УДК 004.9

ОСОБЕННОСТИ РАБОТЫ С ПРОПРИЕТАРНЫМИ СТАНДАРТАМИ КОДИРОВАНИЯ ДАННЫХ

[1] Кириченко М.А., [1]Крюков Д.А.

[1] Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет информационных технологий, радиотехники и электроники» (МИРЭА), Москва, Россия (119454 Россия, г. Москва, проспект Вернадского, 78), e-mail: kmar.kmar@mail.ru, dk@memfis.su

Рассматриваются методы работы с распространенными форматами кодирования данных в прикладных программах, входящих в пакет Microsoft Office, в том числе последней версии.

Ключевые слова: Microsoft Word, Microsoft Excel, OpenXML, кодирование, проприетарный стандарт.

SPECIFICS OF HANDLING PROPRIETARY STANDARD DATA ENCODING

[1] Krykov D.A., [1] Kirichenko M.A.

[1] Federal State Educational Institution of Higher Education «Moscow State University of Information Technologies, Radio Engineering and Electronics»(MIREA),Moscow, Russia (119454 Russia, Moscow, Vernadskogo avenu, 78), e-mail: kmar.kmar@mail.ru, dk@memfis.su

Various methods of handling widespread data encoding formats in applied software included in the Microsoft Office suite, including it's latest version are examined.

Key words: File, Format, DOC, XLS, Microsoft Office.

Введение

Современные продукты офисного пакета Microsoft, такие как текстовый процессор Word, табличный процессор Excel, занимают значительную долю использования не только в операционных системах семейства Windows, но и в MacOS, Android, iOS и других. Монополизм, возникший на рынке редакторов работы с текстовой информацией, подготовки таблиц, отчетов и презентаций позволяет внедрять проприетарные схемы кодирования данных, в том числе и в ущерб обратной совместимости с предыдущими поколениями пакетов собственных программных средств.

Среди информации, содержащейся в данных контейнерах, безусловно наибольшую ценность представляет текстовое содержимое. В целях обработки текстового содержимого рассмотрим возможные методы его извлечения из файла. В некоторых случаях, API операционной системы вполне достаточно для решения этой задачи, так как информация хранится в обычном текстовом формате. К таким относятся форматы TXT, CSV, XML и многие другие. Однако при взаимодействии с документами Microsoft Office, мы встречаемся с ограничениями, обусловленными проприетарной кодировкой данных.

В статье будут рассмотрены особенности работы с данными, закодированными в контейнерах, внешне представленных в распространенных форматах текстового процессора Microsoft Word и табличного процессора Microsoft Excel. Материалы, изложенные в статье, помогут выбрать наиболее эффективный вариант взаимодействия с файловыми массивами при разработке автоматизированных систем и программных средств, предназначенных для решения прикладных задач.

Проприетарные стандарты кодирования

Бинарные форматы Microsoft на сегодняшний день являются распространёнными стандартами кодирования документов и электронных таблиц Это форматы хранения данных DOC и XLS, которые в результате высоких позиций Microsoft Windows на рынке операционных систем, быстро стали стандартами де-факто, несмотря на крайне сложное внутреннее строение. Для работы с бинарными форматами нет необходимости в использовании особых инструментов, корректное чтение и запись могут производиться системными функциями. Бинарные файлы хранят представление данных в памяти компьютера.

В таких случаях необходимо разбираться в тонкостях строения формата, чтобы понимать, какие метаданные файла следует считывать, чтобы в конце-концов узнать необходимый участок памяти. Это отчасти делается теми же системными средствами, но промежуточная обработка, такая как перемещение курсора чтения и интерпретация прочитанных данных, должна задаваться программистом.

Решить такую задачу на первый взгляд представляется возможным следующими способами:

1. Использование библиотеки Interop. Отметим, что в данном случае потребуется установленная версия Microsoft Office, с помощью которой и будет производиться чтение из файла. Данное решение будет слабым звеном конвейера обработки данных в любом программном проекте, потому что на каждое открытие файла будут создаваться собственные COM-объекты.
2. Подключение через OLEDB. Требует наличия интерпретатора OLE файлов, в качестве которого обычно выступает Microsoft Office. Сторонние средства в большинстве случаев не могут открывать файлы бинарных форматов (DOC и XLS).

Данные решения могут рассматриваться в качестве узлов автоматизации обработки данных, но тем самым можно породить зависимость от сторонних приложений, либо, компонентов Microsoft Office. Рассмотрим подробней данные методы.

Возможности работы с проприетарными форматами

Самым распространённым решением задачи структурированного чтения данных, кодированных соответствующим образом, является использование программы,

распространяемой производителем. При разработке своего программного обеспечения и необходимости работы с форматами DOC и XLS на программном уровне как правило применяется сборка Interop, COM-интерфейс или технология OLEDB.

COM-интерфейс

COM-интерфейс позволяет подключаться к другому приложению и использовать его функционал. Для этого часть приложения запускается операционной системой после того, как был успешно подключен COM интерфейс из вызывающего приложения.

Очевидным недостатком этого подхода, препятствующим его эффективное применение, является использование стороннего приложения, наличие которого не может быть гарантировано в целевой системе. Другой проблемой является низкая скорость работы, так как необходимо гораздо больше времени на подключение интерфейса и запуск приложения, чем на простое подключение библиотеки.

Interop

Эта сборка позволит приложению обмениваться данными с Microsoft Office через его объектную модель, основанную на COM.

Эти сборки значительно упрощают взаимодействие между разрабатываемым приложением и COM-подобными библиотеками Office 2003. Сборки Interop Microsoft для компонентной объектной модели позволяют вызывать неуправляемый COM-код из управляемого кода .NET посредством среды .NET и общего языка выполнения. Сборки Interop позволяют неуправляемым приложениям связываться с управляемыми приложениями во время компиляции. Тогда, приложение сможет предоставлять информацию общему языку выполнения о том, как неуправляемые типы должны обрабатываться во время выполнения.

Недостатки этого подхода аналогичны использованию простого подхода с COM-интерфейсом: низкая скорость работы, зависимость от наличия приложения, способного работать с файлами DOC и XLS. Отличием является то, что сборка ориентирована исключительно на работу с приложением из пакета Microsoft Office. В обмен на ограниченность выбора рабочего приложения мы получаем более альтернативную форму работы с файлом и самим обрабатывающим приложением.

Технология OLEDB

OLEDB является программным интерфейсом приложения, основанным на COM для получения доступа к данным. OLEDB поддерживает любые форматы, для которых есть соответствующий провайдер OLEDB.

OLEDB по образу работы идентичен Interop за исключением интерфейса. В целом, можно считать, что технологии идентичны.

В качестве несомненного преимущества OLE можно сказать, что он предоставляет более широкий функционал в сравнении с Interop, так как работает с приложением на более низком уровне. Тем не менее, это решение не позволит гарантировать автономность работы с файлами.

Альтернативные возможности работы с проприетарными форматами

В сложившихся условиях представляется целесообразным рассмотреть решение, разработанное под операционные системы семейства Linux, расширяющие возможности команды cat – чтение содержимого файла. По названию можно понять основное предназначение данного средства — извлечение данных из закрытого формата DOC, что является давней проблемой для любых операционных систем, не распространяющихся под маркой Windows. Возможности catdoc далеко не ограничены форматом DOC. В настоящее время пакет catdoc дополнен программами xls2csv и catppt.

Эта утилита имеет широкое распространение ввиду большой мобильности приложения и высокой скорости работы. Catdoc не требует никакой установки перед использованием и не требует наличия вспомогательных приложений, способных работать с проприетарными форматами. Также утилита стала популярной из-за её практичности в автоматизированной обработке текстового содержимого файлов — нет необходимости в настройке приложений: даже в многопоточном режиме утилита работает без дополнительных параметров. Исходный код этого набора программных средств распространяется свободно, как и сама утилита.

Необходимо отметь, что утилита Catdoc ориентирована исключительно на чтение документа. Очевидно, что программа пренебрегает стилями, шрифтами и прочим форматированием текста игнорируются при чтении.

Среди альтернативных возможностей работы с электронными таблицами XLSX и XLS можно выделить библиотеку roo, распространяемую в составе инструментальных средств языка программирования Ruby. В ходе тестирования были выявлены незначительный недостаток: библиотека имеет задержку открытия файла (как было выяснено, связано это с процедурой распаковки архива во временную директорию).

Возможности работы со стандарты, основанными на Open XML

Функционала чтения данных из целевого файла порой недостаточно, чтобы получить их в формате, требуемом для автоматизированной обработки. Использование системных функций достаточно лишь тогда, когда данные хранятся в файлах в текстовом виде. Однако в некоторых случаях вместе с необходимой нам информацией есть возможность обеспечить получение и метаданных документа.

Примерами такого сценария являются форматы XML и HTML. В этих случаях весь файл действительно можно успешно прочитать, приняв его за текстовой, но в таком случае вместе

с текстом мы получим огромное количество метаданных файла, определяющих стили, структуру, источники данных и внешнее оформление документа.

Похожим образом устроены и современные форматы текстовых документов и электронных таблиц: DOCX и XLSX. Отличие этих форматов от XML лишь в том, что они представляют собой архив набора файлов форматов XML и директорий.

Данный формат именуется Open XML (стандарт ECMA 376), определяющий набор схем XML для представления электронных таблиц, презентаций и текстовых документов. Этот формат используется Microsoft Word и Microsoft Excel начиная с 2007 версии в качестве основного формата. Преимуществом формата Open XML для разработчиков является то, что он основан на двух широко известных технологиях: ZIP и XML.

Для автоматизированной работы с такими файлами достаточно разархивировать исходный документ и обработать содержимое. Основной трудностью в таком подходе является поиск необходимых данных в результирующем наборе файлов и отделение от этих данных соответствующих им метаданных.

OpenXML был изначально спроектирован с целью достоверного представления уже существующего объема документов, презентаций и электронных таблиц, которые создавались с использованием бинарных форматов определённых корпорацией Майкрософт. Процесс стандартизации состоял из зеркального отображения в XML необходимого функционала для представления существующих файлов, расширяя их, предоставляя подробную документацию и предоставляя возможности для взаимодействия. Стандарт определяет форматы для текстовых документов, презентаций и электронных таблиц. Каждый тип документа определён с помощью базового языка разметки: WordprocessingML, PresentationML, или SpreadsheetML. Механизмы внедрения позволяют документу любого из этих типов содержать материал в остальных базовых языках разметки и в некоторых поддерживающих языков разметки.

Файл OpenXML хранится в ZIP архиве в целях сжатия и хранения, согласно рекомендованной реализации открытых схем упаковки (Open Packaging Conventions). В реализации хранения ZIP, файл «[Content_Types].xml» позволяет пользователю определить тип содержимого для каждой части архива.

Пакеты и его части могут содержать явные связи с другими частями пакета или с внешними ресурсами. Каждое явное взаимоотношение содержит идентификатор и тип, что позволяет на него ссылаться и определять необходимый метод обработки информации.

Заключение

Широкое распространение проприетарных форматов DOC и XLS действительно представляет сложности для разработчиков программного обеспечения. Единственным

очевидным методом работы с такими файлами на программном уровне оказались компоненты платформы Microsoft Office. Это обстоятельство может замедлить развитие независимых решений и автоматизированных систем, предлагающих автономный подход к решению задачи работы текстовыми документами и электронными таблицами. В результате проведенного сравнительного анализа обнаружены доступные и привлекательные решения вследствие их независимости от пакетов Microsoft Office.

Список литературы

1. Общие положения по Interop, Корпорация Майкрософт (https://msdn.microsoft.com/en-us/library/15s06t57.aspx)
2. Техническая информация Interop, Корпорация Майкрософт (https://support.microsoft.com/en-us/kb/897646)
3. Введение в OLEDB, Abhishek Sur (http://www.codeproject.com/Articles/37055/Working-with- MS-Excel-xls-xlsx-Using-MDAC-and-Oled)
4. Использование OLEDB для подключения к средствам Microsoft Office, Корпорация Майкрософт (https://msdn.microsoft.com/en-us/library/aa288452(v=vs.71).aspx)
5. Word Extensions to the Office Open XML (.docx) File Format (https://msdn.microsoft.com/en-us/library/dd773189(v=office.12).aspx)
6. Excel (.xlsx) Extensions to the Office Open XML SpreadsheetML File Format (https://msdn.microsoft.com/en-us/library/dd922181(v=office.12).aspx)
7. Excel Binary File Format (.xls) Structure (https://msdn.microsoft.com/en-us/library/office/cc313154(v=office.12).aspx)
8. Применение catdoc. (http://linux.die.net/man/1/catdoc)
9. Ruby roo: применение. (URL: https://github.com/roo-rb/roo)