

ВЫВОД В ДОКУМЕНТЫ MS WORD ОТЧЁТОВ ЭЛЕКТРОННОГО ДИСПЕТЧЕРСКОГО ЖУРНАЛА ДЛЯ SCADA «СИСТЕЛ»

Автор: Байдакова Наталья, 5 курс.

Руководитель: Кульман Никита Юрьевич, канд. физ.-мат. наук.

Высшее учебное заведение: «Международный университет природы, общества и человека «Дубна»», филиал г. Протвино

OUTPUT OF ELECTRONIC DISPATCHER JOURNAL REPORTS TO MS WORD DOCUMENTS FOR SCADA “SYSTEL” Baydakova N.

Одним из основных стандартов электроэнергетики, обязательным для применения, является документ «Правила технической эксплуатации электроустановок потребителей». П. 6.10.1 раздела 6.10 ПТЭ «Автоматизированные системы диспетчерского управления» требует оснащения диспетчерских пунктов всех уровней управления автоматизированными системами диспетчерского управления. Одним из требований к функциональности такой системы является ведение оперативного диспетчерского журнала (п. 6.10.4 ПТЭ).

Данная работа делалась на базе предприятия ООО «СИСТЕЛ» (г. Протвино). На этом предприятии был разработан оперативный информационно-управляющий комплекс [1], предназначенный для приема, обработки, передачи и хранения телеметрической информации о режиме работы предприятий энергетики. При работе комплекса необходима регистрация событий диспетчерского персонала. В настоящее время для этого используется бумажный оперативный журнал. В 2007г. было выпущено Информационное письмо УГЭН Ростехнадзора № 10-04/3009/2 от 08.10.2007г., рекомендующее использование электронного оперативного журнала в сетевых и генерирующих компаниях электроэнергетики.

Электронный оперативный журнал предназначен для ведения, хранения и анализа оперативно-диспетчерской документации диспетчерского центра, включая диспетчеров, дежурных инженеров оперативного планирования, дежурных инженеров служб телемеханики и связи, эксплуатации программно-аппаратного комплекса, дежурного персонала подстанций.

При ведении журнала возникают ситуации, когда требуется автоматически внести данные в один из типовых документов: служебное или информационное сообщение. Причем, такие документы в основном формируются с использованием Microsoft Office Excel или Word. Это происходит в таких, например, случаях:

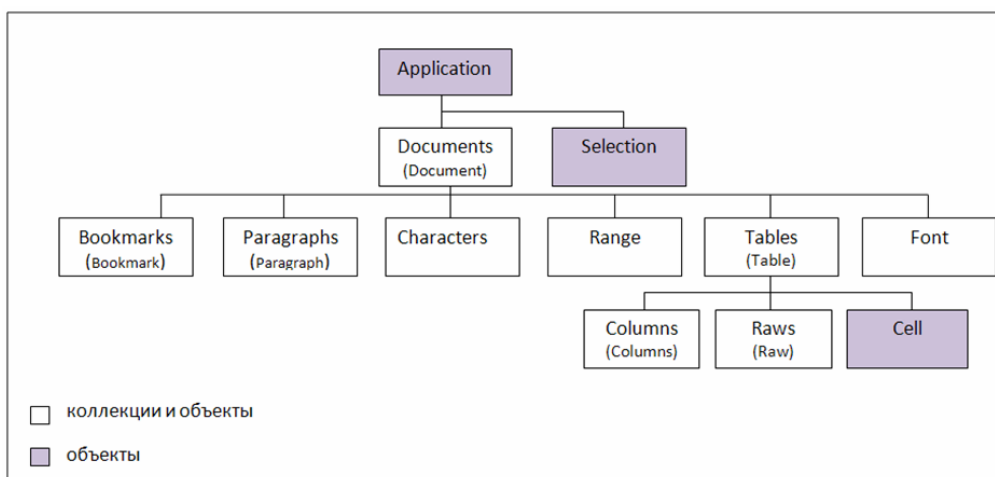
- требуется сформировать служебное сообщение, которое представляет собой довольно длинный текстовый документ, в который нужно вставить несколько чисел или строк – например, даты.
- нужно обеспечить возможность ручного редактирования текста документа. Обычно для этого создается стандартный шаблон документа, который пользователь имеет возможность редактировать.

Для генерации и сохранения отчетов в электронном журнале применяют сервисы, предоставляемые Microsoft Office. Все компоненты Microsoft Office поддерживают один и тот же макроязык: Visual Basic for Applications (VBA), позволяющий создавать приложения непосредственно внутри документов Office (это называется «решения на базе Microsoft Office»). Управление же компонентами Office из других приложений осуществляется с помощью автоматизации [2] (Automation, ранее — OLE Automation) — все приложения Microsoft Office являются серверами автоматизации (или COM-серверами). Для создания таких приложений пригодны любые средства разработки, позволяющие создавать контроллеры автоматизации (COM-клиенты).

Автоматизация — это одна из возможностей, предоставляемых технологией Microsoft COM (Component Object Model) [3]. Автоматизация считается одной из самых мощных OLE-технологий. Это механизм, при помощи которого объект предоставляет другим объектам и приложениям доступ к своим методам (функциям) и свойствам (характеристикам). Объект, предоставляющий доступ к своим функциям, называется сервером автоматизации. Объект или приложение, которые используют функции или управляют действиями сервера автоматизации, называются контроллерами автоматизации. Серверы автоматизации существуют для того, чтобы обслуживать запросы и указания контроллеров автоматизации.

Вся информация об объектах, необходимая контроллерам автоматизации, содержится в библиотеках типов. Библиотеки типов хранятся в специальном бинарном формате и обычно представлены в виде файлов с расширениями *.olb или *.tlb, а также могут содержаться внутри исполняемых файлов (*.exe) или динамически загружаемых библиотек (*.dll).

Внутри себя приложения имеют иерархическую структуру (рис. 1). Самым верхним уровнем является Application (приложение), ниже которого лежат объекты и коллекции объектов. Объектами можно управлять напрямую, а из коллекции сначала нужно извлечь экземпляр, затем уже работать с ним.



«Рис. 1. Иерархия объектов Автоматизации Word»

Основным объектом в иерархии Word, как и во всех других приложениях Microsoft Office, является объект Application, содержащий коллекцию Documents объектов типа Document. Каждый объект типа Document содержит коллекцию Paragraphs объектов типа Paragraph, Bookmarks типа Bookmark, Characters типа Character и т.д. Манипуляция документами, абзацами, символами, закладками реально осуществляется путем обращения к свойствам и методам этих объектов.

Для получения коллекций необходимо создать класс MFC Class From TypeLib и выбрать библиотеки типов Microsoft Word 12.0 Object Library<8.4>.

Для создания нового документа Word используется метод Add коллекции Documents объекта Application: `oBooks.Add(covOptional);`

Обычно вывод отчета в документ Microsoft Office Word (MS Word) применяется при необходимости корректировки данных отчета. Чаще всего, при этом используется шаблон документа, в котором текст вставляется в нужные места.

Шаблон Microsoft Word определяет основную структуру документа и содержит настройки документа, такие как элементы автотекста, шрифты, назначенные сочетания клавиш, параметры страницы, форматирование и стили. Шаблон создается, как и стандартный документ Microsoft Word с указанием типа файла - Шаблон документа. В результате создается документ MS Word с расширением .dot. Чтобы поместить текст в определенное место шаблона используются *закладки*.

Для открытия документов Office необходимо инициализировать систему OLE. Сервера автоматизации заносят данные о себе и своих объектах в реестр, и к ним можно обращаться по символьным именам. Для MS Word - это "Word.Application". Для работы с автоматизацией необходимо проинициализировать OLE с помощью функции `AfxOleInit`. Далее необходимо создать экземпляр Word – `app.CreateDispatch`, и сделать его видимым для приложения `app.SetVisible`.

Для открытия уже существующего шаблона нужно указать имя шаблона в качестве параметра метода Open: `oBooks.Open(FileName, covOptional, covTrue, ..., covOptional);`

Свойство ActiveDocument объекта Word.Application указывает на текущий активный документ среди одного или нескольких открытых. Помимо этого к документу можно обращаться по его порядковому номеру с помощью метода `get_Item`.

Сохранить документ можно с помощью метода `Save`. Этот метод имеет несколько параметров, первый из которых задаёт имя сохраняемому документу. Вторым параметром является формат сохраняемого документа. Также документ можно защитить паролем, установив третий параметр как `COleVariant("password")`.

Если необходимо вывести документ Word на устройство печати, то это можно осуществить с помощью метода `PrintOut` объекта Document. Если нужно изменить параметры печати, следует указать значения соответствующих параметров метода `PrintOut` (для Microsoft Word их около двадцати).

Существует возможность запрета на форматирование документа, при этом используется метод `Protect`.

С помощью средств OLE автоматизации было достигнуто автоматическое внесение данных информационного сообщения в документ Microsoft Office Word. Вывод сообщения в MS Word является наиболее удобным форматом документа для просмотра и редактирования данных.

Работа выполнялась в среде разработки Visual Studio 2008 [4].

В результате выполнения данной работы в электронном журнале реализована возможность вывода отчётов информационного сообщения в документы MS WORD. В настоящее время происходит внедрение разработанного ПО на диспетчерский пункт предприятия кабельных сетей ОАО «ЛЕНЭНЕРГО».

Литература

1. Рыкованов С.Н, Кульман Н.Ю., Ухов В.И. Развитие программного обеспечения верхнего уровня ПТК «Систел». Современные системы сбора, передачи, обработки и отображения информации на объектах электроэнергетики. Сборник докладов. Москва НЦ ЭНАС, 2006г.
2. Джейсон Причард. СОМ и СОRВА. Архитектуры, стратегии и реализации, М.: Издательство «ЛЮРИ»; Москва, 2001. – 372 стр.: ил.
3. Рихтер Джефффри и Назар Кристоф. Windows via C/C++. Программирование на языке Visual C++ / Пер. с англ. – М.: Издательство «Русская Редакция»; СПб.: Питер, 2008. – 896 стр.: ил.
4. Пауэрс Л. Microsoft Visual Studio 2008/ Л Пауэрс, М.Снелл: пер. с англ. – СПб.: БХВ-Петербург, 2009. – 1200 с. : ил.