Advantages of the OpenOffice.org XML File Format Used by the StarOffice™ Office Suite

White Paper
April 2004

Table of Contents

Chapter 1

Preface

The StarOffice™ office suite is based on an open source project that was founded by Sun Microsystems and the OpenOffice.org organization. As a result, both the StarOffice and OpenOffice.org office suites use exactly the same file format. Since the file format is specified in a public document, *"OpenOffice.org XML File Format 1.0 — Technical Reference Manual,"* this white paper refers to the StarOffice file format as the OpenOffice.org XML file format.

The first part of this white paper looks at the benefits and business aspects of an eXtensible Markup Language (XML) file format. The last chapter is targeted at a more technical audience, including people who want to explore the OpenOffice.org XML file format in more detail in order to experience its advantages.

Chapter 2

The Benefits of XML

What Vendors Did In the Past

Until recently, software vendors used proprietary and often poorly documented file formats. The StarOffice suite (up to version 5.2) was not much different in that regard, because it used then-current technology. StarOffice software also used a binary file format for efficiency reasons (XML was still an emerging technology).

In the past, desktop systems did not provide much memory, disk space, and CPU power, so it was not feasible to store complex document information in a memory-inefficient ASCII or text format. Plus, parsing text data is very CPU intense. Therefore, old file formats contained only ASCII-formatted data that did not need to be converted into internal binary information. This meant that some content was kept as ASCII data while layout coordinates or style details were stored as binary data. Since all applications used in-memory representation to store document data in corresponding document files, file formats were very closely related to their creating applications. As a consequence, a text document file from vendor A was incompatible with the format used by vendor B, and vice versa.

Another problem was that every new application version introduced features that required new data fields in the corresponding file formats. Because of this, even applications from the same vendor used different and incompatible file formats for different versions of the same application, which made it difficult to exchange files among people and companies — unless everybody was using the same applications and application versions.

Because there was no guaranteed compatibility between different versions of the same application, being able to open old files required the availability of all corresponding application versions, unless a new application version supported all past file format versions. This was especially problematic in areas where documents had to be archived for a long time. Therefore, many document management systems chose a different file format as a workaround. Most of the time, only the graphical representations of documents were stored in more accessible formats, including Tagged Image File Format (TIFF) or Portable Document Format (PDF).

Storing document information in a binary file format also had another disadvantage. If files became corrupted, it was almost impossible to access the document data. Since many companies store important business data in text and spreadsheet documents, losing the data from just one file could be a significant loss.

A few years ago, most people did not care that much about interoperability. Today, however, the integration of heterogeneous business systems is a very important topic. Most enterprises want to streamline their processes to cut down operation costs. In the past, integration was often achieved by using macro-based scripting capabilities and exchange file formats that more or less represented the most common denominator between different applications. Recently, the demand for interoperability among applications has increased dramatically. This becomes obvious when one looks at the rate of adoption of XML and Web services by most software vendors.

What Is XML?

There are different definitions and opinions about what XML is, and what its real purpose and usage should be. Many vendors introduced XML as an application-independent format of data for exchange, import, and export purposes. Similar to the HyperText Markup Language (HTML), which represents layout information in an application-agnostic format, XML focused on the application-independent representation of data fields and records.

Since the focus of XML is on data, this implies separating data from its representation. However, in the case of office documents, even layout information is important document data, and thus can and should be stored using XML together with the core document data, including its content. In addition, formatting information should be as open as the document content — it should be easily readable and well-documented so that it can be processed by other applications.

Why XML Solves Problems From the Past

OpenOffice.org 1.0 introduced a new, XML-based document format that both OpenOffice.org and the StarOffice office suite now employ as their default file format. It is also the basis for the Organization for the Advancement of Structured Information Standards (OASIS) Open Office XML format, a standard document file format defined by representatives of various industries.

In contrast to the old binary formats, the new XML file format is relatively independent from office applications that generate files and application-specific implementation of document features. It was designed to be application agnostic. Since XML is extensible, new features can be added to applications without modifying underlying file formats. New features simply generate new XML tags that are ignored by earlier applications and application versions that are not aware of these newer features. The benefit is that a software upgrade in one department does not require the new software to be installed throughout the entire company. An open, XML-based file format can help avoid the frequent and costly impact that upgrades often required in the past.

Because XML is “human readable,” document data is still accessible when the corresponding application goes end of life. In the worst case, the data can be accessed using a simple text editor. This is especially important when documents must be archived for many years. Fortunately, there are hundreds of XML tools available that make it convenient to view and modify XML files. It is also very important when it comes to using documents within business processes. In the past, documents were integrated into business processes using the scripting capabilities of corresponding productivity applications. It was very difficult to read data from or write data to existing documents because the binary file formats were optimized for the applications that generated them.

An XML file format, on the other hand, can be integrated into business processes because text files can be easily parsed and processed. The broad adoption of XML and Web services by most major software vendors — as well as excellent XML support in programming and scripting languages such as Java™ technology, Perl, and Python — has significantly reduced the complexity of integrating XML-based document files.

Who Benefits From an Open XML File Format

There are many fields where support for XML within document files was and is especially beneficial. Application systems that archive documents for a long time, such as document management systems (DMS), stored the files in formats for which it was very likely that there would be viewers for many years.

Most of the time, these applications simply used a format that allowed people to reprint documents if needed. Popular formats in this application area were (and still are) TIFF and PDF. Unfortunately, this approach made it impossible to edit the documents again at a later point. In addition, metadata such as author names, change dates, and so on had to be entered manually, and once documents had been stored in the document management system as graphical read-only files, it was impossible to access the content again. Since the OpenOffice.org XML file format contains both content and layout information, it is perfect for usage in document archives. This is an important reason why the National Archive of Australia joined the OASIS Open Office XML Format Technical Committee (TC) and chose OpenOffice.org for archiving documents.

A similar usage is within content management systems (CMS). With XML file formats, it is very easy to generate documents from scratch or merge document templates with business data. In many cases, this can be achieved using simple standard eXtensible Stylesheet Language Transformations (XSLT) operations.

Since most proprietary file formats belong to applications that are available on only one or two operating systems, an open XML-based file format can help to support a multiplatform environment. Software vendors can write applications that implement the standard XML file format, or in the case of OpenOffice.org, they can even use existing source code and port it to new platforms.

In today's world, most human knowledge is still stored in unstructured formats. This includes legal documents, contracts, business plans, technical specifications, and scientific essays. Unfortunately, many of these documents are saved in poorly documented, binary file formats, which can make the content of these documents inaccessible if the corresponding application or application version disappears.

The well-documented XML file format of OpenOffice.org, on the other hand, can help prevent that important content from being locked into a proprietary format and application. This makes the full content accessible even outside of the generating application.

As explained earlier, in any business automation application where documents must be automatically parsed, generated, or modified without human interaction, an XML file format is a significant benefit. There are two basic integration approaches: The first is to create XML files that are optimized for a specific use case or business application. Since users must know up-front how their document files will be used in the future, the OpenOffice.org XML file format follows the second approach, which is to design a generic, multipurpose file that contains all document information.

Chapter 3

Advantages of the OpenOffice.org XML File Format

StarOffice software is based on the open source project and application OpenOffice.org. For this reason, both office suites share the same XML file format and its advantages. The following sections explain these unique file format features in detail.

Summary of Advantages

- Open
  - 500-page specification document available at xml.openoffice.org
  - No undocumented proprietary elements (no lock-in)
  - Basis for the OASIS Open Office XML format standardization effort
  - Leverages established standards like Dublin Core, XSL-FO, XLink, SVG, and HTML
  - File format allows adding features to productivity applications without breaking file format compatibility

- Universal, multipurpose file format
  - XML file format is the default file format (users do not have to choose it)
  - All applications use the same XML file format (word processor, spreadsheet, presentation)
  - Files include full document information (content, metadata, macros, formatting information)
  - Not limited to specific business applications

- Small file size
  - ZIP compression (native support for ZIP format in tools such as Perl, Ant, and most operating systems such as the Solaris™ Operating System, Linux, and Windows XP)
  - Images included as native files instead of Base64-encoded ASCII data

- Separation of content, data, and formatting information
  - Separate XML files for content, metadata, styles, and macros
  - Content file includes assigned style names, but no style details

- Full XML support in all editions
  - Same XML file format for free OpenOffice.org office suite and commercial StarOffice software
  - Platform-independent XML file format (Microsoft Windows, Linux, Solaris OS)

- XSLT-based XML filter tool for standards-based interoperability
  - Allows import and export of third-party file formats using XSLT transformations

- Third-party support
  - Future versions of KOffice will use the OASIS XML file format as the default
  - Cross-industry OASIS technical committee members — not just a single vendor — are defining the future of the OASIS XML file format.

Open

Compared to traditional binary file formats, the XML file format used by StarOffice software has an advantage because all information is also accessible from outside the StarOffice application. Using a simple unzip tool (such as WinZIP) and a text or XML editor, it is possible to open and use content, formatting, metadata, and macro information. This is especially of interest if the files will either be archived for years or processed by other applications.

The 500-page specification of the XML OpenOffice.org file format — the same file format StarOffice software uses — has been published on the OpenOffice.org Web site. A PDF file of the specification is also included in the StarOffice Software Development Kit (SDK). The document explains in detail the structure and contents of the XML file format.

Since the file format does not have undocumented elements, vendor lock-in is impossible. In contrast to other file formats, the OpenOffice.org file format is free from proprietary extensions. In addition, the open source project OpenOffice.org is the basis of the StarOffice application suite. This fact, combined with the licenses under which OpenOffice.org is available — the GNU Lesser General Public License (LGPL) and Sun Industry Standards Source License (SISSL) — also ensures that no vendor, including Sun Microsystems, can make proprietary changes to the file format. In this context, the free, open source OpenOffice.org office suite takes on the role of a reference implementation of a file format that is open and available to everyone.

In order to make transformations into other XML file formats as simple as possible, the OpenOffice.org file format makes use of established standards wherever it can. It leverages standards such as the Dublin Core, eXtensible Stylesheet Language Format Objects (XSL-FO), XML Linking Language (XLink), Scalable Vector Graphics (SVG), and HyperText Markup Language (HTML). All of these standards have limitations that make it impossible to use them directly for an office productivity application file format. However, to enable an optimal level of interoperability with third- party applications, existing standards such as these were leveraged instead of creating new proprietary and incompatible definitions.

StarOffice software is available on a variety of platforms including the Solaris OS, Linux, and Microsoft Windows. The open source community of the OpenOffice.org project has ported the application to many additional platforms. Therefore, the OpenOffice.org file format is also available on platforms such as the Mac OS X, FreeBSD, and Irix. More ports are in various stages of development.

In addition, the StarOffice office suite is available in ten major languages, with more planned in the future. Again, the open source community around OpenOffice.org has created additional localizations that are not yet covered by the StarOffice suite. The list of languages for which there is application support in the XML format is fast growing, and even includes languages that are often ignored by commercial vendors due to their insignificant revenue potential. However, support for languages — such as Zulu in South Africa — allows citizens of these countries to participate in the worldwide exchange of documents in their native language. For this reason, some StarOffice and OpenOffice.org localization projects are sponsored by governments.

Universal, Multipurpose File Format

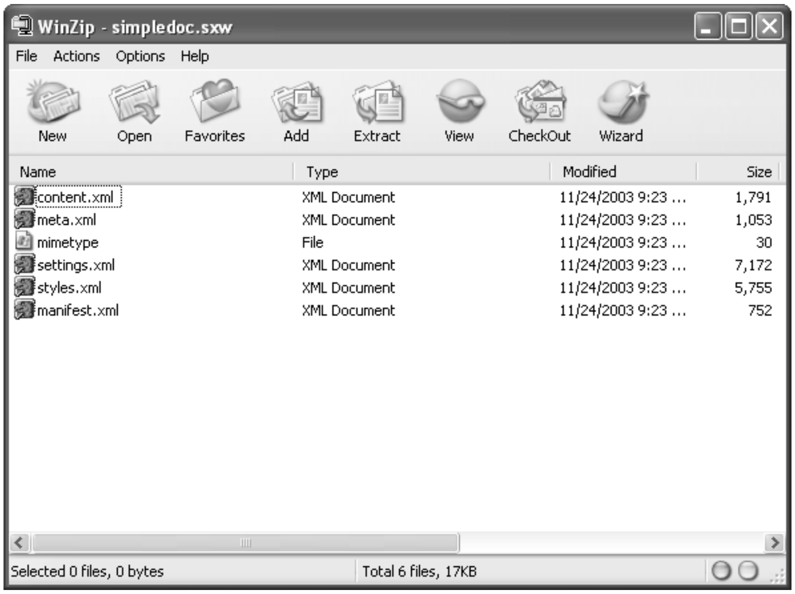
Unlike other office productivity applications that support XML as an additional export or exchange file format, StarOffice software uses the XML file format as its native file format. This means that users do not have to explicitly choose if they want to use the XML file format instead of a binary file format (unless they have voluntarily chosen a different file format as the default). For the same reasons, users also do not have to know up-front how either they, or the recipients of the document, will use the file.

The StarOffice suite employs the same XML file format across all applications, so the he word processor, spreadsheet application, and presentation tool all employ exactly the same XML file format. For example, tables in text documents have the same XML structure as tables in spreadsheet documents, although they are implemented differently within the office productivity application. This is a very powerful feature when it comes to document processing. These tables can be converted into other formats, such as HTML, using the same XSLT transformations.

Since the XML file format is the default in OpenOffice.org and StarOffice software, it is also guaranteed that all document information is included and available in the document files. Because it is the default, use of the XML file format does not cause functional restrictions or presentation losses. Some vendors limit their XML support to data and content exchange, or simply make it difficult to use formatting and layout information in a different application. Unfortunately, this also limits file usability in some scenarios. The XML file format utilized in the StarOffice suite includes not only the document content, but also metadata, macro, and formatting information.

However, to achieve optimal flexibility, the OpenOffice.org file format separates data from its presentation information by separating document information into a variety of XML files that are all stored in a single ZIP archive: The StarOffice .sxw, .sxc, or .sxi file.

Figure 3-1: StarOffice File In the ZIP Tool



Name	Type	Modified	Size
content.xml	XML Document	11/24/2003 9:23 ...	1,791
meta.xml	XML Document	11/24/2003 9:23 ...	1,053
mimetype	File	11/24/2003 9:23 ...	30
settings.xml	XML Document	11/24/2003 9:23 ...	7,172
styles.xml	XML Document	11/24/2003 9:23 ...	5,755
manifest.xml	XML Document	11/24/2003 9:23 ...	752

Selected 0 files, 0 bytes

Total 6 files, 17KB

If a user unzips a StarOffice file, it can be readily seen that the StarOffice file actually consists of a set of files. A StarOffice file is a ZIP archive that contains files such as content.xml, styles.xml, and meta.xml as well as files of included pictures. This enable changing content without touching the formatting, replacing low-resolution images with high-resolution images, or changing all styles at once in order to apply a new corporate formatting.

As previously mentioned, the OpenOffice.org XML file format was not designed for just one set of scenarios or business applications. The intent was to create a file format that could be used across industries and vendors.

Another goal was to have a file format that would not break, enforcing software upgrades every time a new feature was added to an office productivity application. Tags required by new application functionality that are unknown to an older version of OpenOffice.org or the StarOffice suite are simply ignored by applications, unless an entire implementation concept changes. The latter case was purposely chosen for the adoption of the derived OASIS standard file format. The OASIS Open Office XML Format Technical Committee (TC) added a few changes that are incompatible with the original OpenOffice.org file format. For example, the new file format employs the OASIS namespace URI instead of the OpenOffice.org namespace URI. However, because the OASIS file format was defined by a large multivendor, multiindustry committee, it is anticipated that it will remain stable.

Small File Size

Using ZIP compression might seem like a limitation or breach with the simple text concept of XML, but the usage of ZIP archives is actually an advantage. XML files become very large because most data is represented by uncompressed ASCII text information so that it can be read and processed easily. This it not a problem for a single document, however, in an enterprise setting where hundreds of documents are created daily, this can quickly create a storage problem.

ZIP compression is applied to various files that contain XML information. Thus, the content of the XML files themselves is not compressed and can easily be processed. For the same reason, images are included as image files, such as Joint Photographic Experts Group (JPEG), instead of Base64 encoded data. This approach keeps file sizes small, but maximizes information accessibility. Also, it makes loading images on demand possible.

ZIP compression might appear to be a big hurdle with respect to the usage of StarOffice XML files in business processes, but the reality shows that this is not the case. Most operating systems already include tools that allow them to access ZIP archives, including the Solaris OS, Linux, and Windows XP. For older platforms such as Windows NT, third- party tools such as WinZIP fulfill this need.

What is probably more important is that many scripting languages and build tools provide support for creating or opening ZIP archives. Two good examples are Perl and Ant. For most programming languages — commercial or free — open source libraries for the ZIP algorithm are available. Java technology supports ZIP files as part of its Java Archive (JAR) implementation.

Separation of Content and Formatting Information

As previously earlier, StarOffice files separate content, styles, metadata, and macros. Nevertheless, that does not mean that the content's XML file contains only pure data and nothing else. All XML files contain some header information, such as namespace definitions, in addition to the actual content. It is possible to maintain the complete document structure only if there is a relationship between content and layout information. Therefore, the content file must include some style information that links some of its data elements to specific formatting information in the styles file.

It is questionable whether hard formatting — formatting that overrides a style — should be treated as content or style. In most cases, a word marked *bold* should stay that way, even if the styles applied to the word change. Therefore, StarOffice software keeps this information together with the content. Keeping all macros in one location instead of merging them with actual document content also has a security advantage, because document files can be easily parsed for included code. This makes it much simpler to scan documents for malicious macros.

Full XML Support in All Editions

In contrast to other office productivity suites, all editions of StarOffice software provide the same XML support. Both the commercial StarOffice suite as well as the free, open source reference implementation, OpenOffice.org, use the same XML file format. Thus, the amount of XML support users receive is not a budget question.

Other vendors provide full XML support only in the more expensive editions of their products, so users with smaller budgets may be locked out of the full benefits of an XML-based file format.

XSLT-Based XML Filter Tool for Standards-Based Interoperability

The new StarOffice 7 office suite introduces an XML filter tool that simplifies integration with other XML standards and applications. This tool makes it possible to easily add new import and export filters that use XSLT transformations to convert StarOffice documents into third-party formats, and vice versa. Sample filters for the eXtensible HyperText Markup Language (XHTML), Word 2003 XML, and DocBook are included in the application.

Note – These filters must be installed using the custom setup process.

The XML filter tool can also generate custom XML formats such as the Universal Business Language (UBL). These formats could, for example, contain only a small subset of the document information that is required for a specific task.

Third-Party Support

The advantages of the StarOffice suite's multipurpose approach have been validated by the industry, as evidenced by the well-known enterprises and organizations that have joined the OASIS Open Office XML Format Technical Committee (TC). The committee is using the OpenOffice.org XML file format as the basis for its work, with the goal of creating an open, XML-based, cross-industry file format that can be adopted by multiple vendors.

The committee plans to improve areas of the file format that could not be sufficiently defined by a single vendor alone. Efforts like these require the expertise of a variety of parties. Instead of encouraging the creation of custom XML schemas by every user and company, the OASIS TC focuses on the creation of standards that can be used across industries.

The first third-party office suite that adopted the OASIS standard — in addition to OpenOffice.org and StarOffice software — is the KOffice open source productivity suite. Several independent software vendors are also already leveraging the powerful StarOffice suite's XML file format. Major companies, including Software AG and Struktur AG, have created document and content management systems that integrate with OpenOffice.org and the StarOffice suite.

Chapter 4

The OASIS Open Office XML Format

The Benefits of a Cross-Industry Office Document Standard

Considering the current market situation, where one office productivity suite has more than 90-percent market share, people might wonder why that application and its corresponding file formats should not be declared the standard. It is very tempting to do this. If the de facto standard were open and well designed from a technical perspective, and if it did not put other vendors or users in a disadvantageous position through technical limitations and restrictive licensing terms, it might be fine to declare it an open standard. Unfortunately, this is usually not the case with formats controlled by a single vendor.

History has shown how important open standards are, and that the market leader of today may not be a key player in the future. In the office productivity market, applications such as WordPerfect and Lotus 1-2-3 once dominated the market. At the height of their popularity, these applications and their file format were de facto standards. However, it is now clear that it would not have been a sound strategy to adopt these file formats as industry-wide, country-wide, or global document file format standards. Today, these applications have a much lower market share. Although these binary file formats are still supported by their vendors and might gain market share again, it is very difficult for their users to switch to different, possibly more advanced technologies.

Even using open standard technologies such as XML within a document file format does not automatically make the format open. If the format is still cryptic and almost unusable outside its generating application, the usage of XML does not provide any benefit. It is still necessary to access the files by using the application's APIs and macro languages or by writing complex transformations (for example, XSLT). The only party able to use the format may be the vendor that invented it.

It would be even worse if licensing and patent restrictions prevented anyone other than the original vendor from actually using or extending a standard file format. It is always desirable for a format to evolve and improve— that is why the X in XML stands for eXtensible.

Like any other cross-vendor or cross-industry standard, a document file format standard cannot be defined by a single supplier. To achieve the highest level of openness and interoperability, it is important to leverage the expertise of many different industries. For example, a software vendor may not know the documentation or word processing requirements for the design and creation of a 747 jumbo airliner.

There also must be some kind of structural insurance in place to ensure that the standard will continue to evolve, remaining openly and equally accessible to all participants. What qualifies the work of groups as "open" is broad participation and access to publicly documented work.

The evolution and success of Java technology shows what can be achieved through the collaboration of various vendors with respect to defining standards such as the Java 2 Platform, Enterprise Edition (J2EE™ platform) or the Java 2 Platform, Micro Edition (J2ME™ platform). Java technology was originally invented by Sun Microsystems, but today there are many vendors — including Oracle, IBM, BEA, Sony, and Nokia — that define new programming interfaces and technologies. Sun alone would not have been able to define all these standards, because Sun does not have the most expertise in every industry, including mobile phones, databases, 3D imaging, healthcare, document management, and so on. The definition of a document file format standard is no different. No single vendor has complete knowledge of all the requirements emanating from legal bodies, scientists, engineers, archivists, and other specialist communities.

In the case of a document file format that is used for communication between governments and their citizens, it is almost a constitutional right that everybody has the same opportunity to access and use this content. This means it is more inclusive if the application has a free reference implementation that can handle the document's file format. This is becoming more important in a world where governments are implementing e-government strategies that do not rely on paper-based communication.

The following statement of purpose is taken from the OASIS Web site, and explains the goals of the OASIS Open Office XML Format TC:

"The purpose of this TC is to create an open, XML-based file format specification for office applications. The resulting file format must meet the following requirements:

1. it must be suitable for office documents containing text, spreadsheets, charts, and graphical documents,
2. it must be compatible with the World Wide Web Consortium (W3C) XML v1.0 and W3C Namespaces in XML v1.0 specifications,
3. it must retain high-level information suitable for editing the document,
4. it must be friendly to transformations using XSLT or similar XML-based languages or tools,
5. it should keep the document's content and layout information separate such that they can be processed independently of each other, and
6. it should borrow from similar, existing standards wherever possible and permitted."

The OASIS Open Office XML Format TC Members and Their Expertise

The OASIS Open Office XML Format TC brings together the expertise of many vendors, industries, and individuals. Companies, organizations, and projects participating as full members are Arbortext, Corel, OpenOffice.org, KOffice, National Archives of Australia, Society of Biblical Literature, Sun Microsystems (TC Chair), and Boeing. The current member list can be found on the Web at oasis-open.org/committees/membership.php?wg_abbrev=office.

In addition, several other companies and organizations represented through individuals with observer status are not listed on the OASIS site. The OASIS file format specification is influenced by representatives from market-leading desktop application vendors, government authorities, and open source projects.

The OASIS Open Office XML Format TC also leverages work from other standardization committees, including SVG, XForms, XSLT, XML Schema Definition (XSD), and Relax NG. As one of the TC members, Sun also stays in touch with other OASIS efforts, such as the Electronic Business eXtensible Markup Language (ebXML) and the UBL TC, headed by one of XML's founders, Jon Bosak of Sun Microsystems.

The Relationship Between the OpenOffice.org XML File Format and the OASIS Open Office XML Format

Work on the OASIS Open Office XML format started with the OpenOffice.org XML file format specification as its basis. Contributions by all OASIS Open Office XML Format TC members have pushed the OASIS format far beyond the original OpenOffice.org XML file format specification. It is planned that future versions of OpenOffice.org and StarOffice software will adopt the new OASIS file format as their default.

Enhancements and changes will make the OASIS file format incompatible with the original OpenOffice.org file format, partly because the new OASIS format will use the OASIS namespace URI instead of the OpenOffice.org version. However, because the original OpenOffice.org XML file format was open and well designed, it will be easy to write transformation scripts to convert OASIS files into OpenOffice.org XML files and back again. New versions of OpenOffice.org and StarOffice software will support both the old OpenOffice.org file format and the new OASIS format. For older product versions, a patch will be provided so that the adoption of the open standard OASIS format will not cause major disruptions.

Proposed Changes and Enhancements

As outlined earlier, the OASIS Open Office XML format is defined by individuals representing multiple industries. As such, it has evolved based on the expertise of its Technical Committee members. For example, the representatives for desktop publishing-related applications (Corel and KOffice) provided significant input regarding the way features like page-based layout should be implemented within the file format. The same representatives also drove the specification of business charts within documents. Vendors with experience in document management were able to improve the concepts concerning metadata handling. Others had input regarding change tracking. Overall, all OASIS Open Office XML Format TC members helped eliminate application-specific concepts and confusing names, simply because more people were looking at the specification from various angles.

Drafts of the OASIS format specification can be found in the documents section on the OASIS Web site at oasis-open.org/committees/documents.php?wg_abbrev=office. The final specification document of the first phase of the OASIS Open Office XML standardization is expected to be ratified and published in the first half of 2004. Shortly after that, the first applications, including OpenOffice.org, will support the new XML file format.

Who Is Adopting the Standard

People Using OpenOffice.org and the StarOffice Office Suite

Everyone who uses either OpenOffice.org or StarOffice software is adopting the OASIS standard to some degree, because the OASIS Open Office XML format is based on the OpenOffice.org XML file format specification, and it is anticipated that future versions of OpenOffice.org and StarOffice software will implement the OASIS standard.

The user base of OpenOffice.org and StarOffice software is growing rapidly, especially in countries where people simply cannot afford to buy expensive software. This is the case in many Asian, African, and South American countries. Other countries are showing interest in OpenOffice.org and the StarOffice suite because the applications are developed in an open source model. Thus, the OASIS format has a very real potential to be not only an open standard, but also to become the future de facto standard.

The National Archives of Australia

The National Archives of Australia (NAA) will use the OpenOffice.org office suite to preserve the quality and accessibility of government documents. The team responsible for this project decided that using proprietary document file formats for archiving purposes would not ensure that these files could be opened and replicated on computer systems of the future. The NAA is required to archive some documents for thirty years.

KOffice

The open source office suite KOffice will implement the OASIS Open Office XML format in the future because the project team realizes that it does not make sense to “reinvent the wheel.” Since the OpenOffice.org file format meets KOffice requirements with regard to openness and usability, the team plans to use the OASIS format as its default.

Related Initiatives

Several initiatives are related to the efforts of the OASIS Open Office XML Format TC. Some focus on XML standards, while others look at document file format standards from a slightly different angle. A few examples are:

- W3C (w3c.org)
- OASIS (oasis.org)
- The Mozilla Foundation (mozilla.org)
- OpenOffice.org (xml.openoffice.org)
- 1doc.org (1dok.org/eng/index.html)

Chapter 5

Exploring the OpenOffice.org XML File Format

This section explains simple ways to become more familiar with the open XML file format used by OpenOffice.org and the StarOffice productivity suite.

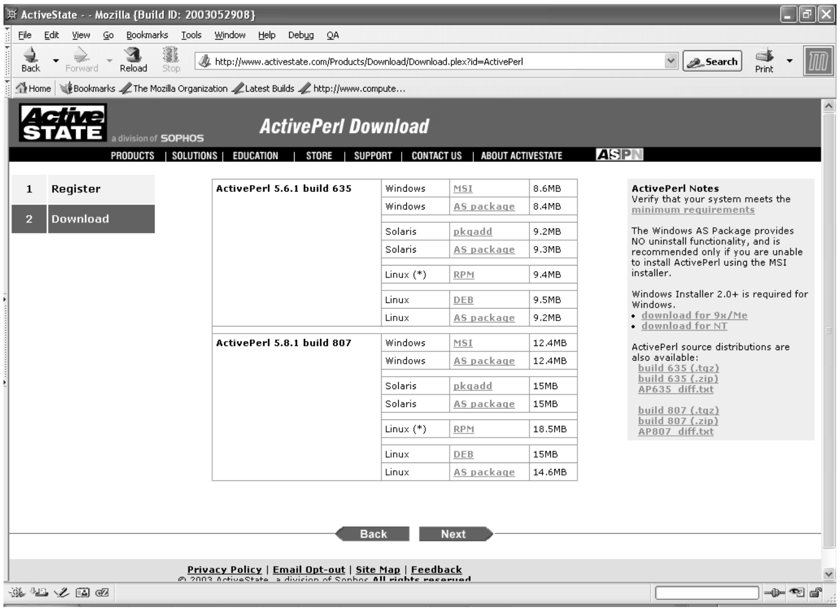
StarOffice software runs on multiple platforms, including the Solaris OS and Linux. Nevertheless, the Microsoft Windows platform is used here to illustrate some capabilities of the XML file format because most people who are not yet familiar with the OpenOffice.org file format are using an office suite that runs on Microsoft Windows. Thus, Microsoft Windows commands and screenshots should make evaluation of the OpenOffice.org file format easier for the majority of users who have not switched to a Linux or Solaris OS desktop yet.

In addition to StarOffice software itself, some extra tools are used. These are either free or inexpensive, and can be evaluated for free. These tools include a ZIP archive tool (WinZIP), a Java technology-based build tool (Ant), and a Perl runtime environment (ActivePerl).

Making a Simple Change to an Existing StarOffice File

For reasons of simplicity, the demonstration begins with an extremely simple example: A new text document that says "StarOffice XML Blue Paper" (Figure 5-1).

Figure 5-1: Simple Document In the StarOffice 7 Office Suite


ActiveState - - Mozilla {Build ID: 2003052908}
File
Edit
View
Go
Bookmarks
Tools
Window
Help
Debug
QA
Back
Forward
Reload
Stop
http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl
Search
Print
Home
Bookmarks
The Mozilla Organization
Latest Builds
http://www.compute...
Active STATE
a division of SOPHOS
ActivePerl Download
PRODUCTS | SOLUTIONS | EDUCATION | STORE | SUPPORT | CONTACT US | ABOUT ACTIVESTATE
ASPN
1 Register
2 Download
Back
Next
Privacy Policy | Email Opt-out | Site Map | Feedback

ActivePerl 5.6.1 build 635	Windows	MSI	8.6MB
	Windows	AS package	8.4MB
	Solaris	pkgadd	9.2MB
	Solaris	AS package	9.3MB
	Linux (*)	RPM	9.4MB
	Linux	DEB	9.5MB
	Linux	AS package	9.2MB
ActivePerl 5.8.1 build 807	Windows	MSI	12.4MB
	Windows	AS package	12.4MB
	Solaris	pkgadd	15MB
	Solaris	AS package	15MB
	Linux (*)	RPM	18.5MB
	Linux	DEB	15MB
	Linux	AS package	14.6MB

ActivePerl Notes
Verify that your system meets the minimum requirements

The Windows AS Package provides NO uninstall functionality, and is recommended only if you are unable to install ActivePerl using the MSI installer.

Windows Installer 2.0+ is required for Windows.
- download for 9x/Me
- download for NT

ActivePerl source distributions are also available:
build 635 (.tgz)
build 635 (.zip)
AP635_diff.txt

build 807 (.tgz)
build 807 (.zip)
AP807_diff.txt

As a next step, a ZIP tool, such as WinZIP, is used to access the StarOffice file's XML content (Figure 5-2).

Figure 5-2: Open StarOffice File With the ZIP Tool

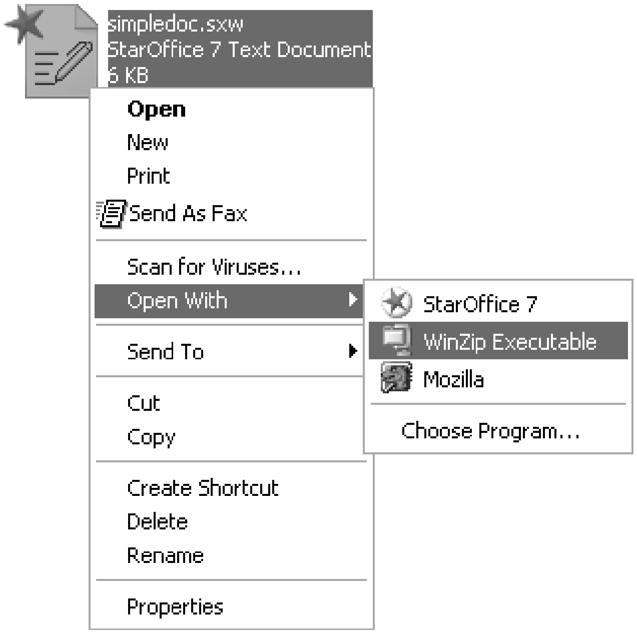
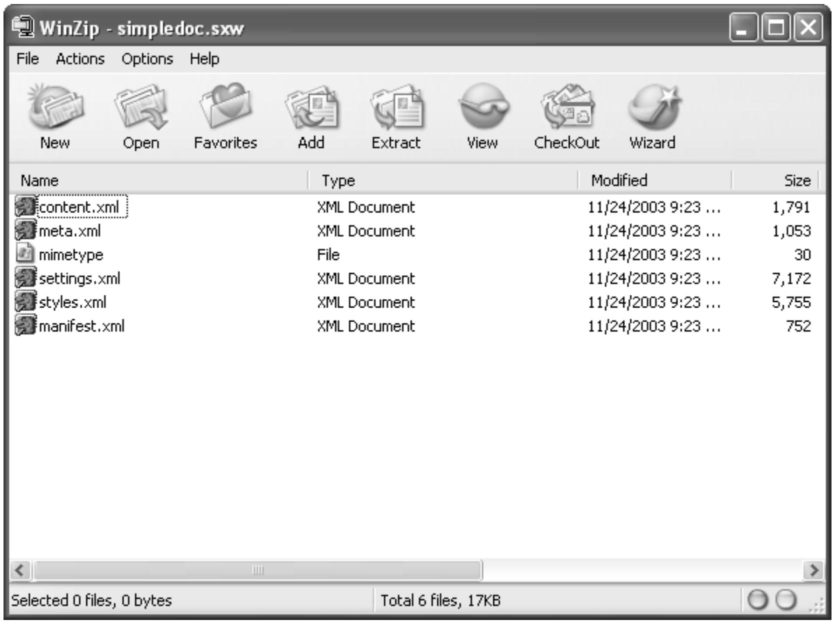


Figure 5-3 shows the various XML files contained by a StarOffice file.

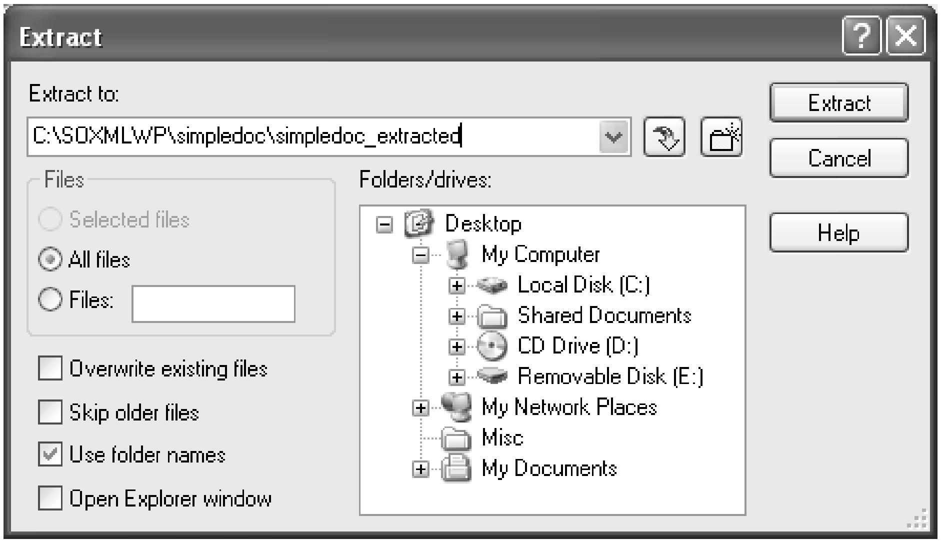


Name	Type	Modified	Size
content.xml	XML Document	11/24/2003 9:23 ...	1,791
meta.xml	XML Document	11/24/2003 9:23 ...	1,053
mimetype	File	11/24/2003 9:23 ...	30
settings.xml	XML Document	11/24/2003 9:23 ...	7,172
styles.xml	XML Document	11/24/2003 9:23 ...	5,755
manifest.xml	XML Document	11/24/2003 9:23 ...	752

Selected 0 files, 0 bytes   Total 6 files, 17KB

Figure 5-3: Contents of a StarOffice File In a ZIP Tool

The StarOffice file has separate XML files for content (content.xml), metadata (meta.xml), and formatting information (styles.xml). To modify the file outside of the StarOffice suite, its content must be extracted into a separate directory (Figure 5-4).



Extract to: C:\SOXMLWP\simpledoc\simpledoc_extracted

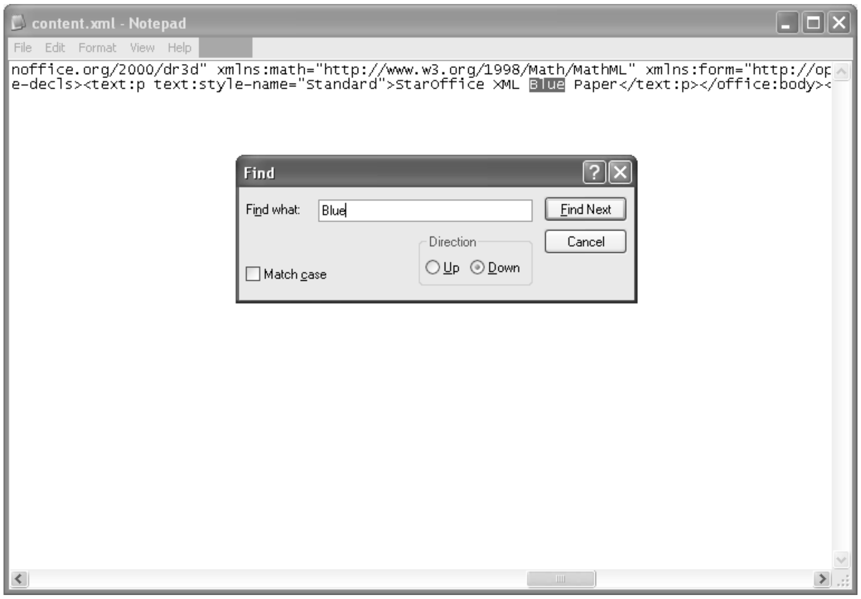
Files
○ Selected files
● All files
○ Files:

□ Overwrite existing files
□ Skip older files
☑ Use folder names
□ Open Explorer window

Figure 5-4: Extract Contents of StarOffice File

A simple text or XML editor, such as Notepad, opens one of the extracted files. The file content.xml contains the actual document data. A Find command (Ctrl + F) on the word *Blue* finds the document text (Figure 5-5).

Figure 5-5: Search for *Blue* In the File Content.xml

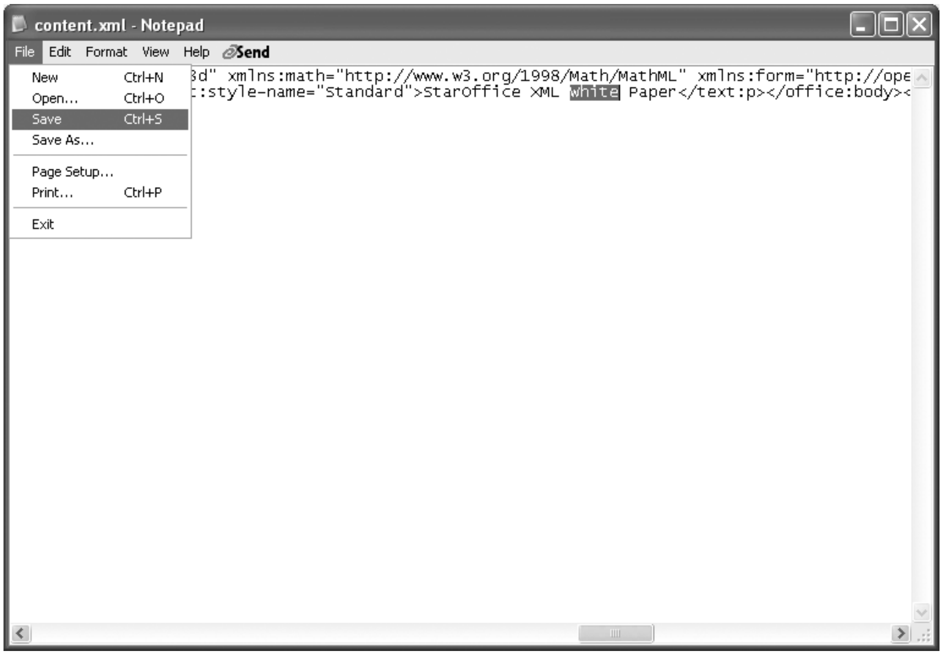


```
noffice.org/2000/dr3d" xmlns:math="http://www.w3.org/1998/Math/MathML" xmlns:form="http://op
e-decls><text:p text:style-name="Standard">StarOffice XML Blue Paper</text:p></office:body><
```

Find what: Blue

Notepad highlights *Blue*, and the word *White* is overtyped. Then, the XML file is saved (Ctrl + S) again (Figure 5-6).

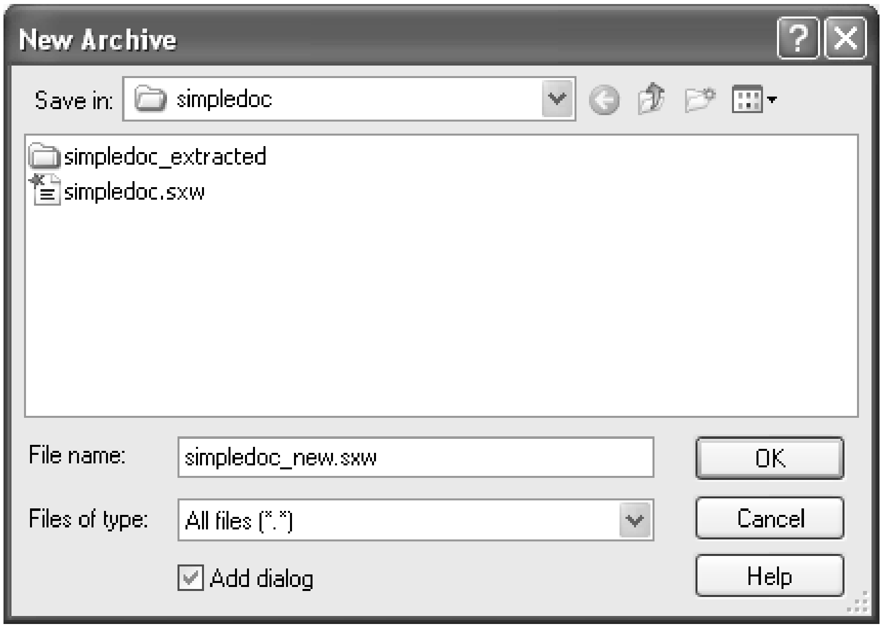
Figure 5-6: Change Blue to White, and Save the File Content.xml Again



```
3d" xmlns:math="http://www.w3.org/1998/Math/MathML" xmlns:form="http://ope
:style-name="Standard">StarOffice XML White Paper</text:p></office:body><
```

Now, a new StarOffice file is created by creating a new, empty ZIP archive. In WinZIP, this can be done by using the New Archive command (Figure 5-7).

Figure 5-7: Create a New Archive In ZIP Tool



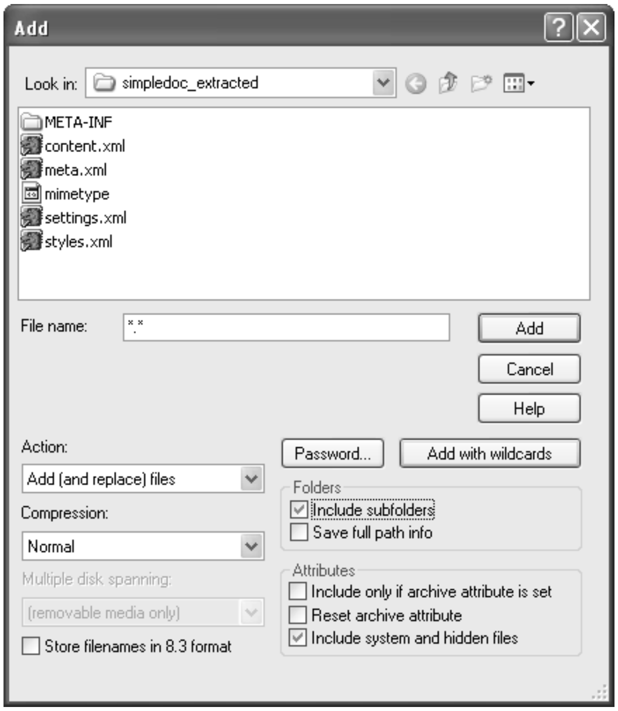
Save in: simpledoc

simpledoc_extracted
simpledoc.sxw

File name: simpledoc_new.sxw
Files of type: All files (*.*)
☑ Add dialog

The new file is not a StarOffice file yet because key components are still missing. Therefore, the XML files from the old StarOffice file must be added, including the one that was just modified. In WinZIP, the directory that contains the files originally extracted is selected, and the *Add with wildcards* button is clicked (Figure 5-8).

Figure 5-8: Add XML Files (Including Content.xml) to the New ZIP Archive



Look in: simpledoc_extracted

META-INF
content.xml
meta.xml
mimetype
settings.xml
styles.xml

File name: *.*

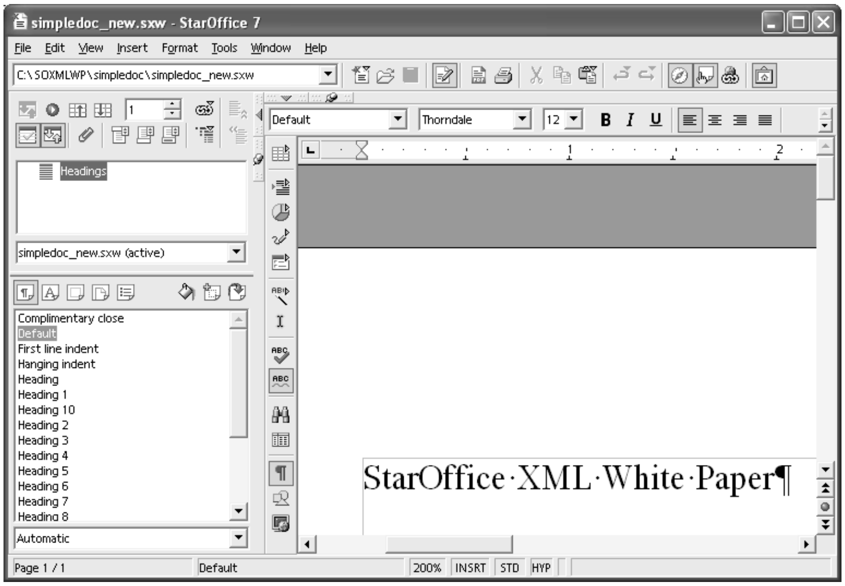
Action: Add (and replace) files
Compression: Normal
Multiple disk spanning: (removable media only)
□ Store filenames in 8.3 format

Folders
☑ Include subfolders
□ Save full path info

Attributes
□ Include only if archive attribute is set
□ Reset archive attribute
☑ Include system and hidden files

A new StarOffice file has been created outside of the StarOffice suite by simply taking content from an old file and inserting it in a new file. The new StarOffice file should now show “StarOffice XML White Paper” as the document content (Figure 5-9).

Figure 5-9: Open New ZIP Archive (New StarOffice File) In the StarOffice Office Suite



StarOffice·XML·White·Paper¶

Removing Change Tracking Information Using a Simple XSLT Transformation

Another interesting use for an XML document file format is to employ scripts to remove confidential information from a document. This can be done on a server before a file is archived in a document management system or sent out via e-mail. Confidential information may include change tracking information or salary data. The following example provides an idea of how this can be achieved with the OpenOffice.org file format. It is important to keep in mind that this example is not complete or perfect. A script that could remove 100 percent of all confidential information would have to be more complex and sophisticated.

The code for the “template.xsl” style sheet that is used to remove the confidential information is:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0"
                xmlns:text="http://openoffice.org/2000/text"
                xmlns:office="http://openoffice.org/2000/office"
                xmlns:xalan="http://xml.apache.org/xalan"
                exclude-result-prefixes="xalan">
<!-- identity copy -->
<xsl:template match="node()|@*" name="copy">
<xsl:copy>
   <xsl:apply-templates select="node()|@*"/>
```

```
</xsl:copy>
</xsl:template>

<!-- remove tracking information -->
<xsl:template match="text:tracked-changes"/>
<xsl:template match="text:change-end"/>
<xsl:template match="text:change-start"/>
<xsl:template match="text:change-region"/>
<xsl:template match="text:change"/>
<!-- remove confidential/private information -->
<xsl:template match="text:p">
   <xsl:choose>
      <xsl:when test="starts-with(., 'Phone')">
         <xsl:comment>
            <xsl:text>Removed phone number!</xsl:text>
         </xsl:comment>
      </xsl:when>

      <xsl:when test="contains(., '@sun.com')">
         <xsl:comment>
            <xsl:text>Removed email address!</xsl:text>
         </xsl:comment>
      </xsl:when>
<xsl:when test="@text:style-name[contains(., 'Confidential')]">
         <xsl:comment>
            <xsl:text>Removed confidential information!</xsl:text>
         </xsl:comment>
      </xsl:when>

      <xsl:otherwise>
         <xsl:call-template name="copy"/>
      </xsl:otherwise>
   </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

To make it easier to use XSLT transformations on StarOffice files, set up a build environment consisting of the Java 2 SDK and Ant build tool. The Java 2 SDK can be found at java.sun.com, and the Ant tool is available from ant.apache.org.

The following “build.xml” script is an Ant script that unzips a StarOffice file, runs the XSLT transformation to remove the confidential data, and creates a new StarOffice file.

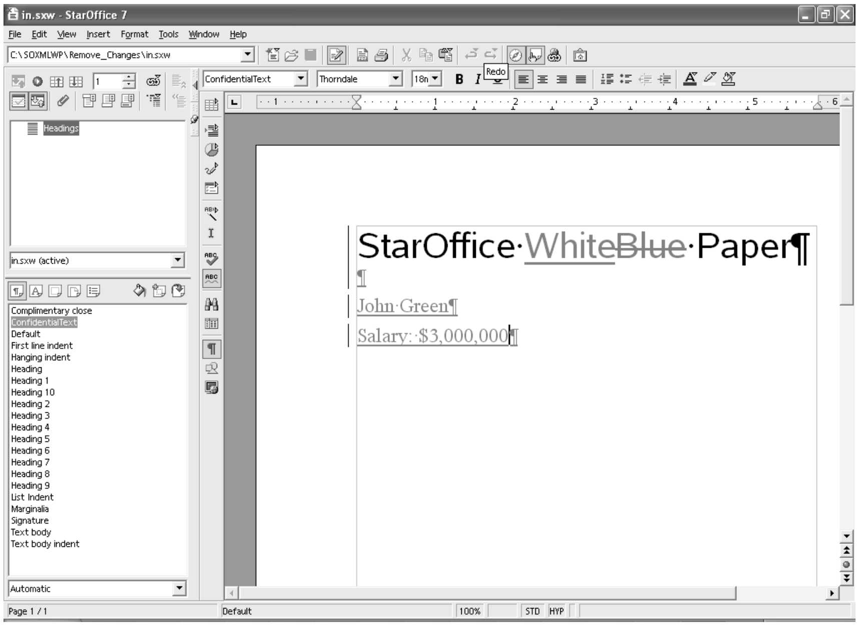
```
<?xml version="1.0" encoding="UTF-8"?>


<!DOCTYPE project>


<project basedir="." default="all" name="Remove tracked changes">


  <!-- readin common defines -->


  <property environment="env"/>
  <property name="soffice.instdir" value="${env.SO_HOME}"/>
  <property name="soffice.dtd" value="${soffice.instdir}/share/dtd/
officedocument/1_0/office.dtd"/>


  <property name="content.style" value="template.xsl"/>
  <property name="content.dest" value="content-out.xml"/>
  <property name="content.saved" value="content-org.xml"/>
  <property name="content.src" value="content.xml"/>
  <property name="so.src" value="in.sxw"/>
  <property name="so.dest" value="out.sxw"/>


  <!-- Extract  content.xml from original StarOffice archive -->
  <target name="extract-content">
<unzip src="${so.src}" dest="." >
     <patternset includes="${content.src}"/>
  </unzip>
  </target>


  <target name="update-content">
  <copy file="${so.src}" tofile="${so.dest}" preservelastmodified="true" />
  <delete file="${content.src}"/>
<copy file="${content.dest}" tofile="${content.src}"/>
  <touch file="${content.src}"/>
  <zip update="true" zipfile="${so.dest}"  >
     <fileset dir="." includes="${content.src}"/>
  </zip>
  </target>


   <!-- Now remove elements -->
  <target name="remove-changes" depends="extract-content">
  <delete file="${content.dest}"/>
  <xslt style= "${content.style}" in="${content.src}" out="${content.dest}">
   <xmlcatalog>
           <dtd publicId="-//OpenOffice.org//DTD OfficeDocument 1.0//EN"
         location="${soffice.dtd}"/>
         </xmlcatalog>
        </xslt>
  </target>

  <!-- pack new content.xml into original StarOffice archive (after copying it)
-->
  <target name="all" depends="remove-changes, update-content">
  </target>

  <target name="clean">
  <delete>
    <fileset dir=".">
      <include name="content*.xml"/>
      <include name="${so.dest}"/>
      <include name="${so.pdf}"/>
      <include name="${so.html}"/>
      <include name="*.class"/>
    </fileset>
  </delete>
</target>

</project>
```

The Ant build script requires a few environment settings that can be defined with a script such as the following:

```
set JAVA_HOME=C:\java\j2sdk1.4.1_02
set SO_HOME=C:\Program Files\StarOffice7
set ANT_HOME=C:\SOXMLWP\ant

set PATH=%JAVA_HOME%\bin;%SO_HOME%\program;%ANT_HOME%\bin;%PATH%
```

Now the build/test environment is in place. The next step is to create a simple file that contains some confidential information, including change tracking data. The confidential salary data uses a special style with the name “ConfidentialText” that marks that paragraph as confidential (Figure 5-10).

Figure 5-10: StarOffice Document With Change Tracking Information



StarOffice·White~~Blue~~·Paper¶

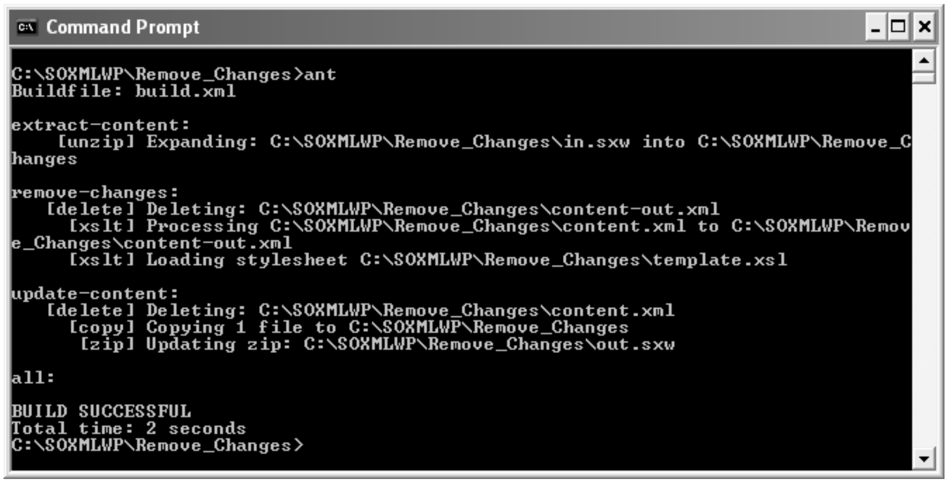
¶

John·Green¶

Salary:·$3,000,000¶

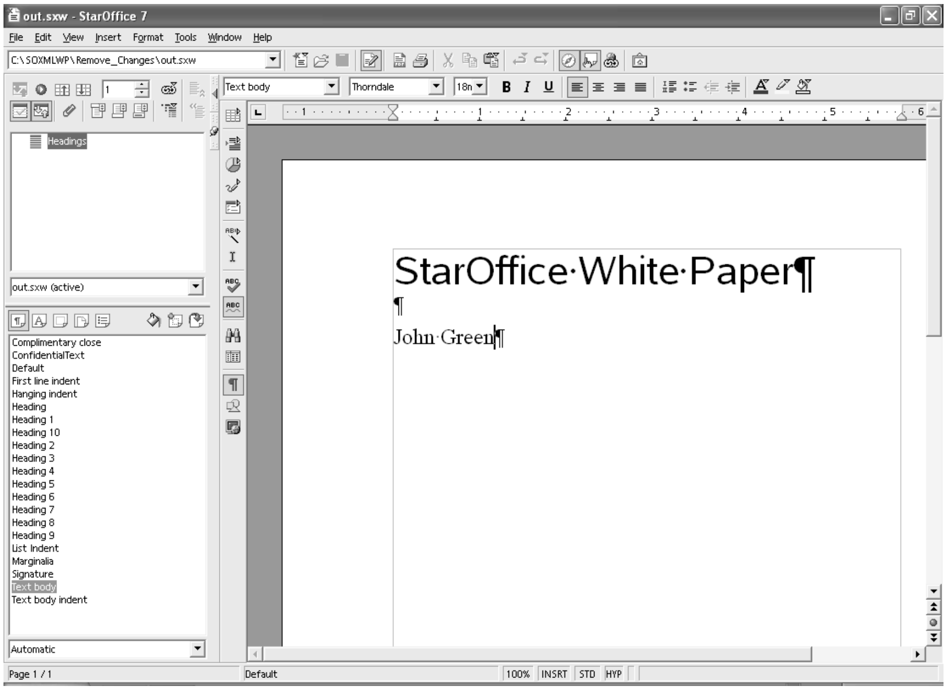
With an Ant environment and build script in place, simply run the Ant command to generate the new StarOffice file (Figure 5-11).

Figure 5-11: Call Ant



```
C:\SOXMLWP\Remove_Changes>ant
Buildfile: build.xml

extract-content:
    [unzip] Expanding: C:\SOXMLWP\Remove_Changes\in.sxw into C:\SOXMLWP\Remove_C
hanges

remove-changes:
   [delete] Deleting: C:\SOXMLWP\Remove_Changes\content-out.xml
     [xslt] Processing C:\SOXMLWP\Remove_Changes\content.xml to C:\SOXMLWP\Remov
e_Changes\content-out.xml
     [xslt] Loading stylesheet C:\SOXMLWP\Remove_Changes\template.xsl

update-content:
   [delete] Deleting: C:\SOXMLWP\Remove_Changes\content.xml
     [copy] Copying 1 file to C:\SOXMLWP\Remove_Changes
      [zip] Updating zip: C:\SOXMLWP\Remove_Changes\out.sxw

all:

BUILD SUCCESSFUL
Total time: 2 seconds
C:\SOXMLWP\Remove_Changes>
```

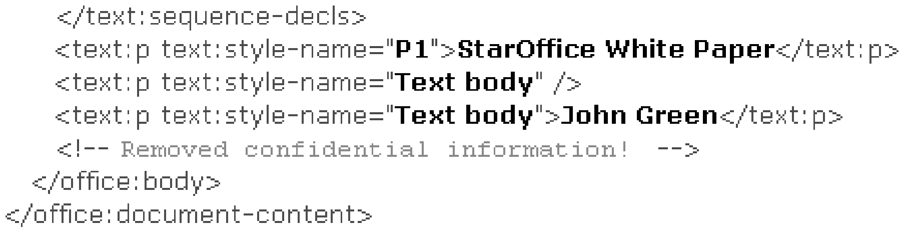
The new file no longer contains the confidential information (Figure 5-12).



StarOffice·White·Paper¶
¶
John·Green¶

Figure 5-12: StarOffice Document With Change Tracking Information Removed

In an XML editor, the content.xml file shows that some confidential information was removed. The comment was inserted by the XSLT transformation (Figure 5-13).



```
    </text:sequence-decls>
    <text:p text:style-name="P1">StarOffice White Paper</text:p>
    <text:p text:style-name="Text body" />
    <text:p text:style-name="Text body">John Green</text:p>
    <!-- Removed confidential information!  -->
  </office:body>
</office:document-content>
```

Figure 5-13: New Content.xml File In XML Editor

Converting a Simple HTML Table into a Spreadsheet

The following XSLT transformation takes a simple HTML table and converts it into a content file for a spreadsheet table so that the XSLT transformation can be used as an XML import filter for simple HTML tables. The transformation is provided by J. David Eisenberg for the purpose of this white paper. He is the author of the upcoming book, *“OpenOffice.org XML Essentials—Using OpenOffice.org’s XML Data Format,”* published by O’Reilly & Associates.

The XSLT definition must be copied to a file with the name “table_to_sxc.xsl,” so that it can be used with the XML filter tool.

```
<?xml version="1.0"?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:office="http://openoffice.org/2000/office"
```

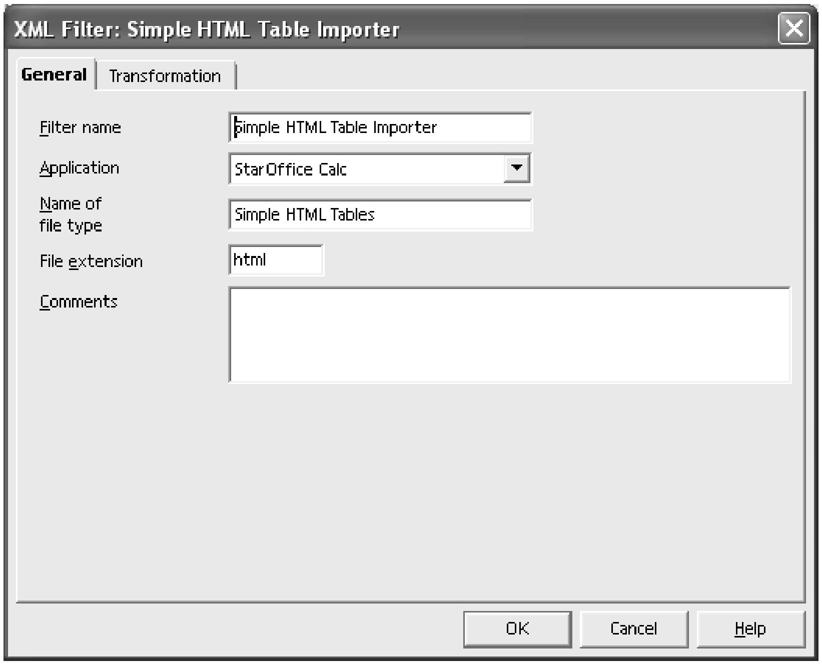
```
xmlns:style="http://openoffice.org/2000/style"
xmlns:text="http://openoffice.org/2000/text"
xmlns:table="http://openoffice.org/2000/table"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:number="http://openoffice.org/2000/datastyle"
xmlns:script="http://openoffice.org/2000/script"
>
<xsl:output method="xml"
doctype-public="-//OpenOffice.org//DTD OfficeDocument 1.0//EN"
doctype-system="office.dtd"/>

<xsl:template match="/">
<office:document-content office:class="spreadsheet" office:version="1.0">

<office:script/>

<office:automatic-styles>

<!-- Column style -->
<style:style style:name="co1" style:family="table-column">
<style:properties fo:break-before="auto"
style:column-width="3.5cm"/>
</style:style>

<!-- Let all the rows have optimal height -->
<style:style style:name="ro1" style:family="table-row">
<style:properties fo:break-before="auto"
style:use-optimal-row-height="true"/>
</style:style>

<!-- The table references a master-page which doesn't exist,
but that doesn't bother OpenOffice.org -->
<style:style style:name="ta1" style:family="table"
style:master-page-name="TAB_Sheet1">
<style:properties table:display="true"/>
</style:style>

<!-- style for heading cells -->
<style:style style:name="heading" style:family="table-cell"
style:parent-style-name="Default">
```

```
<style:properties fo:text-align="center"
fo:font-weight="bold"/>
</style:style>

<!-- style for raw data cells (just use OpenOffice.org defaults) -->
<style:style style:name="normal" style:family="table-cell"
style:parent-style-name="Default"/>
</office:automatic-styles>

<office:body>
<xsl:apply-templates select="table"/>
</office:body>
</office:document-content>
</xsl:template>

<xsl:template match="table">
<!-- start the spreadsheet -->
<table:table table:name="Table{position()}" table:style-name="ta1">
<table:table-column table:style-name="co1"
table:default-cell-style-name="normal"
table:number-columns-repeated="{count(tr[1]/th)}"
/>

<xsl:apply-templates select="tr"/>

</table:table>
</xsl:template>

<xsl:template match="tr">
<table:table-row table:style-name="ro1">
<xsl:apply-templates select="th|td"/>
</table:table-row>
</xsl:template>

<xsl:template match="th">
<table:table-cell table:style-name="heading">
<text:p><xsl:value-of select="."/></text:p>
</table:table-cell>
</xsl:template>
```

```
<xsl:template match="td">
<table:table-cell>
<text:p><xsl:value-of select="."/></text:p>
</table:table-cell>
</xsl:template>

</xsl:stylesheet>
```

In the StarOffice filter tool found under “Tools — XML Filter Settings ...,” configure the new simple XML filter as shown in the next three screen shots (Figure 5-14, Figure 5-15, and Figure 5-16).

Figure 5-14: StarOffice XML Filter Tool – General Tab



XML Filter: Simple HTML Table Importer

Filter name: Simple HTML Table Importer

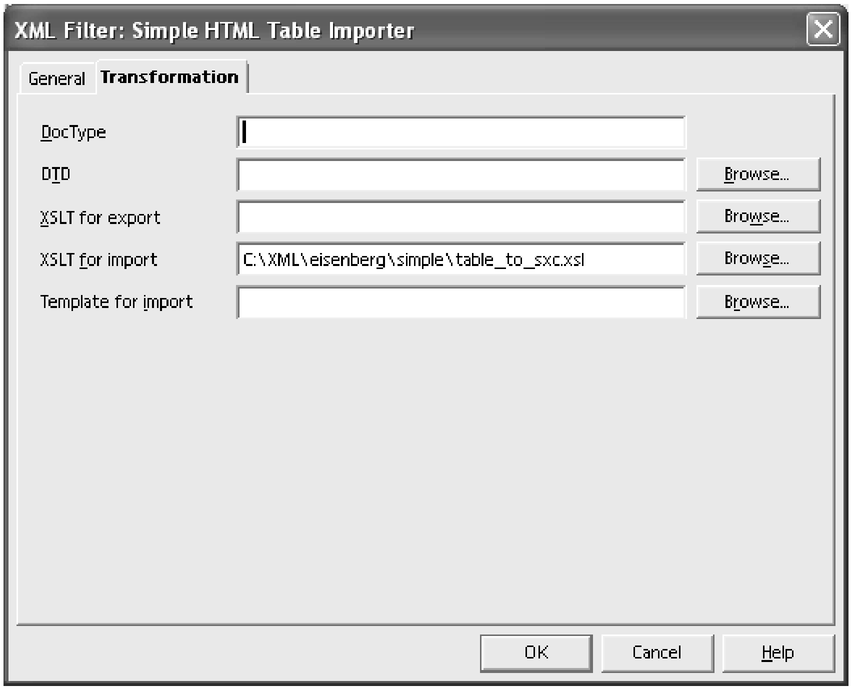
Application: StarOffice Calc

Name of file type: Simple HTML Tables

File extension: html

Comments:

The filter name and the name of the filter type can be chosen freely. In the “Transformation” tab, browse for the import XSLT file that was created in Figure 5-14.



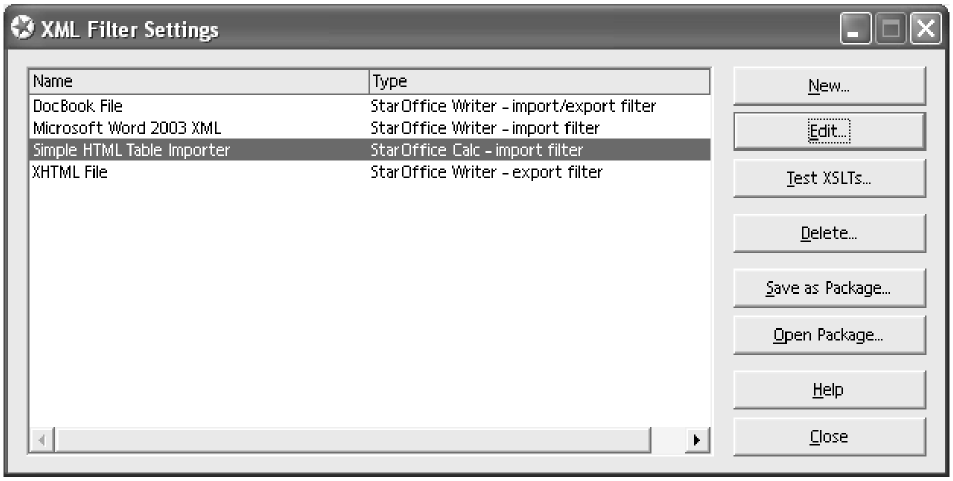
XML Filter: Simple HTML Table Importer

General | Transformation

DocType	
DTD	
XSLT for export	
XSLT for import	C:\XML\eisenberg\simple\table_to_sxc.xsl
Template for import	

Figure 5-15: StarOffice XML Filter Tool – Transformation Tab

After clicking "OK," the new XML filter should appear in the main XML filter dialog (Figure 5-16).



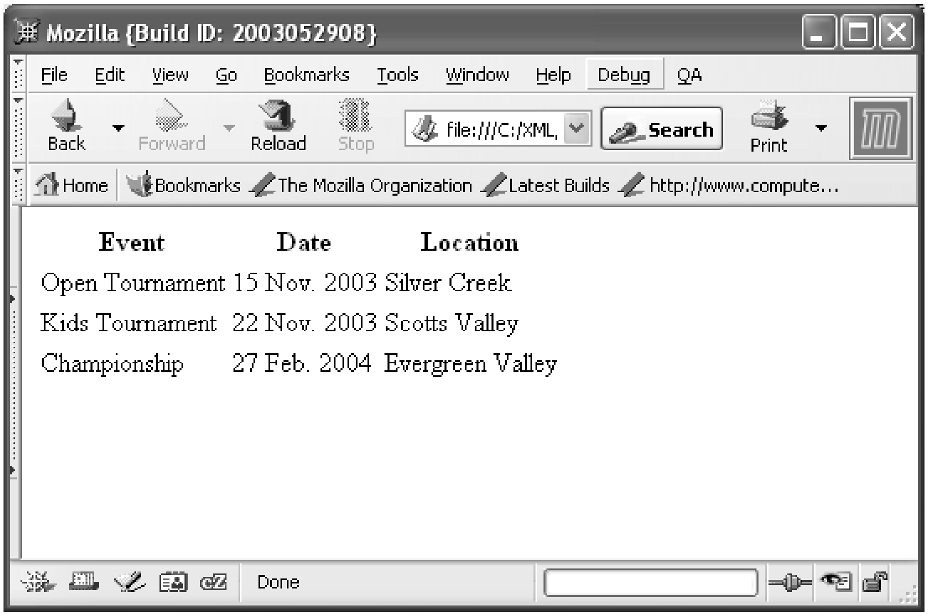
XML Filter Settings

Name	Type
DocBook File	StarOffice Writer - import/export filter
Microsoft Word 2003 XML	StarOffice Writer - import filter
Simple HTML Table Importer	StarOffice Calc - import filter
XHTML File	StarOffice Writer - export filter

Figure 5-16: StarOffice XML Filter Tool – XML Filter Settings

For a test, use a simple HTML table that looks like Figure 5-17:

Figure 5-17: Simple HTML Table In the Mozilla™ Browser



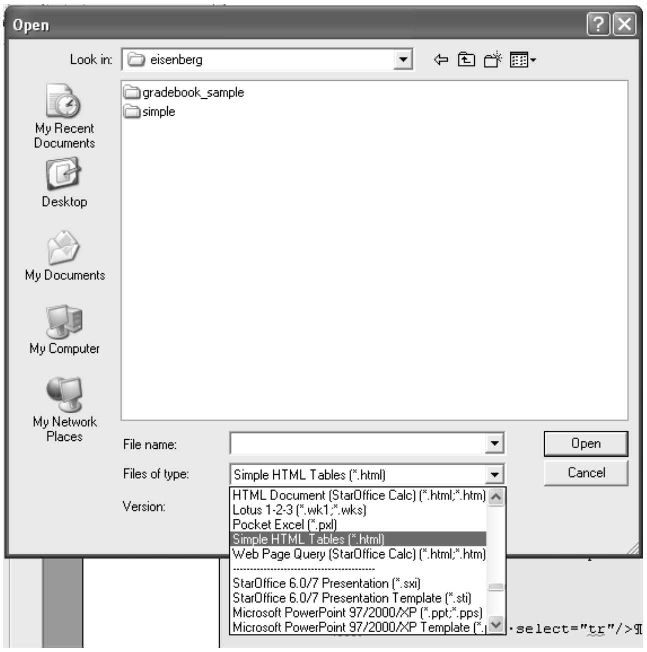
Event	Date	Location
Open Tournament	15 Nov. 2003	Silver Creek
Kids Tournament	22 Nov. 2003	Scotts Valley
Championship	27 Feb. 2004	Evergreen Valley

The source code for the HTML table is:

```
<html>
<head>
</head>
<body>
<table>
  <tbody>
    <tr>
      <th>Event</th>
      <th>Date</th>
      <th>Location</th>
    </tr>
    <tr>
      <td>Open Tournament</td>
      <td>15 Nov. 2003</td>
      <td>Silver Creek</td>
    </tr>
    <tr>
      <td>Kids Tournament</td>
      <td>22 Nov. 2003</td>
      <td>Scotts Valley</td>
    </tr>
    <tr>
```

```
      <td>Championship</td>
      <td>27 Feb. 2004</td>
      <td>Evergreen Valley</td>
    </tr>
  </tbody>
</table>
</body>
</html>
```

In the StarOffice suite, go to “File – Open” and select the new filter that was just defined. Then browse to the file containing the HTML file, and click the Open button (Figure 5-18).


Open
Look in:
eisenberg
gradebook_sample
simple
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
File name:
Files of type:
Version:
Open
Cancel
HTML Document (StarOffice Calc) (*.html;*.htm)
Lotus 1-2-3 (*.wk1;*.wks)
Pocket Excel (*.pxl)
Simple HTML Tables (*.html)
Web Page Query (StarOffice Calc) (*.html;*.htm)
StarOffice 6.0/7 Presentation (*.sxi)
StarOffice 6.0/7 Presentation Template (*.sti)
Microsoft PowerPoint 97/2000/XP (*.ppt;*.pps)
Microsoft PowerPoint 97/2000/XP Template (*.
·select="tr"/>¶

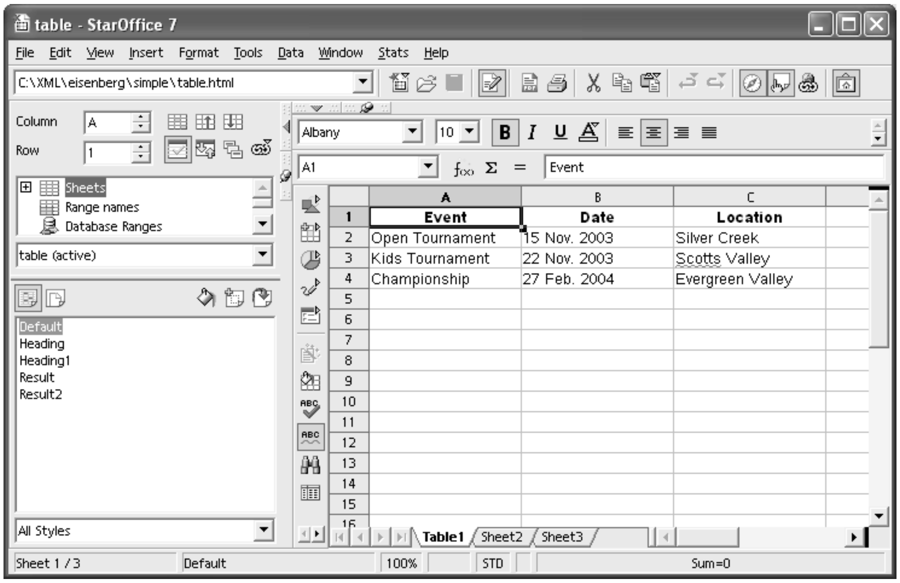
Look in: eisenberg

Files of type: Simple HTML Tables (*.html)

Figure 5-18: Open File Using the New XML Filter

A spreadsheet that contains the contents of the HTML table can now be opened in StarOffice Calc (Figure 5-19).

Figure 5-19: HTML Table In StarOffice Calc



C:\XML\eisenberg\simple\table.html

Column: A
Row: 1

table (active)

Albany 10

A1 = Event

Default
Heading
Heading1
Result
Result2

	A	B	C
1	**Event**	**Date**	**Location**
2	Open Tournament	15 Nov. 2003	Silver Creek
3	Kids Tournament	22 Nov. 2003	Scotts Valley
4	Championship	27 Feb. 2004	Evergreen Valley

Sheet 1 / 3 Default 100% STD Sum=0

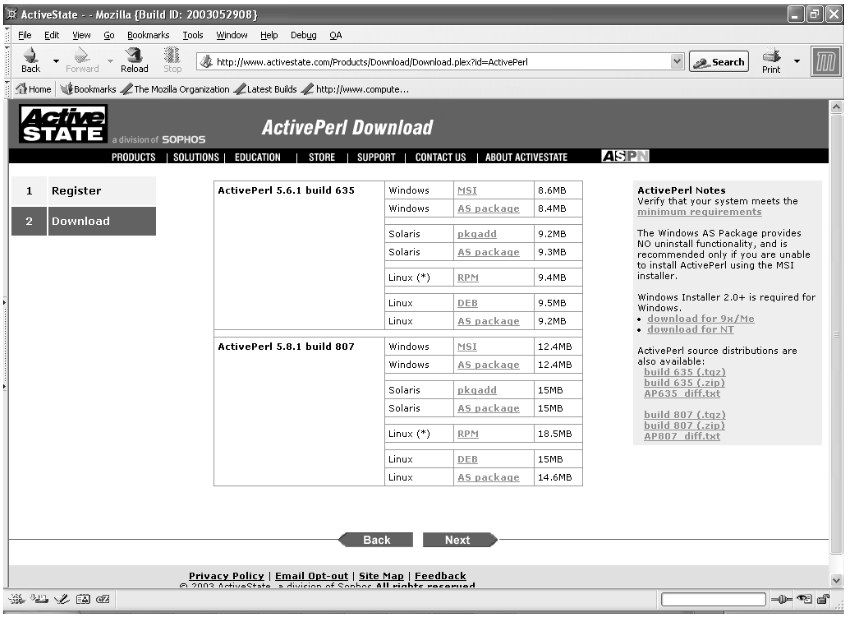
An article that covers a related topic, *"Opening Open Formats with XSLT,"* can be found at xml.com/lpt/a/2004/02/04/tr-xml.html. It explains how to generate speaker notes from an existing presentation file using an XSLT transformation.

Setting Document Fields Using Perl

A French company, Genicorp (genicorp.fr), has created a useful library for the Perl scripting language. The name of the Perl extension is OpenOffice::OODoc. This library makes it very easy to access and modify StarOffice and OpenOffice.org files.

The example shown here is included in the set of samples that Genicorp provides on its Web site. Currently, most documentation for OpenOffice::OODoc is available only in French, but an English translation should become available soon.

To use the Perl library, it is necessary to install a Perl runtime environment and a few additional Perl modules. A Perl environment that works well on Microsoft Windows systems is ActivePerl from ActiveState, which can be downloaded from: activestate.com/Products/Download/Download.plex?id=ActivePerl (Figure 5-20).


ActiveState - - Mozilla {Build ID: 2003052908}
File Edit View Go Bookmarks Tools Window Help Debug QA
Back Forward Reload Stop
http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl
Search
Print
Home Bookmarks The Mozilla Organization Latest Builds http://www.compute...
Active STATE a division of SOPHOS
PRODUCTS | SOLUTIONS | EDUCATION | STORE | SUPPORT | CONTACT US | ABOUT ACTIVESTATE
ASPN
1 Register
2 Download
Back
Next

ActivePerl Download

ActivePerl 5.6.1 build 635	Windows	MSI	8.6MB
	Windows	AS package	8.4MB
	Solaris	pkgadd	9.2MB
	Solaris	AS package	9.3MB
	Linux (*)	RPM	9.4MB
	Linux	DEB	9.5MB
	Linux	AS package	9.2MB
ActivePerl 5.8.1 build 807	Windows	MSI	12.4MB
	Windows	AS package	12.4MB
	Solaris	pkgadd	15MB
	Solaris	AS package	15MB
	Linux (*)	RPM	18.5MB
	Linux	DEB	15MB
	Linux	AS package	14.6MB

ActivePerl Notes
Verify that your system meets the minimum requirements

The Windows AS Package provides NO uninstall functionality, and is recommended only if you are unable to install ActivePerl using the MSI installer.

Windows Installer 2.0+ is required for Windows.
- download for 9x/Me
- download for NT

ActivePerl source distributions are also available:
build 635 (.tgz)
build 635 (.zip)
AP635_diff.txt

build 807 (.tgz)
build 807 (.zip)
AP807_diff.txt

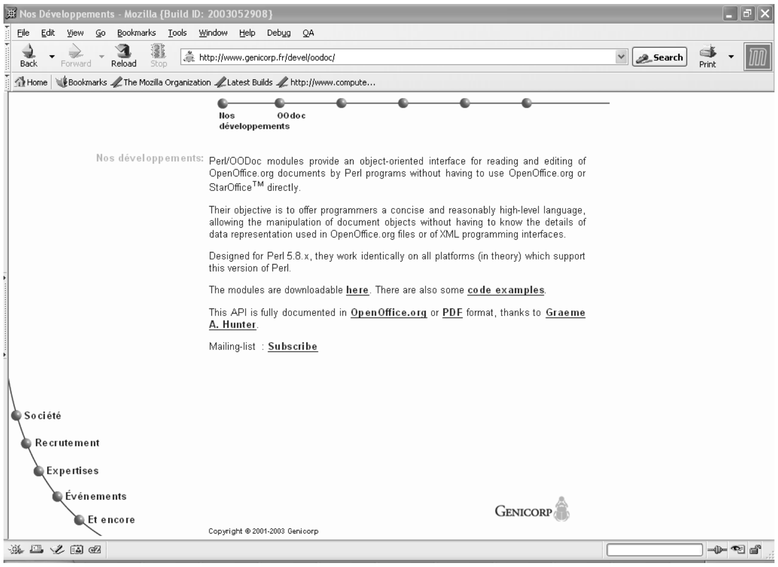
Privacy Policy | Email Opt-out | Site Map | Feedback
© 2003 ActiveState, a division of Sophos All rights reserved

Figure 5-20: IActivePerl Download Page

The 5.8.x version of ActivePerl works well.

The next thing that must be downloaded is the actual Perl extension. This can be found at genicorp.fr/devel/oodoc/ (Figure 5-21). Some download links can also be found there:

- Perl Module
  genicorp.fr/devel/oodoc/oodoc_modules.zip
- Manual
  genicorp.fr/devel/oodoc/oodoc_man_en.pdf
- Sample Code
  genicorp.fr/devel/oodoc/oodoc_exemples.zip


Nos Développements - Mozilla {Build ID: 2003052908}
File Edit View Go Bookmarks Tools Window Help Debug QA
Back Forward Reload Stop
http://www.genicorp.fr/devel/oodoc/
Search
Print
Home Bookmarks The Mozilla Organization Latest Builds http://www.compute...
Nos développements
OOdoc
Société
Recrutement
Expertises
Événements
Et encore
GENICORP

Nos développements: Perl/OODoc modules provide an object-oriented interface for reading and editing of OpenOffice.org documents by Perl programs without having to use OpenOffice.org or StarOffice™ directly.

Their objective is to offer programmers a concise and reasonably high-level language, allowing the manipulation of document objects without having to know the details of data representation used in OpenOffice.org files or of XML programming interfaces.

Designed for Perl 5.8.x, they work identically on all platforms (in theory) which support this version of Perl.

The modules are downloadable **here**. There are also some **code examples**.

This API is fully documented in **OpenOffice.org** or **PDF** format, thanks to **Graeme A. Hunter**.

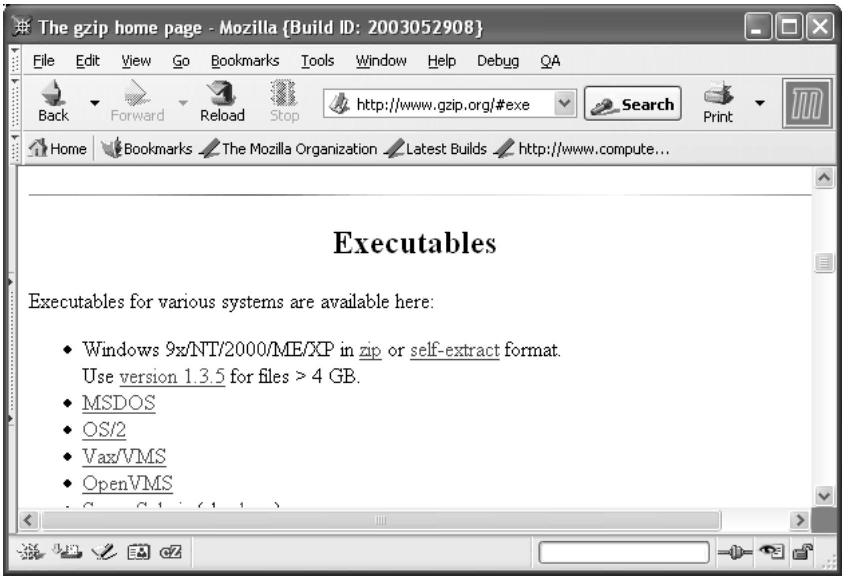
Mailing-list : **Subscribe**

Copyright © 2001-2003 Genicorp

Figure 5-21: Genicorp Download Page

The Perl runtime environment ActivePerl requires a tool, such as GZIP, for the download and installation of additional Perl modules. GZIP for Microsoft Windows can be downloaded at www.gzip.org/#exe (Figure 5-22).

Figure 5-22: GZIP Executables Download Page



Executables

Executables for various systems are available here:

- Windows 9x/NT/2000/ME/XP in zip or self-extract format.
  Use version 1.3.5 for files > 4 GB.
- MSDOS
- OS/2
- Vax/VMS
- OpenVMS

Now that all the necessary files have been downloaded, the configuration process can begin. First, if the ZIP package is chosen instead of the MSI file, the Perl environment must be installed. Unzip the ActivePerl ZIP package, and run the “Installer.bat” file. The installation process should look similar to this:

```
C:\Genicorp\ActivePerl-5.8.1.807-MSWin32-x86>Installer.bat
    Welcome to ActivePerl.


    This installer can install ActivePerl in any location of your choice.
    You do not need Administrator privileges.  However, please make sure
    that you have write access to this location.


Enter top level directory for install [c:\Perl]:


    The typical ActivePerl software installation requires 75 megabytes.
    Please make sure enough free space is available before continuing.


    ActivePerl 807 will be installed into 'c:\Perl'
Proceed? [y]


    If you have a development environment (e.g. Visual Studio) that you
    wish to use with Perl, you should ensure that your environment (e.g.
    %LIB% and %INCLUDE%) is set before installing, for example, by running
    vcvars32.bat first.
```

```
Proceed? [y]

Create shortcuts to the HTML documentation? [y]

Add the Perl/bin directory to the PATH? [y]

    Copying files...
2862 File(s) copied
    Finished copying files...
Configuring Perl installation at c:\Perl

[...]

Building HTML documentation, please wait...

This simplified installation program currently does *not*:

    o set up MSWin32 file associations
    o configure Perl for use with a Web Server

Refer to your Operating System and/or Web Server documentation for
details on how to to perform these modifications.

Thank you for installing ActivePerl!

Press return to exit.
C:\Genicorp\ActivePerl-5.8.1.807-MSWin32-x86>
```

Now that the Perl runtime environment is installed, the additional required libraries must be downloaded and installed. This can be done with the following command:

```
C:\>perl -MCPAN -e shell
```

This shell downloads the additional CPAN modules. However, when the the shell is invoked for the first time, a configuration process starts. In most cases the defaults should work. The location of the GZIP tool, proxy, and country information must be entered at this point.

After the configuration process is complete, the two additional libraries can be downloaded with the following commands:

```
cpan> install Archive::Zip
```

and

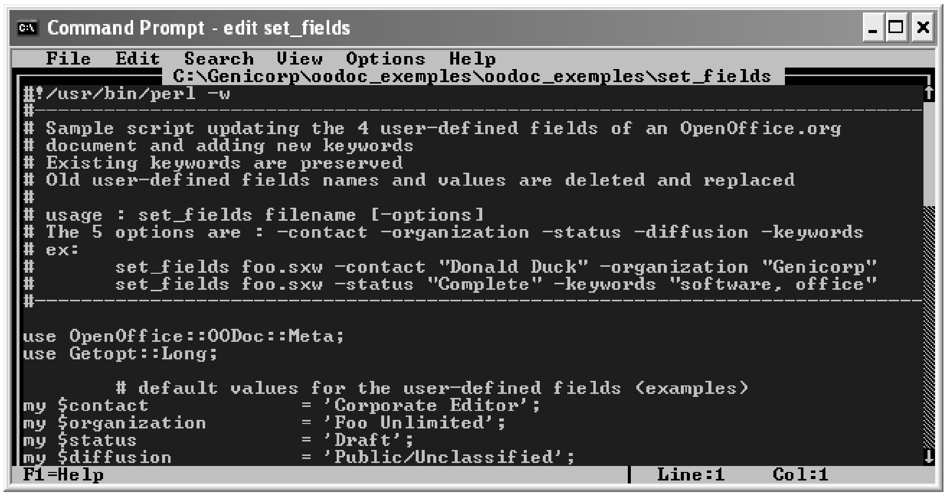
```
cpan> install XML::XPath
```

A sample configuration and installation process is provided in the appendix of this white paper.

As a final configuration step, the contents of the OpenOffice::OODoc ZIP archive (directory “OpenOffice” plus included files) must be extracted and copied into the “lib” directory of the Perl runtime environment. To play with the OODoc samples, the oodoc_exemples.zip file also must be unzipped.

This exercise looks at the included set_fields example, which is able to set metadata information in existing StarOffice files. The usage of the set_fields example is explained at the beginning of the script (Figure 5-23).

Figure 5-23: Set_fields Perl Script

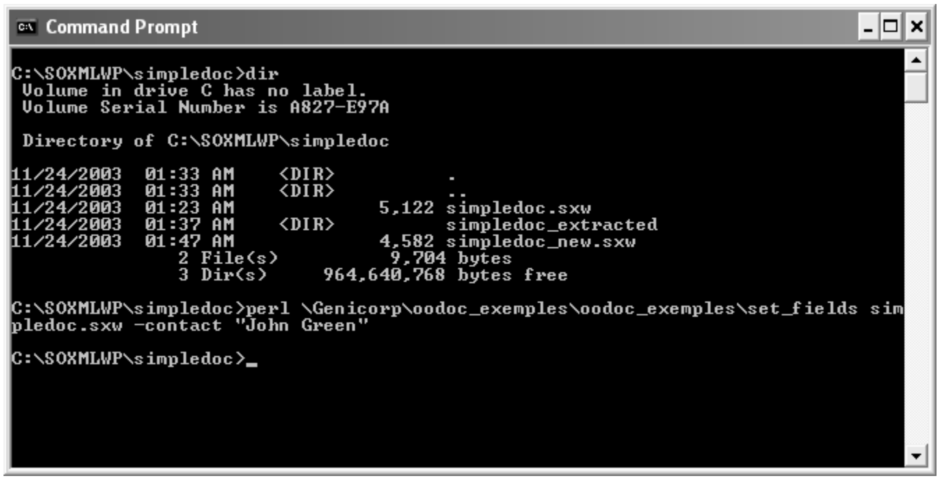

Command Prompt - edit set_fields
File Edit Search View Options Help
C:\Genicorp\oodoc_exemples\oodoc_exemples\set_fields
F1=Help
Line:1 Col:1

```
#!/usr/bin/perl -w
#-----------------------------------------------------------------------------
# Sample script updating the 4 user-defined fields of an OpenOffice.org
# document and adding new keywords
# Existing keywords are preserved
# Old user-defined fields names and values are deleted and replaced
#
# usage : set_fields filename [-options]
# The 5 options are : -contact -organization -status -diffusion -keywords
# ex:
#       set_fields foo.sxw -contact "Donald Duck" -organization "Genicorp"
#       set_fields foo.sxw -status "Complete" -keywords "software, office"
#-----------------------------------------------------------------------------

use OpenOffice::OODoc::Meta;
use Getopt::Long;

        # default values for the user-defined fields (examples)
my $contact             = 'Corporate Editor';
my $organization        = 'Foo Unlimited';
my $status              = 'Draft';
my $diffusion           = 'Public/Unclassified';
```

Basically, the script must be called as shown in Figure 5-24. The command line is:

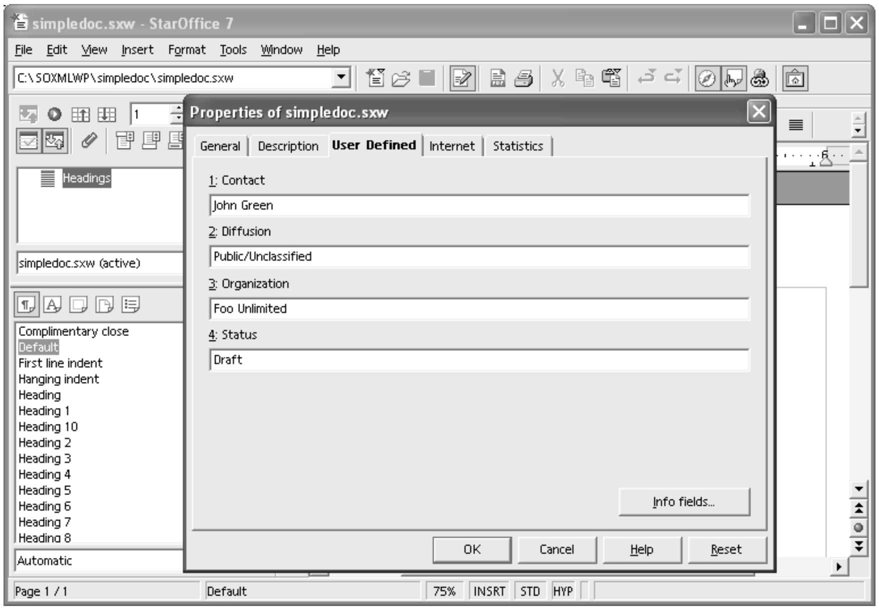
```
perl set_fields simpledoc.sxw -contact “John Green”
```

Figure 5-24: Calling Set_fields Script



```
C:\SOXMLWP\simpledoc>dir
 Volume in drive C has no label.
 Volume Serial Number is A827-E97A

 Directory of C:\SOXMLWP\simpledoc

11/24/2003  01:33 AM    <DIR>          .
11/24/2003  01:33 AM    <DIR>          ..
11/24/2003  01:23 AM             5,122 simpledoc.sxw
11/24/2003  01:37 AM    <DIR>          simpledoc_extracted
11/24/2003  01:47 AM             4,582 simpledoc_new.sxw
               2 File(s)          9,704 bytes
               3 Dir(s)     964,640,768 bytes free

C:\SOXMLWP\simpledoc>perl \Genicorp\oodoc_exemples\oodoc_exemples\set_fields sim
pledoc.sxw -contact "John Green"

C:\SOXMLWP\simpledoc>_
```

As a result, the Contact property of the document is set to “John Green” (Figure 5-25).



1: Contact
John Green

2: Diffusion
Public/Unclassified

3: Organization
Foo Unlimited

4: Status
Draft

Figure 5-25: The Contact Name Properties Field Is Set to "John Green."

Other interesting samples available for OpenOffice::OODoc are scripts that can search for key words in paragraphs and copy them into a new file, or execute a shell command if it finds a specific key word in a document.

Chapter 6

Resources

OpenOffice.org XML Project

- xml.openoffice.org

OpenOffice.org Developer Page

- development.openoffice.org

StarOffice Office Suite

- sun.com/staroffice

Sun Desktop Developer Page

- developer.sun.com/techtopics/desktop

XSLT

- w3.org/TR/xslt

Dublin Core Metadata Initiative

- dublincore.org

XSL-FO

- w3.org/TR/xsl

XLink

- w3.org/TR/xlink

SVG

- w3.org/TR/SVG

HTML

- w3.org/MarkUp

ebXML

- ebxml.org

UBL

- oasis-open.org/committees/ubl

LGPL/SISSL

- openoffice.org/license.html

Java Community Process™

- jcp.org

Chapter 7

Appendix

Comprehensive PERL Archive Network (CPAN) Module Installation Procedure Example

```
C:\>perl -MCPAN -e shell

C:\Perl\lib\CPAN\Config.pm initialized.

CPAN is the world wide archive of perl resources. It consists of about
100 sites that all replicate the same contents all around the globe.
Many countries have at least one CPAN site already. The resources
found on CPAN are easily accessible with the CPAN.pm module. If you
want to use CPAN.pm, you have to configure it properly.

If you do not want to enter a dialog now, you can answer 'no' to this
question and I'll try to autoconfigure. (Note: you can revisit this
dialog anytime later by typing 'o conf init' at the cpan prompt.)

Are you ready for manual configuration? [yes]
```

```
The following questions are intended to help you with the
configuration. The CPAN module needs a directory of its own to cache
important index files and maybe keep a temporary mirror of CPAN files.
This may be a site-wide directory or a personal directory.

I see you already have a directory
    \.cpan
Shall we use it as the general CPAN build and cache directory?

CPAN build and cache directory? [\.cpan]

If you want, I can keep the source files after a build in the cpan
home directory. If you choose so then future builds will take the
files from there. If you don't want to keep them, answer 0 to the
next question.
How big should the disk cache be for keeping the build directories
with all the intermediate files?

Cache size for build directory (in MB)? [10]

By default, each time the CPAN module is started, cache scanning
is performed to keep the cache size in sync. To prevent this,
disable the cache scanning with 'never'.

Perform cache scanning (atstart or never)? [atstart]


To considerably speed up the initial CPAN shell startup, it is
possible to use Storable to create a cache of metadata. If Storable
is not available, the normal index mechanism will be used.

Cache metadata (yes/no)? [yes]

The next option deals with the charset your terminal supports. In
general CPAN is English speaking territory, thus the charset does not
matter much, but some of the aliens out there who upload their
software to CPAN bear names that are outside the ASCII range. If your
terminal supports UTF-8, you say no to the next question, if it
supports ISO-8859-1 (also known as LATIN1) then you say yes, and if it
supports neither nor, your answer does not matter, you will not be
```

```
able to read the names of some authors anyway. If you answer no, names
will be output in UTF-8.

Your terminal expects ISO-8859-1 (yes/no)? [yes]

If you have one of the readline packages (Term::ReadLine::Perl,
Term::ReadLine::Gnu, possibly others) installed, the interactive CPAN
shell will have history support. The next two questions deal with the
filename of the history file and with its size. If you do not want to
set this variable, please hit SPACE RETURN to the following question.

File to save your history? [\.cpan\histfile]
Number of lines to save? [100]

The CPAN module can detect when a module that which you are trying to
build depends on prerequisites. If this happens, it can build the
prerequisites for you automatically ('follow'), ask you for
confirmation ('ask'), or just ignore them ('ignore'). Please set your
policy to one of the three values.

Policy on building prerequisites (follow, ask or ignore)? [ask]

The CPAN module will need a few external programs to work properly.
Please correct me, if I guess the wrong path for a program. Don't
panic if you do not have some of them, just press ENTER for those. To
disable the use of a download program, you can type a space followed
by ENTER.

Warning: gzip not found in PATH
Where is your gzip program? [] C:\Genicorp\gzip124xN
Warning: tar not found in PATH
Where is your tar program? []
Warning: unzip not found in PATH
Where is your unzip program? []
Warning: nmake not found in PATH
Where is your make program? []
Warning: lynx not found in PATH
Where is your lynx program? []
Warning: wget not found in PATH
Where is your wget program? []
```

```
Warning: ncftpget not found in PATH
Where is your ncftpget program? []
Warning: ncftp not found in PATH
Where is your ncftp program? []
Where is your ftp program? [C:\WINDOWS\system32\ftp.EXE]
Warning: gpg not found in PATH
Where is your gpg program? []
What is your favorite pager program? [C:\WINDOWS\system32\more.COM]
What is your favorite shell?
Every Makefile.PL is run by perl in a separate process. Likewise we
run 'make' and 'make install' in processes. If you have any
parameters (e.g. PREFIX, LIB, UNINST or the like) you want to pass
to the calls, please specify them here.

If you don't understand this question, just press ENTER.

Parameters for the 'perl Makefile.PL' command?
Typical frequently used settings:

    PREFIX=~/perl       non-root users (please see manual for more hints)

Your choice:  []
Parameters for the 'make' command?
Typical frequently used setting:

    -j3              dual processor system

Your choice:  []
Parameters for the 'make install' command?
Typical frequently used setting:

    UNINST=1         to always uninstall potentially conflicting files

Your choice:  []

Sometimes you may wish to leave the processes run by CPAN alone
without caring about them. As sometimes the Makefile.PL contains
question you're expected to answer, you can set a timer that will
kill a 'perl Makefile.PL' process after the specified time in seconds.
```

```
If you set this value to 0, these processes will wait forever. This is
the default and recommended setting.

Timeout for inactivity during Makefile.PL? [0]
If you're accessing the net via proxies, you can specify them in the
CPAN configuration or via environment variables. The variable in
the $CPAN::Config takes precedence.

Your ftp_proxy?
Your http_proxy? [129.149.246.4:8080]
Your no_proxy?

If your proxy is an authenticating proxy, you can store your username
permanently. If you do not want that, just press RETURN. You will then
be asked for your username in every future session.

Your proxy user id?
Found \.cpan\sources\MIRRORED.BY as of Sun Nov 23 00:08:58 2003

I'd use that as a database of CPAN sites. If that is OK for you,
please answer 'y', but if you want me to get a new database now,
please answer 'n' to the following question.

Shall I use the local database in \.cpan\sources\MIRRORED.BY? [y]

Now we need to know where your favorite CPAN sites are located. Push
a few sites onto the array (just in case the first on the array won't
work). If you are mirroring CPAN to your local workstation, specify a
file: URL.

First, pick a nearby continent and country (you can pick several of
each, separated by spaces, or none if you just want to keep your
existing selections). Then, you will be presented with a list of URLs
of CPAN mirrors in the countries you selected, along with previously
selected URLs. Select some of those URLs, or just keep the old list.
Finally, you will be prompted for any extra URLs -- file:, ftp:, or
http: -- that host a CPAN mirror.

(1) Africa
(2) Asia
```

```
(3) Central America
(4) Europe
(5) North America
(6) Oceania
(7) South America
Select your continent (or several nearby continents) [] 5
Sorry! since you don't have any existing picks, you must make a
geographic selection.

(1) Canada
(2) Mexico
(3) United States
Select your country (or several nearby countries) [] 3
Sorry! since you don't have any existing picks, you must make a
geographic selection.

(1) ftp://archive.progeny.com/CPAN/
(2) ftp://carroll.cac.psu.edu/pub/CPAN/
(3) ftp://cpan-du.viaverio.com/pub/CPAN/
(4) ftp://cpan-sj.viaverio.com/pub/CPAN/
(5) ftp://cpan.calvin.edu/pub/CPAN
(6) ftp://cpan.cse.msu.edu/
(7) ftp://cpan.digisle.net/pub/CPAN
(8) ftp://cpan.erlbaum.net/
(9) ftp://cpan.llarian.net/pub/CPAN/
(10) ftp://cpan.nas.nasa.gov/pub/perl/CPAN/
(11) ftp://cpan.netnitco.net/pub/mirrors/CPAN/
(12) ftp://cpan.pair.com/pub/CPAN/
(13) ftp://cpan.teleglobe.net/pub/CPAN
(14) ftp://cpan.thepirtgroup.com/
(15) ftp://cpan.uky.edu/pub/CPAN/
(16) ftp://cpan.valueclick.com/pub/CPAN/
43 more items, hit SPACE RETURN to show them
Select as many URLs as you like (by number),
put them on one line, separated by blanks, e.g. '1 4 5' [] 1 5 6

Enter another URL or RETURN to quit: []
New set of picks:
  ftp://archive.progeny.com/CPAN/
  ftp://cpan.calvin.edu/pub/CPAN
  ftp://cpan.cse.msu.edu/
commit: wrote C:\Perl\lib\CPAN\Config.pm
Terminal does not support AddHistory.

cpan shell -- CPAN exploration and modules installation (v1.7601)
ReadLine support available (try 'install Bundle::CPAN')

cpan> install Archive::Zip

[...]

cpan> install XML::XPath
```

Learn More

Get the inside story on the trends and technologies shaping the future of computing by signing up for the Sun Inner Circle program. You'll receive a monthly newsletter packed with information on the latest innovations, plus access to a wealth of resources. Register today to join the Sun Inner Circle Program at sun.com/joinic.

To receive additional information on Sun software, products, programs, and solutions, visit sun.com/software.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 800 786-7638 or +1 512 434-1577 **Web** sun.com

We make the net work.

Sun Worldwide Sales Offices: Africa (North, West and Central) +33-13-067-4680, Argentina +5411-4317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323, Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0, Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India–Bangalore +91-80-2298989/2295454; New Delhi +91-11-6106000; Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500, Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466774, Korea +822-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33 1, Malaysia +603-21161888, Mexico +52-5-258-6100, The Netherlands +00-31-33-45-15-000, New Zealand–Auckland +64-9-976-6800; Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China–Beijing +86-10-6803-5588; Chengdu +86-28-619-9333; Guangzhou +86-20-8755-5900; Shanghai +86-21-6466-1228; Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Singapore +65-6438-1888, Slovak Republic +421-2-4342-94-85, South Africa +27 11 256-6300, Spain +34-91-596-9900, Sweden +46-8-631-10-00, Switzerland–German 41-1-908-90-00; French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-344-6888, Turkey +90-212-335-22-00, United Arab Emirates +9714-3366333, United Kingdom +44-1-276-20444, United States +1-800-555-9SUN or +1-650-960-1300, Venezuela +58-2-905-3800 03/04 R1.0