４．４　トランスレータ設定

ここでは、XML トランスレータのインストール方法、環境設定情報、DOS コマンドの使用方法、そして変換 DLL の使用方法について説明します。

ウィンドウアプリケーションの使い方に関しては、「３．６　トランスレータ設定／変換作業」をご覧ください。

・プログラム導入 …… 4.4.1 参照

・使用環境設定 …… 4.4.2 参照

・コマンドラインからの変換実行 …… 4.4.3 参照

・変換 DLL API 仕様 …… 4.4.4 参照

４．４．１　プログラム導入

XML トランスレータのインストーラは、自己解凍形式の実行ファイルとして、WEB サイトより提供されます(この実行ファイルのファイル名は、WEB サイトから確認してください)。

自己解凍ファイルを実行すると、解凍先に指定したフォルダ中に、インストールに必要なファイル群が生成されます。この中から"setup.exe"を実行するとインストールが開始されます。

インストールには、すべてをインストールするタイプと、変換 DLL だけをインストールするタイプの 2 つがあり、インストール中に選択することができます(変換 DLL だけをインストールするようにすると、ウィンドウアプリケーション、XSL ファイルがインストールされません)。

・パッケージ内容 …… (1)参照

・インストール手順 …… (2)参照

・アンインストール手順 …… (3)参照

（A） 自己解凍ファイル

XML トランスレータのインストーラは、自己解凍形式の実行ファイルとして、WEB サイトより提供されます。この実行ファイルのファイル名は、WEB サイトから確認してください。

（B） 自己解凍ファイルの展開結果

inst32i.ex
_isdel.exe
_setup.dll
_sys1.cab
_user1.cab
Data.tag
data1.cab
lang.dat
layout.bin
os.dat
Setup.exe
Setup.ini
setup.ins
setup.lid

（C） インストールされるファイル（すべてをインストールした場合）

（a） 実行ファイル(XML トランスレータをインストールしたフォルダに格納されます)

xmtr.exe, sjacxtr.exe

（b） DLL 等(Windows のシステムディレクトリに格納されます)

sjacxtr.dll, erces-c_1_0.dll, icuuc.dll
Msvcirt.dll, Msvcrt.dll, Icu フォルダ
Asycfilt.dll, Cmdlgjp.dll, Comctl32.dll
Stdole2.tlb, Vb6jp.dll, Comcat.dll
Comdlg32.ocx, Msvbvm60.dll, Oleaut32.dll
Olepro32.dll, Scrrun.dll

（c） 環境設定ファイル(XML トランスレータをインストールしたフォルダに格納されます)

env.xml, codes.xml

（d） DTD ファイル(XML トランスレータをインストールしたフォルダの下の DTD フォルダに格納されます)

*.dtd

（e） XSL ファイル(XML トランスレータをインストールしたディレクトリの下の XSL ディレクトリに格納されます)

*.xsl

（f） Readme ファイル(XML トランスレータをインストールしたディレクトリに格納されます)

Readme ファイルには、変換 DLL の使い方および注意事項が記載されています。

Readme.txt

（D） インストールされるファイル（変換 DLL だけをインストールした場合）

（a） 実行ファイル(XML トランスレータをインストールしたフォルダに格納されます)

xmtr.exe, sjacxtr.exe

（b） DLL 等(Windows のシステムディレクトリに格納されます)

sjacxtr.dll, xerces-c_1_0.dll, icuuc.dll
Msvcirt.dll, Msvcrt.dll, Icu フォルダ

（c） DTD ファイル(XML トランスレータをインストールしたフォルダの下の DTD フォルダに格納されます)

*.dtd

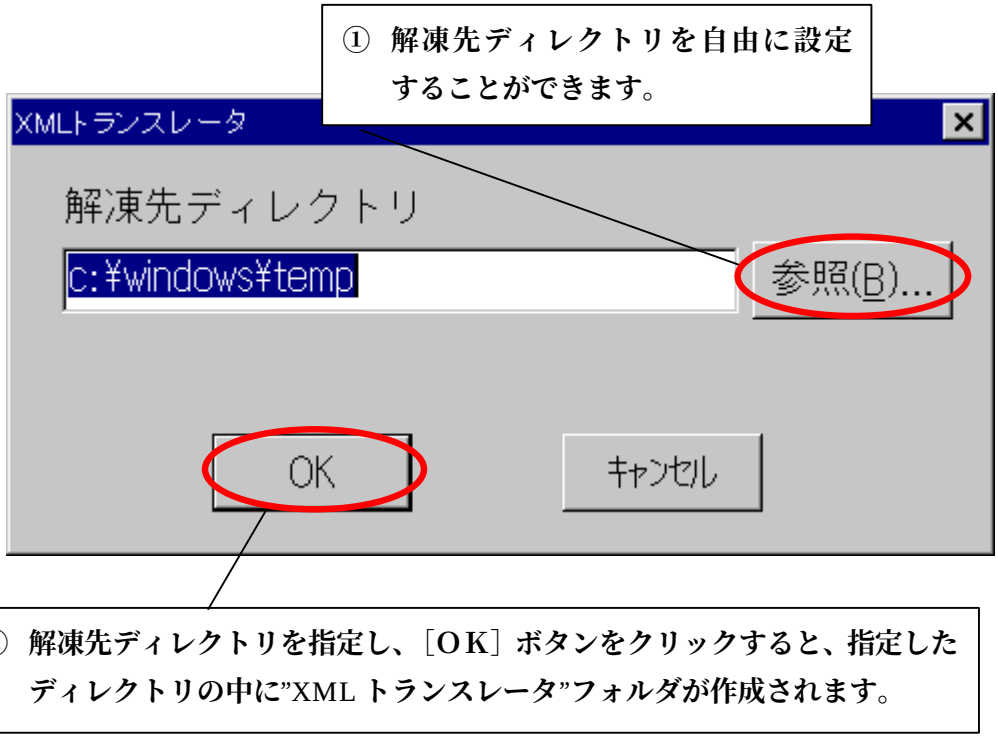
（d） Readme ファイル(XML トランスレータをインストールしたディレクトリに格納されます)

Readme ファイルには、変換 DLL の使い方および注意事項が記載されています。

Readme.txt

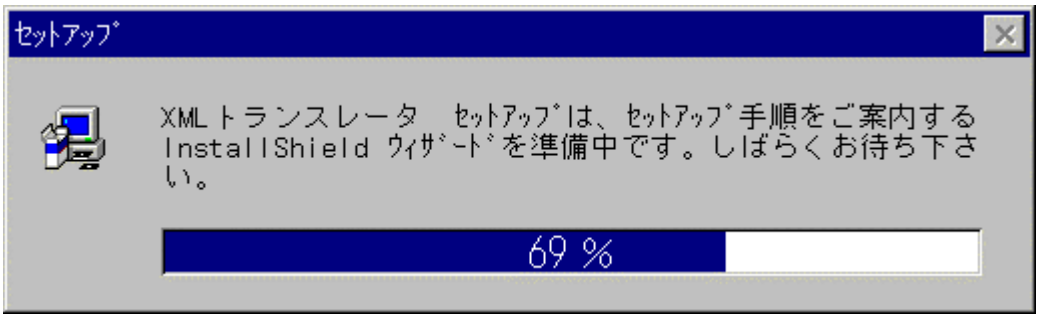
（A） XNL トランスレータを公開している WEB サイトより、自己解凍形式の実行ファイルをダウンロードします(ダウンロード方法および実行ファイルのファイル名に関しては、WEB サイトの管理者にご確認ください)。

（B） ダウンロードした実行ファイルをダブルクリックすると、「解凍先ディレクトリ選択画面」が表示されます。

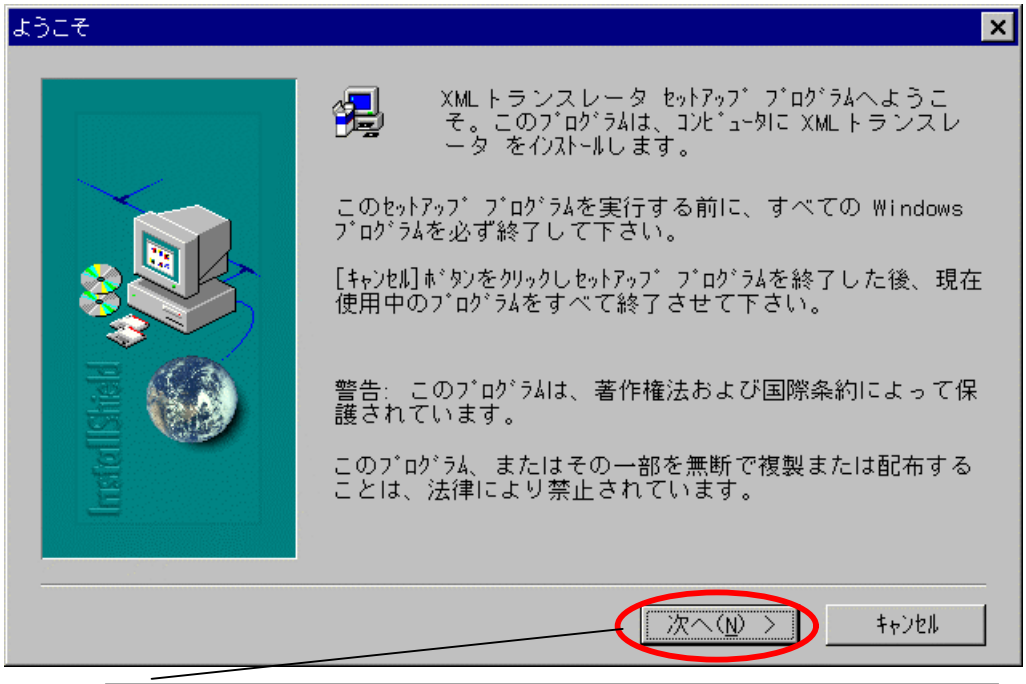


② 解凍先ディレクトリを指定し、［OK］ボタンをクリックすると、指定したディレクトリの中に”XML トランスレータ”フォルダが作成されます。

（C） ”XML トランスレータ”フォルダ中の Setup.exe をダブルクリックすると、インストールが始まります。



（D） インストール開始画面

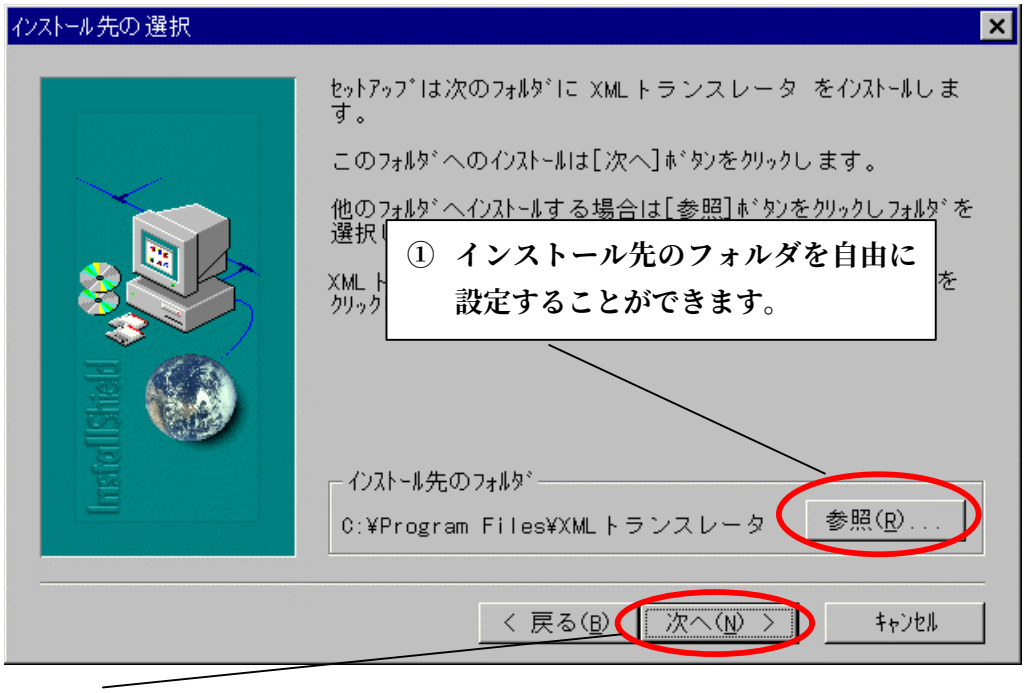


① [次へ]ボタンをクリックし、インストールを進めます。

<補足説明>

[キャンセル]ボタンをクリックすれば、いつでもインストール作業を中止することができます。

（E） インストール先を選択します。

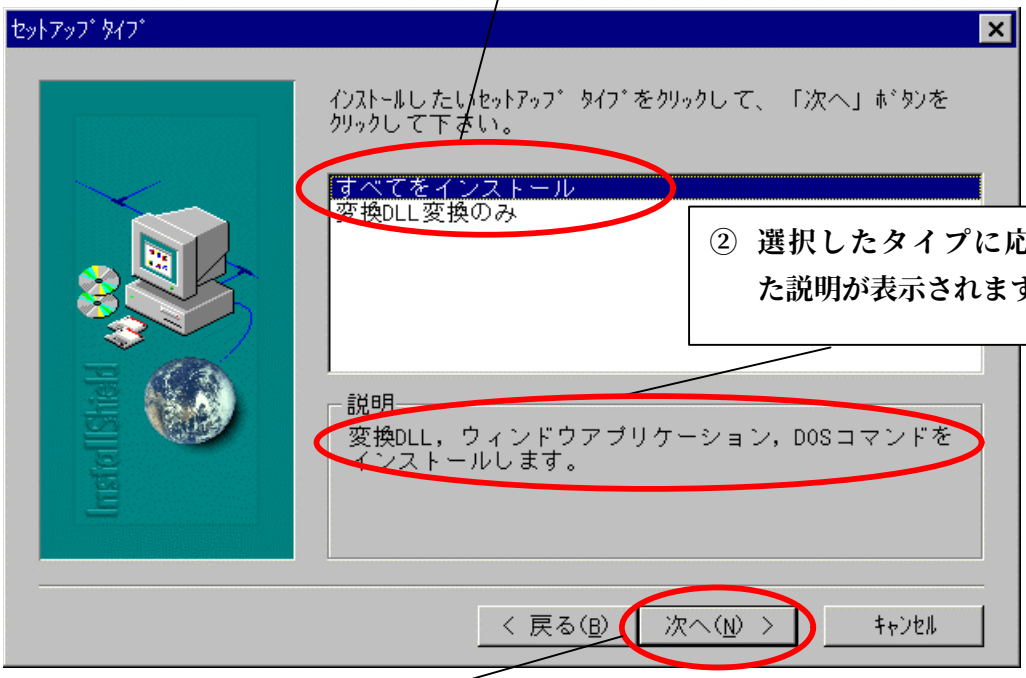


②インストール先のフォルダを決定したら、[次へ]ボタンをクリックし、インストールを進めます。

<補足説明>
[戻る]ボタンをクリックすれば、いつでも１つ前のインストール作業からやり直すことができます。

（F） セットアップタイプを選択します。

① セットアップタイプを選択します。

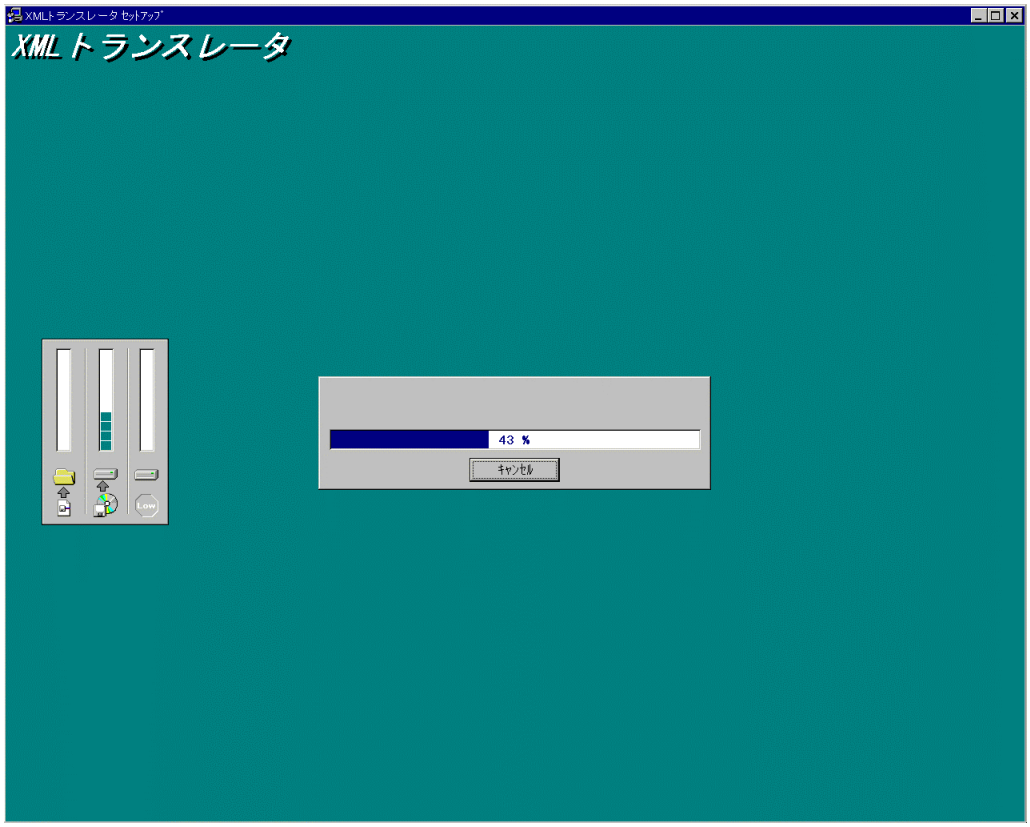


② 選択したタイプに応じた説明が表示されます。

③ セットアップタイプを決定したら、[次へ]ボタンをクリックし、インストールを進めます。

<注意>
セットアップタイプの選択画面で、[次へ]ボタンをクリックすると、ファイルの転送が開始されます。

（G） ファイルの転送が行われます。



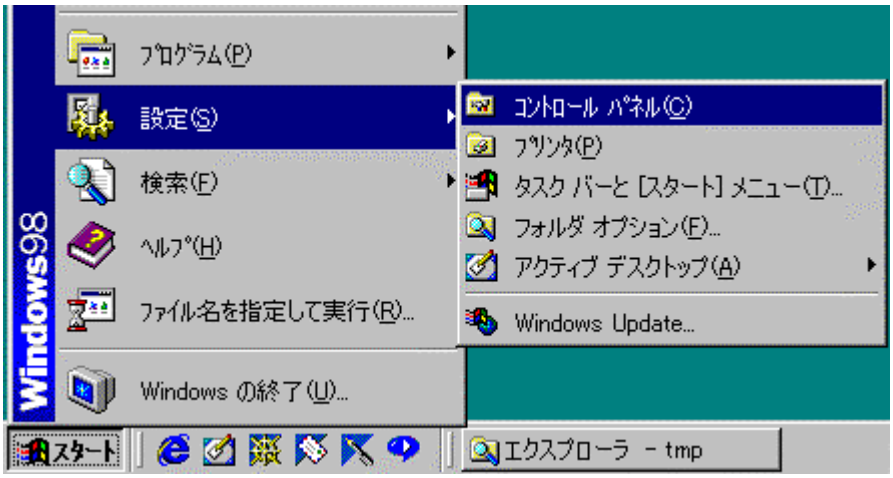
（H） インストールを終了します。

① [完了]ボタンをクリックし、インストールを終了します。

<補足説明>
インストール終了後、最初に圧縮ファイルを解凍してできた”XML トランスレータ”フォルダ及び、その中のファイルは不要ですので削除して下さい。

<補足説明>
インストールを行うときの Windows の状態によっては､インストール終了時に｢Winsows の再起動｣を求められることがあります。この場合、使用中のアプリケーションをすべて終了させてから、Windows を再起動してください。

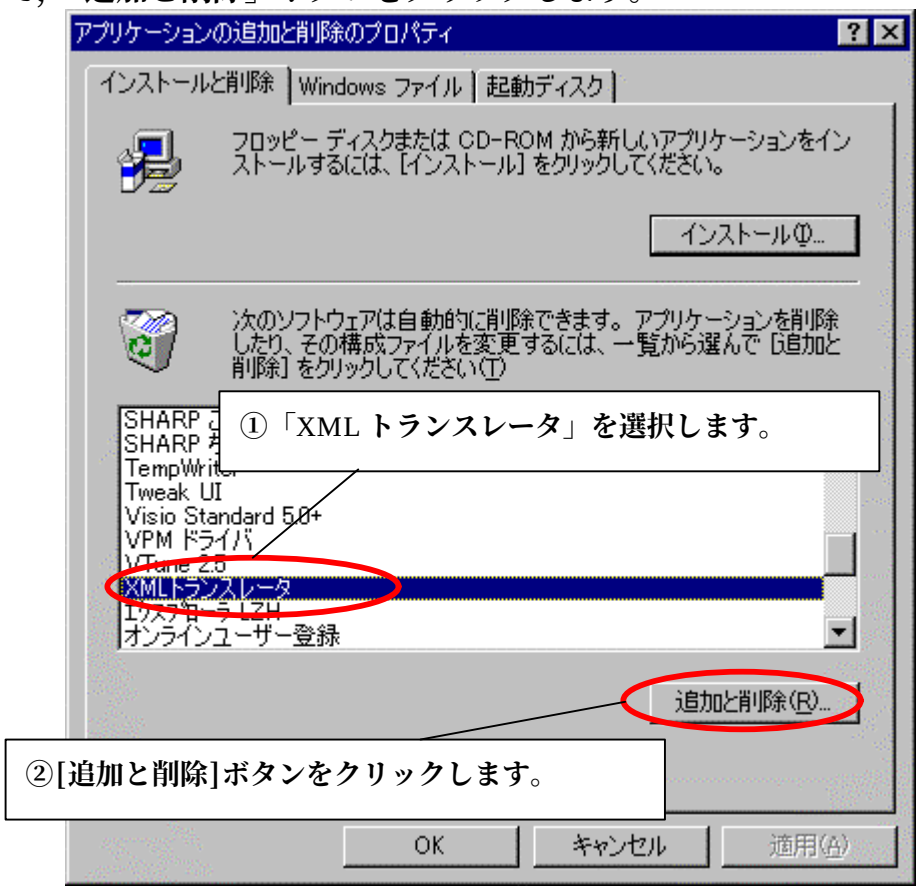
（A） 「スタート」メニューより、「設定」→「コントロールパネル」を実行して、コントロールパネルを開きます。



（B） コントロールパネルより、「アプリケーションの追加と削除」をダブルクリックします。

① 「アプリケーションの追加と削除」を実行します。

（C）「セットアップと削除」の設定タブより，「XML トランスレータ」の項目を選択して，「追加と削除」ボタンをクリックします。



<補足説明>

XML トランスレータをインストールしたフォルダ、およびそのサブフォルダの中に、インストールによって作成したファイル以外のファイルがある場合、又は DTD ファイルをアップデートした場合、そのファイルは削除されません。独自に作成したファイルが存在する場合は、アンインストール後に、手動で XML トランスレータをインストールしたフォルダを削除してください。

４．４．２　使用環境設定

ここでは、XML トランスレータが使用するファイルの内容および格納場所について説明します。

・ファイル内容 …… (1)参照

（A） XML トランスレータで使用するファイル一覧

以下の表に、XML トランスレータで使用するファイルの一覧を示します。なお、表中の「格納位置」には、各ファイルが格納されるフォルダ名が記述されています(%XMLTR% は XML トランスレータをインストールしたディレクトリを表し、空欄はユーザが作成するファイルであることを表します)。

ファイル種別	格納位置	ファイル名／拡張子
標準コードファイル	%XMLTR%	codes.xml
ウインドウ AP 環境設定ファイル	%XMLTR%	env.xml
標準コード変換ルールファイル		*.crl
項目変換ルールファイル		*.trl
変換実行ファイル		*.xif
エラーログファイル		*.log
DTD ファイル	%XMLTR%\dtd	*.dtd
ＸＭＬファイル		*.xml
ＸＳＬファイル	%XMLTR%\xsl	*.xsl
非 XML ファイル		*.csv（CSV 項目名なし／あり） *.fix（固定長） *.cii（CII） *.edi（EDIFACT）

<注意>

DTD ファイルをアップデートする場合には、必ずトランスレータをインストールしたフォルダの中の“ DTD ”フォルダに格納されている DTD ファイルに変更を反映してください。

<注意>

インストーラは XML トランスレータをインストールしたディレクトリを、 レジストリキー[HKEY_LOCAL_MACHINE¥SOFTWARE¥SJAC¥XMLTranslator]に、"home"という名前付きデータとして書込みます。XML トランスレータは標準コードファイル、変換ルールファイル等の XML ファイルや、DTD ファイルがカレントのディレクトリにない場合、上記のディレクトリより探します。このため、上記レジストリキーのデータを変更しないでください。

（B） ウインドウ AP 環境設定ファイル

ウインドウ AP 環境設定ファイルには、以下の表に示す環境情報が定義されています。これらの内容は、環境設定画面により変更することができます。

（a） 情報区分コードと DTD ファイルの対応

情報区分コードおよび内容	DTD ファイル名
0301 見積依頼情報	SJAC0301.dtd
0302 見積回答情報	SJAC0302.dtd
0303 円貨確定依頼情報	SJAC0303.dtd
0304 円貨確定情報	SJAC0304.dtd
0500 注文情報	SJAC0500.dtd
0601 注文残高確認情報	SJAC0601.dtd
0701 納期確認情報	SJAC0701.dtd
0702 納期回答情報	SJAC0702.dtd
0704 出荷情報	SJAC0704.dtd
0705 入荷情報	SJAC0705.dtd
0801 検査情報	SJAC0901.dtd
0901 検収情報	SJAC0801.dtd
1101 買掛明細情報	SJAC1101.dtd

（b） 標準コードファイル名

codes.xml

（C） 標準コードファイル

標準コードファイルには、標準コードの一覧が定義されています。標準コードは、約 50 種類存在します。これらの各種類ごとに、「コード種別」及び「コード内容」の情報が設定されています。

４．４．３　コマンドラインからの変換実行

ここでは DOS コマンドの使用方法について説明します。DOS コマンドでは、変換実行ファイルの内容から、XML⇔非 XML 変換を行うことができます。

DOS コマンドを使用することにより、バッチ処理を行わせることができます。

・コマンドラインからの変換実行　……　(1)参照

（A） コマンドプロンプトより、以下のコマンドを実行します。

①コマンド名として"sjacxtr"を指定します。

C:¥WINDOWS>sjacxtr *convert.xif*

②引数を一つ与えます。この引数には、変換実行ファイルのフルパスを指定します。

<補足事項>

コマンド名として"sjaxctr"だけを入力して実行する場合、コマンドを実行するときのカレントディレクトリの中に"sjacxtr.exe"が存在するか、または"sjacxtr.exe"が格納されているディレクトリへのフルパスが、環境変数 PATH に設定されていなければなりません。

環境変数 PATH を設定しない場合は、コマンド名に"sjacxtr.exe"へのフルパスを指定するようにしてください("sjacxtr.exe"は XML トランスレータをインストールしたディレクトリに格納されています)。例えば、デフォルトのインストール先にインストールした場合、以下のように指定します。

C:¥WINDOWS>"C:¥Program Files¥XML トランスレータ¥sjacxtr" *convert.xif*

４．４．４　変換 DLL API 仕様

ここでは、変換 DLL が公開する API 関数の仕様を説明します。

変換 DLL を使用することにより、さまざまなアプリケーション(プログラム)において、XML⇔非 XML の変換処理を行うことができるようになります。

・概要 …… (1)参照

・XMLtoNonXML …… (2)参照

・NonXMLtoXML …… (3)参照

・Convert …… (4)参照

（A） 変換 DLL のファイル名は "sjacxtr.dll"です。この DLL が使用する他のファイルは、インストールを行ったときに、自動的にインストールされます。

変換 DLL は、XML⇔非 XML の変換を行うためのインタフェースとして、以下に示す 3 つの API 関数を提供しています。

API 関数名	内容
XMLtoNonXML	XML→非 XML 変換を行います。
NonXMLtoXML	非 XML→XML 変換を行います。
Convert	変換実行ファイルに従って、XML→非 XML 変換および非 XML→XML 変換を行います。

<注意>

API 関数には引数としてファイル名を指定します。このとき、各ファイル名は、フルパスで指定しなければなりません。相対パスで指定した場合、変換を実行した時のプロセスの状態(カレントディレクトリの状態)により、参照するファイルが異なることになります。

（A） 形式

```
int XMLtoNonXML(
    char *InputFilename,
    char *OutputFilename,
    char *TagRuleFilename,
    char *CodeRuleFilename,
    char *LogFilename
    );
```

（B） 内容

XML ファイルを、非 XML ファイルに変換します。このとき、1 つの XML ファイルを変換して、1 つの非 XML ファイルを生成します。出力ファイル(非 XML ファイル)が既存の場合、無条件に上書きします。

（C） 引数

引数名	型	内容
InputFilename	char *	入力ファイル(XML ファイル)名を指定します。
OutputFilename	char *	出力ファイル(非 XML ファイル)名を指定します。
TagRuleFilename	char *	項目変換ルールファイル名を指定します。
CodeRuleFilename	char *	標準コード変換ルールファイル名を指定します。
LogFilename	char *	変換のログを出力するファイル名を指定します。LogFilename に NULL を指定すると、ログの出力を行いません。

（D） 戻り値

戻り値	内容
1	生成したファイル数(変換に成功)。
-2	ログファイルにエラー内容を出力。

（A） 形式

```
int NonXMLtoXML(
    char *InputFilename,
    char *OutputFilename,
    char *TagRuleFilename,
    char *CodeRuleFilename,
    char *LogFilename,
    int suffix
    );
```

（B） 内容

非 XML ファイルを、XML ファイルに変換します。このとき、1 つの非 XML ファイルを変換して、複数の XML ファイルを生成します(非 XML の中に現れる、メッセージグループヘッダの単位ごとに異なるファイルが生成されます)。出力ファイル(XML ファイル)が既存の場合、無条件に上書きします。

<補足事項>

出力されるファイル名は、OutputFilename で指定されるファイル名に、suffix で指定される整数値を加えたものとなります。複数のファイルが生成された場合、各ファイルには、suffix+0，suffix+1，suffix+2，...の整数値が OutputFilename に付加されます。

例えば、OutputFilename に"c:¥xmltr¥foo.xml"が、suffix に 12 が指定された場合、実際に出力されるファイル名は、以下のようになります。

1 つ目のファイル名	c:¥xmltr¥foo-12.xml
2 つ目のファイル名	c:¥xmltr¥foo-13.xml
3 つ目のファイル名	c:¥xmltr¥foo-14.xml

（C） 引数

引数名	型	内容
InputFilename	char *	入力ファイル(非 XML ファイル)名を指定します。
OutputFilename	char *	出力ファイル(XML ファイル)名を指定します。
TagRuleFilename	char *	項目変換ルールファイル名を指定します。
CodeRuleFilename	char *	標準コード変換ルールファイル名を指定します。
LogFilename	char *	変換のログを出力するファイル名を指定します。 LogFilename に NULL を指定すると、ログの出力を行いません。
suffix	int	出力ファイル名に付与される整数値の開始値。

（D） 戻り値

【戻り値】

戻り値	内容
n	生成したファイル数(変換に成功)。
0	一つもファイルを生成しなかった場合。
-2	ログファイルにエラー内容を出力。

（A） 形式

```
int Convert(
    char *ConvertFilename
    );
```

（B） 内容

変換実行ファイルの内容に従って、非 XML⇔XML ファイルの変換を行います。非 XML→XML 変換、XML→非 XML 変換のどちらを行うかは、項目変換ルールファイルの内容によって、自動的に決定されます。

<補足事項>
非 XML→XML 変換の場合、出力ファイルに付与される整数の開始値は、常に 1 となります。

（C） 引数

引数名	型	内容
ConvertFilename	char *	変換実行ファイル名を指定します。

（D） 戻り値

戻り値	内容
n	生成したファイル数(変換に成功)。
0	一つもファイルを生成しなかった場合。
-1	引数に指定された変換実行ファイルの読み込みに失敗。
-2	ログファイルにエラー内容を出力。

<補足事項>
ログファイル名は、変換実行ファイルの中で定義されます。このため、変換実行ファイルの読み込みに失敗した場合は、エラー内容をログファイルに出力することができません。