

Technical writing test

# Task 1. Authoring

Please explain — step-by-step — how to insert a table into a Microsoft Word document.

* The table must contain ten rows and six columns, fitting the page width.
* The header background must be 10% grey.
* When the table spans multiple pages, its header must repeat on every page.
* The header text must be centred.
* The table body text must be aligned centre-left.

These instructions help you create the above mentioned table.

**To insert a table**

1. Open your Word document and navigate to the **Insert** tab.
2. Select **Table** > **Insert Table**.
*This opens the* ***Insert Table*** *dialog box that allows you to specify the table parameters:* 
In the **Insert Table** dialog box, do the following, and then click **OK**:
* In **Number of columns** and **Number of rows**, specify the number of columns and rows, respectively (such as 6 and 10).
* Select the **AutoFit to window** option.

**To repeat the table header on each page**

1. Right-click the first row of the newly created table, and click **Table properties** on the shortcut menu.
*This opens the* ***Table Properties*** *tabbed page similar to the following screen*:

2. Navigate to the **Row** tab, select the **Repeat as header row** **at the top of each page** checkbox, and then click **OK**.

**To configure the table header properties**

1. Highlight the table header row.
2. Right-click the header row, click **Table properties** on the shortcut menu, and then navigate to the **Table** tab.
3. Click **Borders and Shading** and navigate to the **Shading** tab of the **Borders and Shading** dialog box that opens:


4. From the **Fill** list, select the header background (such as grey).
5. Highlight the header row, and then press CTRL +E.
*This centers the header text.*

**NOTE**: The table body text is center-left aligned by default.

# Task 2. Editing, Language and Grammar

The text below is an extract from a user manual for a desktop email client. Please check the text for issues, apply formatting and rearrange the material to your discretion in order to improve it.

Note: Before beginning this task, please enable the **Track Changes** feature for this document.

## Eudora Setup

Before you start using Eudora, ensure that your environment meets the minimum system requirements, and configure Eudora following the provided instructions.

 Minimum System Requirements

This specification defines the minimum system requirements for Eudora:

* An IBM PC-compatible computer running Windows 95.
* An instance of Eudora Lite application is installed and currently running.
* Internet access
* POP3 Email account.

NOTE: If you do not know the type of your Email account, contact your Internet Service Provider. POP3 is a client/server protocol in which Email is received and held by your mail software on the remote server. Presently, most mail software uses the POP3 protocol.

Configuring Eudora

To use Eudora for performing basic Email tasks, start Eudora and configure the program settings.

.

*.*

Eudora needs a few information to send and receive your mail messages.: Figure 1 illustrates Eudora User Interface and menu bar:



Figure 1. Eudora User Interface

**To configure Eudora**

1. On the Eudora menu bar (see Figure 1), point to **Tools**, and then click **Options**.
*This opens the* ***Options*** *dialog box similar to the following screen:*

.

1. Under **Categories**, select the **Getting Started** icon, and then specify your POP account information in the **POP account** and **Real name** text boxes, respectively.
*TIP: Your POP Account is your username followed by the "@" symbol and the name of your Email server. For example, if your user name is "jdoe" and your server is ".mail.com", enter "jdoe@ mail.com" in the* ***POP Account*** *space.   )*
2. In the **Real Name** text box, enter your real name.
*This name identifies you for your Email recipients.*
3. Under **Connection Method**, select the **Winsock (Network, PPP,  SLIP** ) option.
4. Click **OK** to save your changes.

After completing this procedure, Eudora is ready to communicate with your Email server.

# Task 3. Translation

Please translate the following texts. Keep the **Track Changes** feature enabled.

**A.** Into English:

## О чем этот пост

Ни для кого не секрет, что производительность приложения зависит от множества различных факторов, начиная с серверов, структуры БД, кода самого приложения и ещё кучи всего.
В этом посте я попыталась описать последовательность действий, средства, утилиты и прочие штуки, которые я использую для идентификации проблем с производительностью баз данных на SQL Server 2000-2008. Всё, что здесь описано, нажито непосильным трудом и долгими годами. Это не значит, что я противник других СУБД, так сложилось, что большинство моих проектов именно на MS SQL Server’ах. Многие скрипты, утилиты и прочее имеют свои авторские права и лицензионные соглашения, поэтому прошу учитывать это, если будете использовать.

Кому это будет полезно?

Всем, кто занимается нагрузочным тестированием, администрированием, поддержкой, разработкой приложений с БД на SQL Server 2005-2008. С SQL Server 2000 это так же может быть совместимо, но некоторые запросы придется подкорректировать.

Пример, когда это может быть использовано

Есть некое .NET приложение, использование ресурсов которым на сервере приложения под нагрузкой нормально, но при этом зашкаливает время отклика приложения и растет очередь обработки запросов («ASP.NET\Requests Queued» performance counter). Если профилирование приложения (к примеру, dotTrace профайлером) не дал никаких результатов или явно указывает на проблемы со временем выполнения запросов к БД.

[Replace with your text]

Summary

It well-known that an application performance depends on several factors, such as the server types, DB structure, the application code, etc.

In this article, I tried to describe the procedures, tools and other utilities used to detect the performance issues of databases managed by SQL Server 2000-2008. This article is a result of my intensive and multi-year work. This does not mean that I ignore other database engines. In fact, most of my projects use MS SQL Server. Please note that most of scripts, utilities and other tools are protected by copyrights and license agreements.

Target Audience

This article is for IT specialists that deal with development, and load testing of the DB solutions based on SQL Server 2005-2008. Additionally, after correcting some of requests, you can use this information for SQL Server 2000.

Example of Use

Consider the following case: a .NET application works properly on the application server under normal load conditions, but the application response time drastically increases that results in the request queue («ASP.NET\Requests Queued») full. This technique is also useful when the performance profiling data (for example, with the use of dotTrace) does not help you detect any issues or reveals issues related to the DB query response time.

**B.** Into Russian:

## Integration part

The communication interface between merchant and WHLS system is performed server-to-server via HTTPS (using POST method). An action processor will parse these commands and generate appropriate responses that will be sent back. For security reasons, WHL MII can be invocated only from certain merchant server IP (IP range in case of clusters), which is defined in Merchant application Form.

To integrate WHLS into a merchant’s system, two schemas can be used:

* The simplest integration model implies that merchant customers will be redirected to a WHL Lobby page. This page will be customized to each merchant individually including unique logo, sub-domain name, and game list. In this case, all WHLS games will be mapped to the corresponding WHL Lobby page.

This integration model is more stable then the second one, because WHL Lobby functionality is automatically supported by WHL back-end system. In case of changes in game URLs or merchant integration parameters, all changes are implemented automatically.

[Replace with your text]

**Схема интеграции**

Интерфейс взаимодействия между сервером торговой точки и сервером WHLS (система оптовых продаж) реализован на основе модели сервер-сервер через протокол HTTPS (используется POS метод). Модуль обработки распознает запросы и генерирует соответствующие отклики, которые отправляются обратно. По соображениям безопасности, доступ к интерфейсу WHL может осуществляться только с серверов торговых точек с определенным IP адресом (или с определенного диапазона IP адресов в кластерной модели). Допустимые адреса определяются в заявке торговой точки.

Для интеграции системы оптовых продаж с сервером торговой точки могут использоваться две схемы:

* В самой простой модели интеграции, клиенты торговой точки перенаправляются на Главную страницу системы оптовых продаж. Эта страница настраивается индивидуально для каждой торговой точки и включает уникальный логотип, уникальный поддомен и список игр. В этом случае, все игры WHLS будут доступны на соответствующей Главной странице WHL.

Данная модель более устойчива, чем другая модель, поскольку функциональность Главной страницы WHL автоматически поддерживается внутренним cервером WHL. В случае изменения адреса игры WHL или параметров интеграции торговой точки, все изменения будут учитываться автоматически.

## Method description.

whlapi.status(callback)- void. The method retrieves user login status and session information. It is called automatically on page load. After WHL provider checks the current user status the callback should be called. Callback is a function getting one object parameter, which contains the fields:

* uid : Unique connection or client identification string
* token : Authentication token for Server-Server call, if user info present, string.
* status : string *'real'* for logged in user or *“anon”* for anonymous,
* message : Error message, if present, string,

In case status is *'anon'* (user logged out), data object will contain antifraud elements to be sent by whlapi.login and whlapi.resetPassword functions.

This is done automatically, no need to perform antifraud measures on our side.

[Replace with your text]

**Описание метода**

whlapi.status(callback)- void.

Этот метод получает статус подключения пользователя и информацию о сессии. Метод автоматически вызывается при загрузке страницы. После того, как провайдер WHL проверит статус текущего пользователя, будет вызвана функция callback , callback – это ссылка на функцию с одним входным параметром – объектом. Этот объект содержит следующие поля:

* uid : уникальный идентификатор пользователя или подключения (тип – строка).
* token : маркер аутентификации для вызова сервер- сервер, при наличии информации о пользователе. (тип – строка).
* status : *'real'* для зарегистрированного пользователя или *“anon”* для анонимного пользователя. (тип – строка)
* message : сообщение об ошибке (при наличии ошибок) (тип – строка).

В случае, если статус “anon” (пользователь не вошел в систему), объект будет содержать элементы для предотвращения несанкционируемого доступа (защиты от мошенников), которые будут отосланы через функции whlapi.login and whlapi.resetPassword. Это выполняется автоматически и нет необходимости принимать дополнительных мер по защите на нашей стороне.