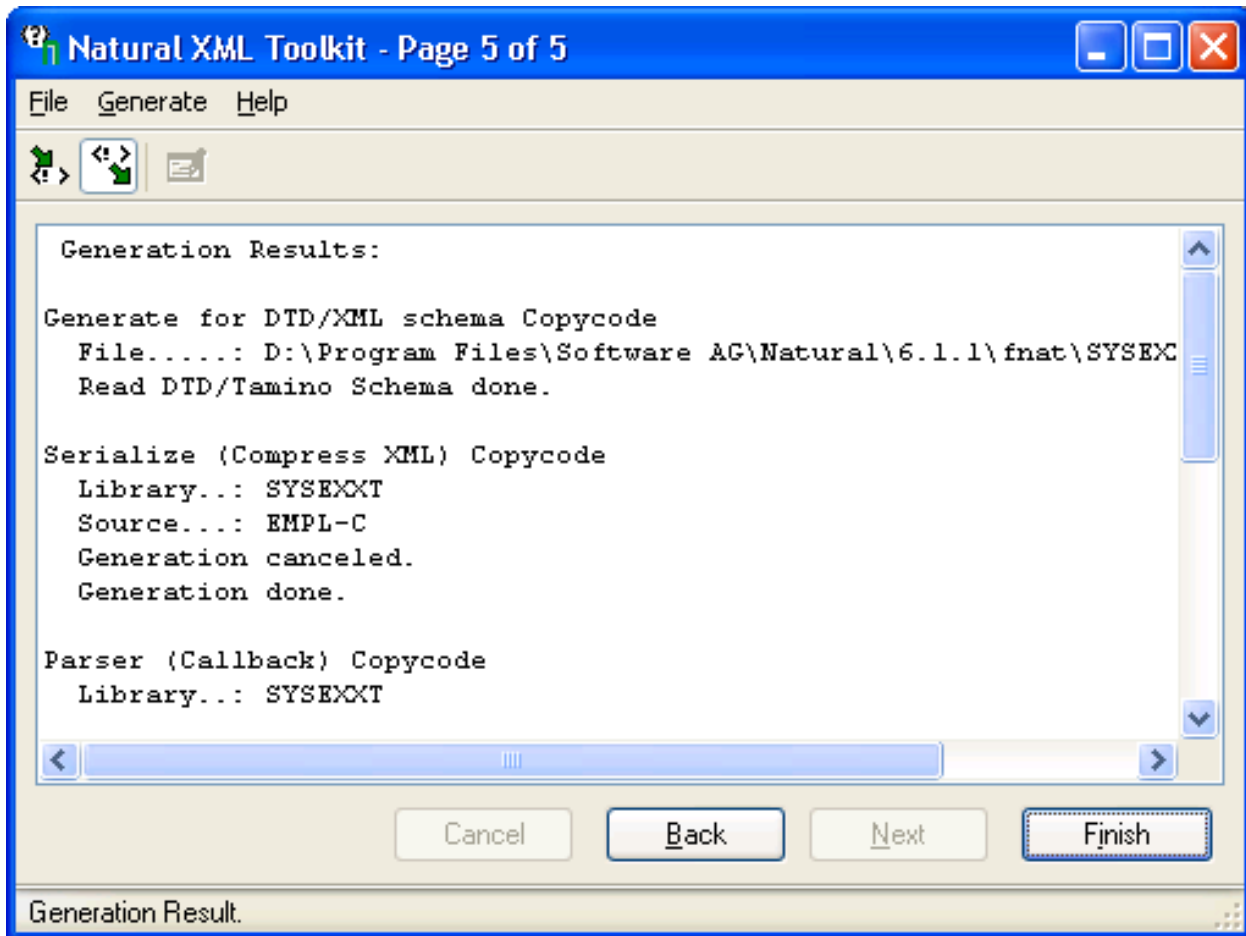
XML パーサーコールバックのためのサブプログラムの生成

このダイアログボックスでは、指定グループのXMLパーサーコールバックの実装としてサブプログラムを生成します。

上述したダイアログと同じエントリをコピーコードに使用しますが、[Type] フィールドには [Subprogram] というエントリが含まれています。

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。



[Finish] をクリックして生成処理を終了します。

155 Natural のシンプルな XML パーサー

■ パーサーの説明と例	456
■ パーサーの制限事項	463

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 同様の構文内で現在の要素、テキスト、コメントの名前を使用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不十分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）。

エラー処理を拡張するために、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は、以下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	解析する XML ファイル
2	A	要素構造を表す ex-XPATH
3	A1	XPATH 内容のタイプ： ? 処理の手順 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	解析したデータ
5	L	解析したデータが空の場合は 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 PARSE_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以下のようになります。

```

<?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>
```



Note: 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以下のようになります。

```
?=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（処理の手順を含む）。開始タグのみを終了タグと等しくする必要があります（処理の手順を含む）。

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／処理の手順の内容

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）

156 例

■ コピーコードのシリアライズ	466
■ サブルーチンのシリアライズ	468
■ 生成された Natural データエリア	476
■ Natural DTD パーサー	478
■ 生成されたタイプ定義	479
■ パーサー CALLBACK コピーコード	480

以下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は、次のオペランドをとります。

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 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	解析したデータ	オペランド 4
4	L	解析したデータが空の場合は 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&,&3&) 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
```

XML スキーマの例：

```
<?xml
version="1.0" encoding="ISO-8859-1"?> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="EMPLOYEE"> <xs:complexType> <xs:sequence>
<xs:element ref="FULL-NAME"/> <xs:element ref="FULL-ADDRESS"/>
<xs:element ref="TELEPHONE"/> <xs:element ref="JOB-TITLE"/>
<xs:element ref="INCOME" minOccurs="0" maxOccurs="6"/>
</xs:sequence> <xs:attribute name="PERSONNEL-ID" use="optional">
<xs:simpleType> <xs:restriction base="xs:string"/> </xs:simpleType>
</xs:attribute> </xs:complexType> </xs:element> <xs:element
name="FULL-NAME"> <xs:complexType> <xs:sequence> <xs:element
ref="FIRST-NAME"/> <xs:element ref="NAME"/> </xs:sequence>
</xs:complexType> </xs:element> <xs:element name="FIRST-NAME">
<xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength
value="20"/> </xs:restriction> </xs:simpleType> </xs:element>
<xs:element name="NAME"> <xs:simpleType> <xs:restriction
base="xs:string"> <xs:maxLength value="20"/> </xs:restriction>
</xs:simpleType> </xs:element> <xs:element name="FULL-ADDRESS">
<xs:complexType> <xs:sequence> <xs:element ref="ADDRESS-LINE"
minOccurs="0" maxOccurs="6"/> <xs:element ref="CITY"/>
<xs:element ref="ZIP"/> <xs:element ref="COUNTRY"/>
</xs:sequence> </xs:complexType> </xs:element> <xs:element
name="ADDRESS-LINE"> <xs:simpleType> <xs:restriction base="xs:string">
<xs:maxLength value="20"/> </xs:restriction> </xs:simpleType>
</xs:element> <xs:element name="CITY"> <xs:simpleType>
<xs:restriction base="xs:string"> <xs:maxLength value="20"/>
</xs:restriction> </xs:simpleType> </xs:element> <xs:element
```

```

name="ZIP"> <xs:simpleType> <xs:restriction base="xs:string">
<xs:maxLength value="20"/> </xs:restriction> </xs:simpleType>
</xs:element> <xs:element name="COUNTRY"> <xs:simpleType>
<xs:restriction base="xs:string"> <xs:maxLength value="3"/>
</xs:restriction> </xs:simpleType> </xs:element> <xs:element
name="TELEPHONE"> <xs:complexType> <xs:sequence> <xs:element
ref="AREA-CODE"/> <xs:element ref="PHONE"/> </xs:sequence>
</xs:complexType> </xs:element> <xs:element name="AREA-CODE">
<xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength
value="6"/> </xs:restriction> </xs:simpleType> </xs:element>
<xs:element name="PHONE"> <xs:simpleType> <xs:restriction
base="xs:string"> <xs:maxLength value="15"/> </xs:restriction>
</xs:simpleType> </xs:element> <xs:element name="JOB-TITLE">
<xs:simpleType> <xs:restriction base="xs:string"> <xs:maxLength
value="25"/> </xs:restriction> </xs:simpleType> </xs:element>
<xs:element name="INCOME"> <xs:complexType> <xs:sequence>
<xs:element ref="SALARY"/> <xs:element ref="BONUS"
minOccurs="0" maxOccurs="4"/> </xs:sequence> </xs:complexType>
</xs:element> <xs:element name="SALARY"> <xs:simpleType>
<xs:restriction base="xs:string"> <xs:maxLength value="9"/>
</xs:restriction> </xs:simpleType> </xs:element> <xs:element
name="BONUS"> <xs:simpleType> <xs:restriction base="xs:string">
<xs:maxLength value="9"/> </xs:restriction> </xs:simpleType>
</xs:element> </xs:schema>

```

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

```

サブルーチンのシリアライズ

XML ツールキットを使用して、Natural グループ構造を XML スキーマに変換するサブルーチンを生成できます。

サブルーチン **EMPLP** の例 :

```

* -----
* Generated from NATURAL XML TOOLKIT
*
*       'EMPLP'
*
* DESCRIPTION
*
*       XML Parser implementation
*       using PARSE XML statement for
*       datastructure 'EMPL'
*
* -----
*
DEFINE DATA PARAMETER
1 #XML_INPUT           (A) DYNAMIC BY VALUE
PARAMETER USING EMPL
*
LOCAL
1 #XML_PATH           (A) DYNAMIC
1 #XML_VALUE          (A) DYNAMIC
*
LOCAL
1 #CX                 (I4)
1 #CY                 (I4)
1 #CZ                 (I4)
END-DEFINE
*
* ----- INCLUDE THE PARSER
PARSE XML #XML_INPUT INTO PATH #XML_PATH VALUE #XML_VALUE
*
* DTD SYSEXXT EMPL
DECIDE ON FIRST #XML_PATH
  VALUE 'EMPLOYEE'
    RESET EMPLOYEE
  VALUE 'EMPLOYEE/@PERSONNEL-ID'
    /* #IMPLIED
    EMPLOYEE.PERSONNEL-ID := #XML_VALUE
  VALUE 'EMPLOYEE/FULL-NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/FIRST-NAME/$'
    EMPLOYEE.FIRST-NAME := #XML_VALUE
  VALUE 'EMPLOYEE/FULL-NAME/NAME'
    IGNORE
  VALUE 'EMPLOYEE/FULL-NAME/NAME/$'
    EMPLOYEE.NAME := #XML_VALUE
  VALUE 'EMPLOYEE/FULL-ADDRESS'
    IGNORE
  VALUE 'EMPLOYEE/FULL-ADDRESS/ADDRESS-LINE'
    /* optional multiple
    ADD 1 TO EMPLOYEE.C@ADDRESS-LINE

```

```
EXPAND ARRAY EMPLOYEE.ADDRESS-LINE TO
  (1:EMPLOYEE.C@ADDRESS-LINE)
VALUE 'EMPLOYEE/FULL-ADDRESS/ADDRESS-LINE/$'
  #CX := EMPLOYEE.C@ADDRESS-LINE
  EMPLOYEE.ADDRESS-LINE(#CX) := #XML_VALUE
VALUE 'EMPLOYEE/FULL-ADDRESS/CITY'
  IGNORE
VALUE 'EMPLOYEE/FULL-ADDRESS/CITY/$'
  EMPLOYEE.CITY := #XML_VALUE
VALUE 'EMPLOYEE/FULL-ADDRESS/ZIP'
  IGNORE
VALUE 'EMPLOYEE/FULL-ADDRESS/ZIP/$'
  EMPLOYEE.ZIP := #XML_VALUE
VALUE 'EMPLOYEE/FULL-ADDRESS/COUNTRY'
  IGNORE
VALUE 'EMPLOYEE/FULL-ADDRESS/COUNTRY/$'
  EMPLOYEE.COUNTRY := #XML_VALUE
VALUE 'EMPLOYEE/TELEPHONE'
  IGNORE
VALUE 'EMPLOYEE/TELEPHONE/AREA-CODE'
  IGNORE
VALUE 'EMPLOYEE/TELEPHONE/AREA-CODE/$'
  EMPLOYEE.AREA-CODE := #XML_VALUE
VALUE 'EMPLOYEE/TELEPHONE/PHONE'
  IGNORE
VALUE 'EMPLOYEE/TELEPHONE/PHONE/$'
  EMPLOYEE.PHONE := #XML_VALUE
VALUE 'EMPLOYEE/JOB-TITLE'
  IGNORE
VALUE 'EMPLOYEE/JOB-TITLE/$'
  EMPLOYEE.JOB-TITLE := #XML_VALUE
VALUE 'EMPLOYEE/INCOME'
  /* optional multiple
  ADD 1 TO EMPLOYEE.C@INCOME
  EXPAND ARRAY EMPLOYEE.INCOME TO
    (1:EMPLOYEE.C@INCOME)
VALUE 'EMPLOYEE/INCOME/SALARY'
  IGNORE
VALUE 'EMPLOYEE/INCOME/SALARY/$'
  #CX := EMPLOYEE.C@INCOME
  EMPLOYEE.SALARY(#CX) := #XML_VALUE
VALUE 'EMPLOYEE/INCOME/BONUS'
  /* optional multiple
  #CX := EMPLOYEE.C@INCOME
  ADD 1 TO EMPLOYEE.C@BONUS(#CX)
  EXPAND ARRAY EMPLOYEE.BONUS TO
    (*:*,1:EMPLOYEE.C@BONUS(#CX))
VALUE 'EMPLOYEE/INCOME/BONUS/$'
  #CX := EMPLOYEE.C@INCOME
  #CY := EMPLOYEE.C@BONUS(#CX)
  EMPLOYEE.BONUS(#CX,#CY) := #XML_VALUE
NONE
```



```

        IGNORE
    END-DECIDE
*
END-PARSE
*
END

```

サブルーチン **EMPL2S** の例：

```

* -----
* Generated from NATURAL XML TOOLKIT
*
*       'EMPL2S'
*
* DESCRIPTION
*           XML serialize implementation for
*           'EMPL' datastructure
* -----
*
DEFINE DATA PARAMETER
1 #XML_SERIALZE_OUTPUT (A) DYNAMIC
PARAMETER USING EMPL
LOCAL
1 #CX           (I4)
1 #CY           (I4)
1 #CZ           (I4)
END-DEFINE
*
#XML_SERIALZE_OUTPUT := '<?xml version="1.0" encoding="ISO-8859-1"?>'
*
* DTD SYSEXXT EMPL
COMPRESS #XML_SERIALZE_OUTPUT '<EMPLOYEE'
' PERSONNEL-ID="'EMPLOYEE.PERSONNEL-ID "'
'>' INTO #XML_SERIALZE_OUTPUT LEAVING NO
/* now the children
COMPRESS #XML_SERIALZE_OUTPUT '<FULL-NAME'
'>' INTO #XML_SERIALZE_OUTPUT LEAVING NO
/* now the children
COMPRESS #XML_SERIALZE_OUTPUT '<FIRST-NAME'
'>'
EMPLOYEE.FIRST-NAME
'</FIRST-NAME>' INTO #XML_SERIALZE_OUTPUT LEAVING NO
COMPRESS #XML_SERIALZE_OUTPUT '<NAME'
'>'
EMPLOYEE.NAME
'</NAME>' INTO #XML_SERIALZE_OUTPUT LEAVING NO
/*
COMPRESS #XML_SERIALZE_OUTPUT '</FULL-NAME>'
INTO #XML_SERIALZE_OUTPUT LEAVING NO
COMPRESS #XML_SERIALZE_OUTPUT '<FULL-ADDRESS'

```

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

```

        EMPLOYEE.BONUS(#CX,#CY)
        '</BONUS>' INTO #XML_SERIALZE_OUTPUT LEAVING NO
    END-FOR
/*
    COMPRESS #XML_SERIALZE_OUTPUT '</INCOME>'
    INTO #XML_SERIALZE_OUTPUT LEAVING NO
END-FOR
/*
COMPRESS #XML_SERIALZE_OUTPUT '</EMPLOYEE>'
INTO #XML_SERIALZE_OUTPUT LEAVING NO
END

```

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(xm1)
    ESCAPE BOTTOM
END-IF
/*
EXAMINE SUBSTRING(xm1,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:*)
    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:*)
    3 SALARY(A9)
    3 C@BONUS(I4)

```

```
3 BONUS(A9/1:*)  
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（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。これは必須です。空の Natural グループは許可されません。
- 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。
- 連続または選択が複数組み合わせられている場合（`<!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。これは、コンテキストまたは考えられる多様性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タイプ A253 の変数として生成されます。
- 複数のエレメントが常に次元 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、追加のカウントフィールド `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は、名前を一意にするために接尾辞で拡張されます。
- Natural 名に無効な特殊文字は、有効な Natural 名に変換されます。変換の設定については、XML ツールキットの [Natural XML Options] ダイアログボックス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 データ構造になる 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 )>

```

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 (Aℓ)	<!ELEMENT G1 (E1)> <!ELEMENT E1 (#PCDATA)>
1 G1 2 E1 (Aℓ) 2 E2 (Aℓ) 2 E3 (Aℓ)	<!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) 2 E1 (Aℓ/1:C@E1_MAX)	<!ELEMENT G1 (E1*)> <!ELEMENT E1 (#PCDATA)>
1 C@E1_MAX (I4) CONST <10> 1 G1 2 C@E1 (I4) 2 E1 (Aℓ/1:C@E1_MAX)	<!ELEMENT G1 (E1+)> <!ELEMENT E1 (#PCDATA)>
1 G1 2 E1 (Aℓ)	<!ELEMENT G1 (E1?)> <!ELEMENT E1 (#PCDATA)>
1 G1 2 E1 (Aℓ) 2 E2 (Aℓ) 2 E3 (Aℓ)	<!ELEMENT G1 (E1 E2 E3)> <!ELEMENT E1 (#PCDATA)> <!ELEMENT E2 (#PCDATA)> <!ELEMENT E3 (#PCDATA)>
1 G1 2 E1 (Aℓ) 2 E2 (Aℓ) 2 G2 2 E1_2 (Aℓ) 2 E3 (Aℓ)	<!ELEMENT G1 (E1, E2, G2)> <!ELEMENT E1 (#PCDATA)> <!ELEMENT E2 (#PCDATA)> <!ELEMENT G2 (E1, E3)> <!ELEMENT E3 (#PCDATA)>
1 #G1 2 #E1 (Aℓ)	<!ELEMENT G1 (E1)> <!ELEMENT E1 (#PCDATA)>
2 E1 (Aℓ) 3 ATTRIBUTES_OF_E1 4 A1 (Aℓ) CONST <'schema'>	<!ELEMENT E1 (#PCDATA)> <!ATTLIST E1 A1 #FIXED "schema" A2 NMTOKEN #IMPLIED

Natural	文書型定義
4 A2 (A!)	A3 ID #REQUIRED>
4 A3 (A!)	

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（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 名に無効な特殊文字は、有効な名前に変換されます。変換設定については、XML ツールキットの [\[Natural XML Options\] ダイアログボックス](#) 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は、次のオペランドをとります。

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 内容のタイプ： ? 処理の手順 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	見つかったエレメントの内容	オペランド 4
4	L	解析したデータが空の場合は 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 PARSE-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 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 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 PARSE_ERROR
IGNORE

```

例

```
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
```

157 解析エラーメッセージ

以下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	Parse ended without errors.	valid/*
-01	Wrong character set/Document does not start with '<'.	not-wf/sa/147.xml
-02	Processing instruction was not closed. Position %2%.	not-wf/sa/004.xml
-03	A CDATA section was not closed. Position %2%.	esi/001.xml
-04	!DOCTYPE section was not closed. Position %2%.	not-wf/sa/055.xml
-05	Incorrect syntax was used in a comment. Position %2%.	not-wf/sa/006.xml
-06	A comment was not closed. Position %2%.	not-wf/sa/027.xml
-07	A CDATA section was not closed. Position %2%	not-wf/sa/017.xml
-08	A comment section was not closed. Position %2%.	esi/002.xml
-09	Closing tag name was started with an invalid character. Position %2%.	not-wf/sa/019.xml
-10	Closing tag without starting element. Position %2%.	not-wf/sa/042.xml
-11	Closing tag '%3%' does not match the start tag '%1%'. Position %2%.	not-wf/sa/039.xml
-12	Closing tag was not closed. Position %2%.	esi/003.xml
-13	Closing tag '%1%' was not closed. Position %2%.	例はありません
-14	Starting tag name was started with an invalid character. Position %2%.	not-wf/sa/035.xml
-15	Attribute name of tag '%1%' not found. Position %2%.	例はありません
-16	Attribute name of tag '%1%' contains an invalid character. Position %2%.	not-wf/sa/001.xml
-17	Attribute value of tag '%1%' ending quotation mark missing. Position %2%.	not-wf/sa/013.xml
-18	Attribute value of tag '%1%' ending apostrophe missing. Position %2%.	esi/005.xml
-19	Starting tag section was not closed. Position %2%.	esi/006.xml
-20	Tag '%1%' section was not closed. Position %2%.	not-wf/sa/176.xml
-21	A section was not closed. Position %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	User defined error messages, parser ends.	例はありません
< -9000	User defined error messages, PARSER_ERROR is called and parser ends.	例はありません

索引

シンボル

概要

XML ツールキットプラグイン, 407

構成

DCOM

Web インターフェイス, 20

D

DCOM パラメータ

初期化ファイル

Web サーバーエクステンション, 63

H

H3-ANCHOR

SYSWEB, 199

SYSWEB3, 339

H3-ASCII-URL-TABLE

SYSWEB, 239

SYSWEB3, 379

H3-BUTTON

SYSWEB, 201

SYSWEB3, 341

H3-CHECKBOX-GROUP

SYSWEB, 203

SYSWEB3, 343

H3-CLOSE-FORM

SYSWEB, 215

SYSWEB3, 355

H3-CLOSE-HTML

SYSWEB, 217

SYSWEB3, 357

H3-CLOSE-LIST

SYSWEB, 219

SYSWEB3, 359

H3-COMMENT

SYSWEB, 205

SYSWEB3, 345

H3-HEADER

SYSWEB, 207

SYSWEB3, 347

H3-IMAGE

SYSWEB, 209

SYSWEB3, 349

H3-INPUT

SYSWEB, 211

SYSWEB3, 351

H3-LINE-BREAK

SYSWEB, 213

SYSWEB3, 353

H3-LINE_BREAK

SYSWEB, 213

SYSWEB3, 353

H3-LIST-ITEM

SYSWEB, 219

SYSWEB3, 359

H3-OPEN-FORM

SYSWEB, 215

SYSWEB3, 355

H3-OPEN-HTML

SYSWEB, 217

SYSWEB3, 357

H3-OPEN-HTML-JAVASCRIPT

SYSWEB, 217

SYSWEB3, 357

H3-OPEN-LIST

SYSWEB, 219

SYSWEB3, 359

H3-PARAGRAPH

SYSWEB, 223

SYSWEB3, 363

H3-RADIO-GROUP

SYSWEB, 225

SYSWEB3, 365

H3-RULE

SYSWEB, 227

SYSWEB3, 367

H3-SCROLLING-LIST

SYSWEB, 229

SYSWEB3, 369

H3-TABLE

SYSWEB, 231

SYSWEB3, 371

H3-TABLE-COLOR

SYSWEB, 231

SYSWEB3, 371

H3-TAG

SYSWEB, 235

SYSWEB3, 375

H3-TEXT-AREA

SYSWEB, 237

SYSWEB3, 377

H3-TEXT-TO-URL

SYSWEB, 239

SYSWEB3, 379

H3-TIME-DATE

SYSWEB, 241
SYSWEB3, 381
H3-TIME_DATE
SYSWEB, 241
SYSWEB3, 381
HTML エクステンション
SYSWEB
 オンラインドキュメント, 138
SYSWEB3
 オンラインドキュメント, 267
HTTP サーバー変数
 初期化ファイル
 Web サーバーエクステンション, 67

J

JavaScript
 デモアプリケーション
 Web インターフェイス, 101

N

NAT-DATA
 SYSWEB3, 389
NAT-DIR
 SYSWEB, 249
 SYSWEB3, 391
NAT-DIRR
 SYSWEB3, 393
NAT-DOCU
 SYSWEB, 259
 SYSWEB3, 403
NAT-ENV
 SYSWEB, 251
 SYSWEB3, 395
NAT-HTML
 SYSWEB, 253
 SYSWEB3, 397
NAT-INFO
 SYSWEB, 255
 SYSWEB3, 399
NAT-LIB
 SYSWEB, 243
 SYSWEB3, 383
NAT-LIST
 SYSWEB, 257
 SYSWEB3, 401
NAT-RES
 SYSWEB, 261
 SYSWEB3, 405
Natural データエリアの選択
 XML ツールキット, 432

P

PAL パラメータ
 初期化ファイル
 Web サーバーエクステンション, 63

R

RPC パラメータ
 初期化ファイル
 Web サーバーエクステンション, 62

RPC パラメータ
 初期化ファイル
 Web サーバーエクステンション, 66

S

SYSWEB
 H3-ANCHOR, 199
 H3-ASCII-URL-TABLE, 239
 H3-BUTTON, 201
 H3-CHECKBOX-GROUP, 203
 H3-CLOSE-FORM, 215
 H3-CLOSE-HTML, 217
 H3-CLOSE-LIST, 219
 H3-COMMENT, 205
 H3-HEADER, 207
 H3-IMAGE, 209
 H3-INPUT, 211
 H3-LINE-BREAK, 213
 H3-LINE_BREAK, 213
 H3-LIST-ITEM, 219
 H3-OPEN-FORM, 215
 H3-OPEN-HTML, 217
 H3-OPEN-HTML-JAVASCRIPT, 217
 H3-OPEN-LIST, 219
 H3-PARAGRAPH, 223
 H3-RADIO-GROUP, 225
 H3-RULE, 227
 H3-SCROLLING-LIST, 229
 H3-TABLE, 231
 H3-TABLE-COLOR, 231
 H3-TAG, 235
 H3-TEXT-AREA, 237
 H3-TEXT-TO-URL, 239
 H3-TIME-DATE, 241
 H3-TIME_DATE, 241
 NAT-DIR, 249
 NAT-DOCU, 259
 NAT-ENV, 251
 NAT-HTML, 253
 NAT-INFO, 255
 NAT-LIB, 243
 NAT-LIST, 257
 NAT-RES, 261
 W3-ASCII-HTML-TABLE, 173
 W3-ASCII-URL-TABLE, 177
 W3-ASCII-XML-TABLE, 175
 W3APPLY-XSLT-RESOURCE, 193
 W3APPLY-XSLT-XML, 191
 W3APPLY-XSLT-XML-TO-DYNAMIC, 195
 W3CHECK-RESOURCE, 187
 W3CLEAR, 143
 W3CONTENT-TYPE, 145
 W3COUNT, 147
 W3DELETE-RESOURCE, 189
 W3END, 157
 W3ERROR, 149
 W3ERROR-TEMPLATE, 149
 W3ERROR-TEMPLATE-XML, 149
 W3ERROR-TEXT, 149
 W3HTML, 151
 W3HTMLARRAY, 151
 W3HTMLDYNAMIC, 151
 W3HTMLLINEDYNAMIC, 151

- W3HTMLLINE, 151
- W3HTTP, 153
- W3HTTP-HEADER, 153
- W3HTTPARRAY, 153
- W3HTTPDYNAMIC, 153
- W3INFO, 155
- W3INIT, 157
- W3LIST-ENVIRONMENT, 159
- W3LIST-ENVIRONMENT-TO-DYNAMIC, 159
- W3LOAD-RESOURCE, 181
- W3LOCATION, 161
- W3NEWLINE, 171
- W3READ-ENVIRONMENT, 163
- W3READ-ENVIRONMENT-ARRAY, 163
- W3READ-ENVIRONMENT-GROUP, 165
- W3READ-ENVIRONMENT-TEXTAREA, 167
- W3READ-ENVIRONMENT-TO-DYNAMIC, 163
- W3READ-OUTPUT, 197
- W3READ-RESOURCE, 183
- W3REPLACE, 179
- W3TEXT, 169
- W3TEXT-TO-HTML, 173
- W3TEXT-TO-URL, 177
- W3TEXT-TO-XML, 175
- W3TEXTARRAY, 169
- W3TEXTDYNAMIC, 169
- W3TEXTLINE, 169
- W3TEXTLINEDYNAMIC, 169
- W3WRITE-RESOURCE, 185
- WEB-ONL, 245
- WEB-WIZ, 247
- インストール
 - Web サーバーエクステンション, 54
- オンラインドキュメント, 135
- プログラムの呼び出し
 - Web サーバーエクステンション, 56
- テストユーティリティ WEB-ONL
 - Web サーバーエクステンション, 77
- SYSWEB から SYSWEB3
 - 移行
 - Web インターフェイス, 107
- SYSWEB3
 - H3-ANCHOR, 339
 - H3-ASCII-URL-TABLE, 379
 - H3-BUTTON, 341
 - H3-CHECKBOX-GROUP, 343
 - H3-CLOSE-FORM, 355
 - H3-CLOSE-HTML, 357
 - H3-CLOSE-LIST, 359
 - H3-COMMENT, 345
 - H3-HEADER, 347
 - H3-IMAGE, 349
 - H3-INPUT, 351
 - H3-LINE-BREAK, 353
 - H3-LINE_BREAK, 353
 - H3-LIST-ITEM, 359
 - H3-OPEN-FORM, 355
 - H3-OPEN-HTML, 357
 - H3-OPEN-HTML-JAVASCRIPT, 357
 - H3-OPEN-LIST, 359
 - H3-PARAGRAPH, 363
 - H3-RADIO-GROUP, 365
 - H3-RULE, 367
 - H3-SCROLLING-LIST, 369
 - H3-TABLE, 371
 - H3-TABLE-COLOR, 371
 - H3-TAG, 375
 - H3-TEXT-AREA, 377
 - H3-TEXT-TO-URL, 379
 - H3-TIME-DATE, 381
 - H3-TIME_DATE, 381
 - NAT-DATA, 389
 - NAT-DIR, 391
 - NAT-DIRR, 393
 - NAT-DOCU, 403
 - NAT-ENV, 395
 - NAT-HTML, 397
 - NAT-INFO, 399
 - NAT-LIB, 383
 - NAT-LIST, 401
 - NAT-RES, 405
 - W3-ASCII-HTML-TABLE, 309
 - W3-ASCII-URL-TABLE, 313
 - W3-ASCII-XML-TABLE, 311
 - W3APPLY-XSLT-RESOURCE, 329
 - W3APPLY-XSLT-XML, 327
 - W3APPLY-XSLT-XML-TO-DYNAMIC, 331
 - W3BINARY, 271
 - W3CHECK-RESOURCE, 323
 - W3CLEAR, 273
 - W3CONTENT-TYPE, 275
 - W3COUNT, 277
 - W3DELETE-RESOURCE, 325
 - W3END, 287
 - W3ERROR, 279
 - W3ERROR-TEMPLATE, 279
 - W3ERROR-TEMPLATE-XML, 279
 - W3ERROR-TEXT, 279
 - W3HTML, 281
 - W3HTMLARRAY, 281
 - W3HTMLDYNAMIC, 281
 - W3HTMLLINEDYNAMIC, 281
 - W3HTMLLINE, 281
 - W3HTTP, 283
 - W3HTTP-HEADER, 283
 - W3HTTPARRAY, 283
 - W3HTTPDYNAMIC, 283
 - W3INFO, 285
 - W3INIT, 287
 - W3LIMIT, 277
 - W3LIST-ENVIRONMENT, 289
 - W3LIST-ENVIRONMENT-TO-DYNAMIC, 289
 - W3LIST-RESOURCE, 333
 - W3LOAD-RESOURCE, 317
 - W3LOCATION, 295
 - W3MIME-DATA, 291
 - W3MIME-TYPE, 293
 - W3NEWLINE, 305
 - W3READ-ENVIRONMENT, 297
 - W3READ-ENVIRONMENT-ARRAY, 297
 - W3READ-ENVIRONMENT-GROUP, 299
 - W3READ-ENVIRONMENT-TEXTAREA, 301
 - W3READ-ENVIRONMENT-TEXTAREA-DYN, 301
 - W3READ-ENVIRONMENT-TO-DYNAMIC, 297
 - W3READ-INPUT, 335
 - W3READ-OUTPUT, 337
 - W3READ-RESOURCE, 319
 - W3REPLACE, 315

W3SOURCE-TO-XARRAY, 307
W3TEXT, 303
W3TEXT-TO-HTML, 309
W3TEXT-TO-URL, 313
W3TEXT-TO-XML, 311
W3TEXTARRAY, 303
W3TEXTDYNAMIC, 303
W3TEXTLINE, 303
W3TEXTLINEDYNAMIC, 303
W3WRITE-RESOURCE, 321
WEB-ONL, 385
WEB-WIZ, 387
インストール
 Web サーバーエクステンション, 57, 58
オンラインドキュメント, 263
プログラムの呼び出し
 Web サーバーエクステンション, 60
テストユーティリティ WEB-ONL
 Web サーバーエクステンション, 83

U

URL デコードのための英数字変換
管理
 Web インターフェイス, 96

V

View
 プラグイン
 Web インターフェイス, 127

W

W3-ASCII-HTML-TABLE
 SYSWEB, 173
 SYSWEB3, 309
W3-ASCII-URL-TABLE
 SYSWEB, 177
 SYSWEB3, 313
W3-ASCII-XML-TABLE
 SYSWEB, 175
 SYSWEB3, 311
W3APPLY-XSLT-RESOURCE
 SYSWEB, 193
 SYSWEB3, 329
W3APPLY-XSLT-XML
 SYSWEB, 191
 SYSWEB3, 327
W3APPLY-XSLT-XML-TO-DYNAMIC
 SYSWEB, 195
 SYSWEB3, 331
W3BINARY
 SYSWEB3, 271
W3CHECK-RESOURCE
 SYSWEB, 187
 SYSWEB3, 323
W3CLEAR
 SYSWEB, 143
 SYSWEB3, 273
W3CONTENT-TYPE
 SYSWEB, 145
 SYSWEB3, 275
W3COUNT

 SYSWEB, 147
 SYSWEB3, 277
W3DELETE-RESOURCE
 SYSWEB, 189
 SYSWEB3, 325
W3END
 SYSWEB, 157
 SYSWEB3, 287
W3ERROR
 SYSWEB, 149
 SYSWEB3, 279
W3ERROR-TEMPLATE
 SYSWEB, 149
 SYSWEB3, 279
W3ERROR-TEMPLATE-XML
 SYSWEB, 149
 SYSWEB3, 279
W3ERROR-TEXT
 SYSWEB, 149
 SYSWEB3, 279
W3HTML
 SYSWEB, 151
 SYSWEB3, 281
W3HTMLARRAY
 SYSWEB, 151
 SYSWEB3, 281
W3HTMLDYNAMIC
 SYSWEB, 151
 SYSWEB3, 281
W3HTMLINEDYNAMIC
 SYSWEB, 151
 SYSWEB3, 281
W3HTMMLINE
 SYSWEB, 151
 SYSWEB3, 281
W3HTTP
 SYSWEB, 153
 SYSWEB3, 283
W3HTTP-HEADER
 SYSWEB, 153
 SYSWEB3, 283
W3HTTPARRAY
 SYSWEB, 153
 SYSWEB3, 283
W3HTTPDYNAMIC
 SYSWEB, 153
 SYSWEB3, 283
W3INFO
 SYSWEB, 155
 SYSWEB3, 285
W3INIT
 SYSWEB, 157
 SYSWEB3, 287
W3LIMIT
 SYSWEB3, 277
W3LIST-ENVIRONMENT
 SYSWEB, 159
 SYSWEB3, 289
W3LIST-ENVIRONMENT-TO-DYNAMIC
 SYSWEB, 159
 SYSWEB3, 289
W3LIST-RESOURCE
 SYSWEB3, 333
W3LOAD-RESOURCE

- SYSWEB, 181
- SYSWEB3, 317
- W3LOCATION
 - SYSWEB, 161
 - SYSWEB3, 295
- W3MIME-DATA
 - SYSWEB3, 291
- W3MIME-TYPE
 - SYSWEB3, 293
- W3NEWLINE
 - SYSWEB, 171
 - SYSWEB3, 305
- W3READ-ENVIRONMENT
 - SYSWEB, 163
 - SYSWEB3, 297
- W3READ-ENVIRONMENT-ARRAY
 - SYSWEB, 163
 - SYSWEB3, 297
- W3READ-ENVIRONMENT-GROUP
 - SYSWEB, 165
 - SYSWEB3, 299
- W3READ-ENVIRONMENT-TEXTAREA
 - SYSWEB, 167
 - SYSWEB3, 301
- W3READ-ENVIRONMENT-TEXTAREA-DYN
 - SYSWEB3, 301
- W3READ-ENVIRONMENT-TO-DYNAMIC
 - SYSWEB, 163
 - SYSWEB3, 297
- W3READ-INPUT
 - SYSWEB3, 335
- W3READ-OUTPUT
 - SYSWEB, 197
 - SYSWEB3, 337
- W3READ-RESOURCE
 - SYSWEB, 183
 - SYSWEB3, 319
- W3REPLACE
 - SYSWEB, 179
 - SYSWEB3, 315
- W3SOURCE-TO-XARRAY
 - SYSWEB3, 307
- W3TEXT
 - SYSWEB, 169
 - SYSWEB3, 303
- W3TEXT-TO-HTML
 - SYSWEB, 173
 - SYSWEB3, 309
- W3TEXT-TO-URL
 - SYSWEB, 177
 - SYSWEB3, 313
- W3TEXT-TO-XML
 - SYSWEB, 175
 - SYSWEB3, 311
- W3TEXTARRAY
 - SYSWEB, 169
 - SYSWEB3, 303
- W3TEXTDYNAMIC
 - SYSWEB, 169
 - SYSWEB3, 303
- W3TEXTLINE
 - SYSWEB, 169
 - SYSWEB3, 303
- W3TEXTLINEDYNAMIC
 - SYSWEB, 169
 - SYSWEB3, 303
- W3WRITE-RESOURCE
 - SYSWEB, 185
 - SYSWEB3, 321
- Web インターフェイス
 - HTTP サーバー, 18
 - RPC
 - サーバー, 18
 - Web サーバーエクステンション
 - SYSWEB, 53
 - 構成, 20
 - アーキテクチャ, 6
 - 移行
 - SYSWEB から SYSWEB3, 107
 - エラーメッセージ, 105
 - 管理, 93
 - URL デコードのための英数字変換, 96
 - 英数字を HTML に変換, 96
 - ユーザー定義エラーページ XHTML の作成, 95
 - ユーザー定義エラーページの作成, 95
 - リターンページトランスポートバッファのサイズの設
定, 94
 - 概要, 3
 - 機能, 10, 11
 - 構成, 15
 - サブプログラム
 - 構築, 35
 - テスト, 47
 - 使用, 31
 - セキュリティ, 12
 - 設定
 - 前提条件, 33
 - デモアプリケーション, 97
 - JavaScript, 101
 - トラブルシューティング
 - メッセージ, 27
 - はじめに, 5
 - プラグイン, 109
 - View, 127
 - オプション, 125
 - オンラインテストユーティリティ, 130
 - 機能, 112
 - クラスの生成, 128
 - コマンド, 110
 - 置換文字列の挿入, 122
 - 変換プログラムの使用, 112
 - プログラミング
 - ヒント, 46
 - プログラミングのヒント, 89
 - 命名規則, 49
 - モジュール, 9
 - Web サーバーエクステンション
 - SYSWEB
 - Web インターフェイス, 53
 - インストール
 - SYSWEB, 54
 - SYSWEB3, 57, 58
 - エラーメッセージ, 71
 - 初期化ファイル, 61
 - DCOM パラメータ, 63
 - HTTP サーバー変数, 67
 - PAL パラメータ, 63
 - RPC パラメータ, 62

- エラーテンプレート, 67
- 設定, 63
- プログラムの呼び出し
 - SYSWEB, 56
 - SYSWEB3, 60
- Web インターフェイス
 - DCOM
 - サーバー, 20
- Web サーバーエクステンション
 - テストユーティリティ WEB-ONL
 - SYSWEB, 77
 - SYSWEB3, 83
 - 初期化ファイル
 - データ転送設定, 66
- WEB-ONL
 - SYSWEB, 245
 - SYSWEB3, 385
- WEB-WIZ
 - SYSWEB, 247
 - SYSWEB3, 387

X

- XML ツールキット
 - Natural DTD パーサー, 478
 - エラーメッセージ, 485
 - 機能, 410
 - 使用, 419
 - シンプルな XML パーサー, 455
 - 制限, 463
 - 説明, 456
 - 制限, 414
 - 生成オプション, 423
 - 名前の置換, 424
 - パス設定, 427
 - 生成された Natural データエリア, 476
 - 生成されたタイプ定義, 479
 - 特徴, 410
 - はじめに, 409
 - パーサー CALLBACK コピーコード, 480
 - 呼び出し, 420
 - 例
 - コピーコードのシリアライズ, 466
 - サブルーチンのシリアライズ, 468
- XML ドキュメントの生成
 - Natural データソースから, 431
 - Natural データエリアの選択, 432
 - シリアライズの生成, 437
 - 生成レポート, 440
 - パラメータの設定, 439
 - パーサーの生成, 438
 - ルートグループ, 440
 - 外部データソースから, 443
 - Natural データエリアの生成, 448
 - XML パーサーコールバックのためのサブプログラムの生成, 453
 - 再帰レベル, 446
 - 生成結果, 453
- XML ツールキット
 - 使用
 - ヘルプ, 421
- XML ツールキットプラグイン
 - 概要, 407
- XML ドキュメントの生成

- Natural データソースから
 - DTD 定義または XML スキーマでのファイルの生成, 435
 - データタイプの選択, 434
- 外部データソースから
 - DTD または XML スキーマからの生成, 444
 - XML パーサーコールバックのためのコピーコードの生成, 451
 - シリアライズのためのコピーコードの生成, 449
 - ルートエレメントまたは文書型, 446

あ

- アーキテクチャ
 - Web インターフェイス, 6

え

- 英数字を HTML に変換
 - 管理
 - Web インターフェイス, 96
- エラーテンプレート
 - 初期化ファイル
 - Web サーバーエクステンション, 67
- エラーメッセージ
 - Web インターフェイス, 105
 - Web サーバーエクステンション, 71
 - XML ツールキット, 485

お

- オプション
 - プラグイン
 - Web インターフェイス, 125
- オンラインテストユーティリティ
 - プラグイン
 - Web インターフェイス, 130
- オンラインドキュメント
 - SYSWEB, 135
 - HTML エクステンション, 138
 - 基本モジュール, 136
 - デモアプリケーション, 141
 - ユーティリティ, 140
 - SYSWEB3, 263
 - HTML エクステンション, 267
 - 基本モジュール, 264
 - デモアプリケーション, 269
 - ユーティリティ, 268
- オンラインドキュメント
 - SYSWEB
 - テンプレート/XSLT 処理, 138
 - SYSWEB3
 - テンプレート/XSLT 処理, 266

か

- 管理
 - Web インターフェイス, 93

き

- 機能
 - XML ツールキット, 410

- プラグイン
 - Web インターフェイス, 112
- 基本モジュール
 - SYSWEB
 - オンラインドキュメント, 136
 - SYSWEB3
 - オンラインドキュメント, 264

く

- クラスの生成
 - プラグイン
 - Web インターフェイス, 128

こ

- 構成
 - Web インターフェイス, 15
 - Web サーバーエクステンション
 - Web インターフェイス, 20
- 構築
 - サブプログラム
 - Web インターフェイス, 35
- コマンド
 - プラグイン
 - Web インターフェイス, 110

し

- 初期化ファイル
 - Web サーバーエクステンション, 61
- 使用
 - Web インターフェイス, 31
 - XML ツールキット, 419

せ

- 制限
 - XML ツールキット, 414
- 生成オプション
 - XML ツールキット, 423
- セキュリティ
 - Web インターフェイス, 12
- 設定
 - 初期化ファイル
 - Web サーバーエクステンション, 63
- 前提条件
 - Web インターフェイス, 33

ち

- 置換文字列の挿入
 - プラグイン
 - Web インターフェイス, 122

て

- テスト
 - サブプログラム
 - Web インターフェイス, 47
- デモアプリケーション
 - SYSWEB
 - オンラインドキュメント, 141

- SYSWEB3
 - オンラインドキュメント, 269
- Web インターフェイス, 97
 - テンプレート/XSLT 処理
 - SYSWEB
 - オンラインドキュメント, 138
 - SYSWEB3
 - オンラインドキュメント, 266

と

- 特徴
 - XML ツールキット, 410

な

- 名前の置換
 - 生成オプション
 - XML ツールキット, 424

は

- はじめに
 - Web インターフェイス, 5
 - XML ツールキット, 409
- パス設定
 - 生成オプション
 - XML ツールキット, 427

ひ

- ヒント
 - プログラミング
 - Web インターフェイス, 46

ふ

- プラグイン
 - Web インターフェイス, 109
- プログラミングのヒント
 - Web インターフェイス, 89

へ

- 変換プログラムの使用
 - プラグイン
 - Web インターフェイス, 112
- ヘルプ
 - 使用
 - XML ツールキット, 421

め

- 命名規則
 - Web インターフェイス, 49
- メッセージ
 - トラブルシューティング
 - Web インターフェイス, 27

ゆ

- ユーザー定義エラーページ XHTML の作成
 - 管理

- Web インターフェイス, 95
- ユーザー定義エラーページの作成
管理
 - Web インターフェイス, 95
- ユーティリティ
 - SYSWEB
 - オンラインドキュメント, 140
 - SYSWEB3
 - オンラインドキュメント, 268

よ

- 呼び出し
 - XML ツールキット, 420

り

- リターンページトランスポートバッファのサイズの設定
管理
 - Web インターフェイス, 94