



iPEX

Version 2.0

テクニカルホワイトペーパー

2000年5月

インフォテリア株式会社

XMLの重要性

1998年にW3Cによって勧告が発表されてから2年が経ち、XMLはBtoBおよびBtoC市場、さらには今後続々と登場するであろうさまざまなASPなどの用途で、重要なテクノロジーとして認知されるようになってきています。XMLがこれだけ注目を集めているのは、柔軟性が高くさまざまなデータを表現可能な点、インターネットへの対応、そしてベンダーやソフトウェアに非依存である点などが挙げられるでしょう。こうした特徴によって、XMLは標準データフォーマットとしての地位を着実に固めつつあります。

またXMLは、ソフトウェアの開発にもいくつものメリットをもたらします。たとえばプラットフォームに依存しないという点を活かして、クロスプラットフォーム環境のソフトウェアの標準データフォーマットとすることにより、複雑なコンバート作業などを行うことなく、データ交換を容易に行えます。また、XSLTと呼ばれる規格を実装することにより、プロダクト間でのデータの相互利用といったことも可能になります。

XMLに対応したソフトウェアの開発には、まずXMLパーサと呼ばれるプログラムが必要になります。XMLパーサとは読み込まれたXMLが妥当であるかどうかの検証やデータの読み込みや追加、削除を行うための機能をアプリケーションに提供するソフトウェアです。これを利用することによって、開発者は容易にソフトウェアをXMLに対応させることができます。

また他企業とのデータのやり取りなど、作成した XML データの応用範囲を広げて活用するには、「XSLT」や「XPath」といった規格への対応も必須になります。XSLT は、XML にスタイル情報を付加するための規格として考え出された「XSL」という規格から分離したもので、ある構造を持つ XML データを、別の構造の XML へと変換する機能を提供します。XPath は XSLT での変換に用いられる規格で、XML のアドレッシング機能などを提供します。

iPEX Version 2.0 の概要

iPEX Version 2.0 は、アプリケーション上で XML を扱い、さらに活用するための機能を提供する XML 処理エンジンです。DOM に準拠したパーサとして操作できるので、開発者は iPEX をソフトウェアに組み込むことによって、XML データの読み込みや書き込み、追加、削除などといった操作が可能になります。また、Version 2.0 では XSLT と XPath もサポートし、XSLT データを用意するだけで簡単に XML のスキーマの変換などが行えます。そのほかサポートする仕様は以下のとおりです。

XML 1.0

XML Namespaces

DOM Level1 + Level2 の一部

XPath 1.0

XSLT 1.0

iPEX Version 2.0 の特長

XSLT / XPath をサポート

今回のバージョンアップで、XSLT および XPath をサポートいたしました。これによって、XML を構造の異なる XML への変換や、XML から HTML への変換が可能になります。もちろん Compact HTML や WAP への変換も可能ですので、Web ベースのモバイルソリューションの構築にも威力を発揮します。

URL 指定による XML データの読み込みが可能

HTTP、もしくは FTP による URL を指定しての XML データの読み込みに対応

しました。サーバに蓄積されている XML データを読み込み、それをベースに XML を作成するといった処理が可能になります。

C++ライブラリ形式で提供

iPEX Version 2.0 は、C++用のスタティックライブラリとして提供されます。これによってアプリケーションへ直接リンクすることが可能なため、Windows 環境での DLL バージョン不整合によるシステム動作の不具合などを未然に防ぐことができます。また C++を採用したことで高いパフォーマンスを実現し、さらに高次元で安定して動作します。そのため高速性・安定性が求められるシステムへの適用にも安心してご利用いただけます。

また各 Windows 版は、C++形式のほか COM 版でも提供します。こちらをお使いいただくことで、Visual Basic などでも iPEX の優れた機能をご利用いただけます。

混在プラットフォーム環境での開発にも対応

iPEX Version 2.0 は、Windows 95 / 98 / NT、Windows 2000 のほか、Solaris や WindowsCE など数多くのプラットフォームをサポートします。いずれの環境でも同一のインターフェイスを提供していますので、プラットフォームの違いを意識することなく XML データの制御のために iPEX Version 2.0 をご利用いただけます。

対応プラットフォーム

Windows 95 / 98 / NT / 2000

Solaris

WindowsCE(*)

Linux

Windows COM 版

* WindowsCE 版には、DTD の読み込みおよび XSLT/XPath 対応部分は省かれています。

14 種類のエンコーディングに対応

UTF-8 / UTF-16 をはじめとする 14 種類のエンコーディングに対応しています。もちろん日本語を扱う上で欠かせない「Shift_JIS」「EUC-JP」「ISO-2022-JP」といったエンコーディング方式にも対応しました。

対応エンコーディング一覧

UTF-16
UTF-8
Shift_JIS(x-sjis-cp932)
euc-jp(x-eucjp-open-19970715-ms)
iso-2022-jp(x-iso2022-jp-cp932)
US-ASCII
欧文エンコーディング 9 種

充実のサポート体制

さらに iPEX は、iPEX を組み込んだパッケージソフトウェアを第三者に販売するために、プロフェッショナルエディションをご用意しています。こちらをご購入いただくことで、「iPEX 本体」「ランタイムライセンス」に加えて、最新バージョンの iPEX を常にご利用いただける「メンテナンス契約」といったサービスを提供します。また、iPEX Professional Editiron Version 2.0 には、弊社製品「iXSLT」で提供している XSLT 変換専用の DLL および COM モジュールも付属しています。

iPEX Version 2.0 の実装例

iPEX Version 2.0 は、効率的なソフトウェア開発をサポートするために、少ないソースの記述で簡単に XML の機能を取り込めるよう開発されています。XML の呼び出しや DOM 木の作成と XML の書き込み、XSLT を使った変換など、以下のような簡単な記述で利用できます。

```
// Sample1
// ストリームからの XML の読み込み

Document* readXML(const IPEXWSTRCLASS& strUrl)
{
    InputStreamByUrl is(strUrl);
    if (!is)
        return 0;
    std::auto_ptr<Document> pDoc = IPEXDocument::createDocumentObject();
    XMLReader reader(&is);
    if (!reader.read(pDoc.get()))
        return 0;
    return pDoc.release();
}

// Sample2
// DOM 木の作成

Document* constructTree()
{
    try {
        std::auto_ptr<Document> pDoc = IPEXDocument::createDocumentObject();
        std::auto_ptr<Element> pElem = pDoc->createElement(L"doc");
        pElem->setAttribute(L"foo", L"bar");
        pDoc->appendChild(pElem.release());
        pElem->appendChild(pDoc->createTextNode(L"sample text"));
        return pDoc.release();
    }
    catch (DOMException&) {
    }
    return 0;
}
```

```
// Sample3
// XML の書き込み

bool writeXML(Document* pDoc, const IPEXWSTRCLASS strPath)
{
    OutputStreamByFile os(strPath);
    if (!os)
        return false;
    XMLWriter writer(&os, L"euc-jp");
    if (!writer.write(pDoc) || !os.close())
        return false;
    return true;
}

// Sample4
// XSLT を使った XML 変換

Node* processXSLT(Document* pDocXml, Document* pDocXslt)
{
    return pDocXml->evalXSLT(pDocXslt->getDocumentElement());
}
```