INTRANET SYSTEM
ONLINE LEAVE APPLICATION SYSTEM

by

TEOH WEI PEI

PROJECT SUPERVISOR: PN. ZAILANI IBRAHIM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF INFORMATION TECHNOLOGY,
COLLEGE OF ENGINEERING,
UNIVERSITI TENAGA NASIONAL

2002

DECLARATION

I hereby declare that this thesis, submitted to Universiti Tenaga Nasional as a partial fulfillment of the requirements for the degree of Bachelor in Information Technology has not been submitted as an exercise for a Degree at any other university. I also certify that the work described here is entirely my own except for excerpts and summaries whose sources are appropriately cited in the references.

This thesis may be made available within the university library and may be photocopied or loaned to other libraries for the purpose of consultation.

5 April 2002

Teoh Wei Pei
IT 02478

ACKNOWLEDGMENTS

The successful production of this report involves many valuable contributions from a number of parties. Firstly, I would like to thank my project supervisor, Pn. Zailani Ibrahim, and my project second reader, Mr. Steve Teoh Chee Hooi, for the valuable advices, guidance, and assistant in the completion of this thesis report. She had helped me a lot from the very beginning until the completion of the documentation of this project. She helped to solve my problems and uncertainties by giving some effective and relevant solutions to me.

Besides, I would like to thank all the COE staffs and lecturers. I really appreciate the comments, suggestions and cooperation given. With their helps, I managed to identify and collect the user requirements on the project.

Furthermore, I would like to thank Pn. Shamsidah for her valuable time to accept my interview. By having interview with her, I manage to know the actual operation of the existing leave system in COE UNITEN. And also a special thank you to Mr. Nadzri Yusoff, who has given me valuable advices on designing the network part of my system.

Moreover, special thanks to my friends and family for their care and support. Their understanding and encouragement given to me is a valuable one.

Thank you.

ABSTRACT

An intranet is a website designed to be accessed within a company or an organization. An intranet is designed with information sharing between various departments and groups in mind. The Internet concept and technology can be applied to use as intra-network, replacing old-fashioned computer networks. Every function of the Internet can all work on the Intranet. One of advantages of the Intranet is to have an Intranet Web site to communicate within organization. This project is concern about one of the task that can be performed by Intranet, which is the Online Leave Application System. This system aims to computerize leave applications, leave approval and leave balance maintenance, and an up-to-date annual leave balance would be automatically maintained by the system. The system, however, may also be used by the same or other staff to record other leave types, for example, long leave, sick leave, etc. which are not detrimental to annual leave. Recording leave other than annual leave will give a full picture of whether a staff is on leave, and would provide a handy way for colleagues of the same department to check out for themselves who and when will be away from his/her office. This system also enables the administrator to administrate the system in a sufficient, effective and efficient way. This automated leave application system will be proposed to the College of Engineering Department of Universiti Tenaga Nasional. This proposal states the project definition, objectives and scope, and methodology proposed to be used to accomplish the project, literature review that gives a general description about Intranet Leave Application System, description about the existing and proposed system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

PROJECT OVERVIEW

1.1 Introduction

In this era of science and technology, computer is one of the most important components in our life. Works can be done in a better way by the help of computer. So is the leave application system in a company or an organization or department. By applying an online leave application system, there will be no more complicated paper works have to be done. Task can be done in a faster, easier and sufficient way. This also helps to achieve the objective of towards a paperless office.

The purpose of this project is to create an Intranet system about online leave application system for the COE staffs that will enable both the user and administrator in increasing the efficiency and effectiveness on leave application. The system will be user-friendly, ease to use and administrate.

For the current leave application system, all the procedures being carried out manually by paper works and manpower. When a staff wants to apply leave, he has to fill in a form and bring it to the dean/head of department to sign for approval. Then the form will be passed to the administrator. The administrator will check the staff's leave records and carry on some procedures. This is very insufficient and it needs to involve few stages of manpower. It wastes a lot of time and cost.

Therefore, the new system is a proper online leave application system that is able to perform several tasks. It enables the staffs to apply leave through online Intranet system, online leave approval or disapproval by dean/head of department, viewing their own leave application records, leave status of themselves and other staffs, change or cancel leave applied and change their login password. The numbers of leave days will be calculated automatically through the system. Besides, the administrator can also administrate the system easily by adding, deleting, and updating the data. This will be more efficient and sufficient.

1.2 **Problem Statement**

These are the problems that assist in the current system:

- The system is done manually
- Without a proper and effective online leave application system
- Without a proper system to keep track of the staffs' leave records
- Without a proper system for the staffs to check their leave records and information
- Involves a lot of paper works and manpower
- Waste of time and manpower

- Various problems may involves during leave application process, for example, missing of paper documents may cause problems because of no back-up data
- Insufficient, inefficiency and inconvenient

1.3 **Proposed Solutions**

The new system will solve the problems that occur in the current system. It is a proper online leave application system that will be able to perform several important and manageable tasks. It enables the staffs to apply leave through online Intranet system, online approval or disapproval by dean/head of department, viewing their own leave application records, leave status of themselves and other staffs, change or cancel leave applied and change their login password. The numbers of leave days will be calculated automatically through the system. Besides, the administrator can also administrate the system easily by adding, deleting, and updating the data. This will be more efficient and transparent to the users.

CHAPTER 2

OBJECTIVE AND SCOPE

2.1 Objective

- To implement paperless office concept
- To provide an easier and efficient leave application system for COE staffs
- To provide a service that enable the administrator to administrate the system effectively and sufficiency
- To provide an online task that enable the dean/head of department to approve or disapprove the application
- To reduce wastage in time and manpower costing
- To provide a better working environment and increase the working efficiency of the company

2.2 Scope

UNITEN COE Online Leave Application System is developed strictly for the use of UNITEN COE communities. Any usage of this web site by outsiders is prohibited as this leave application system is developed for internal use only. All the transactions, including leave application, cancellation, approval / disapproval, administration and so on, that to be catered are only involving one level of supervisor-subordinate. This will be more efficient and transparent to the users.

CHAPTER 3

METHODOLOGY

3.1 Development Process

Each development of a system must have a few defined stages that will eventually lead to a successful completion. These stages varied from one project to another. Usually it has a minimum of 3 steps and maximum of 12 steps (Source: Jeffrey A. Hoffer). This project is developed using the System Development Life Cycle (SDLC), which consist of 7 stages:

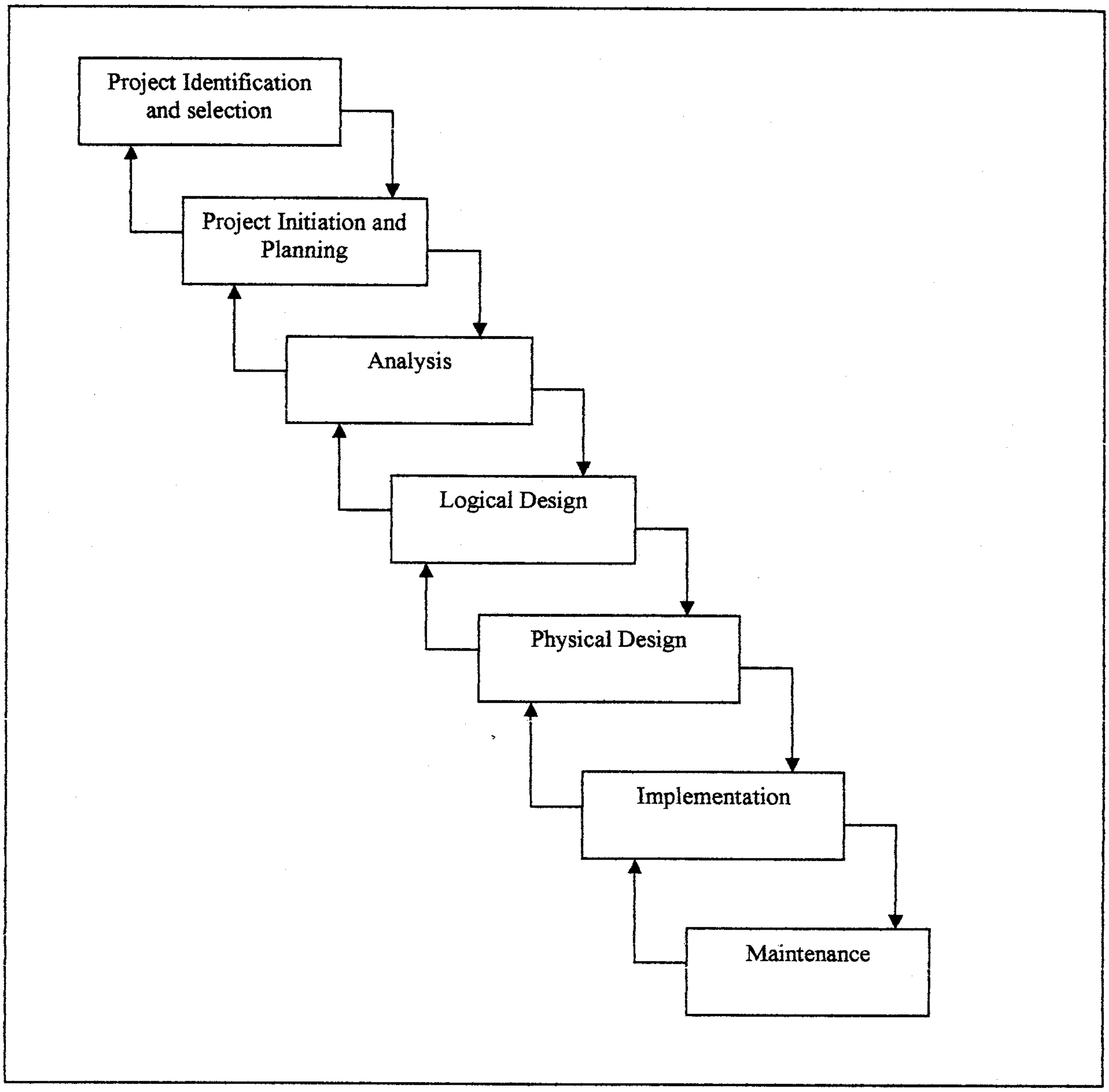
Project Identification and selection
Project Initiation and Planning
Analysis
Logical Design
Physical Design
Implementation
Maintenance

Figure 3.1 **System Development Life Cycle (SDLC)**

3.1.1 **Project Identification and Selection**

The first phase of the SDLC is project identification and selection. This stage consist of three primary activities:

3.1.1.1 ***Identifying potential development project***

After the finalized copy of projects was released, a few choices of interest were noted down as my potential final project. Among the projects are Intranet Application for Leave Application / Approval / Cancellation / Record Through Electronic Mail, Room Booking System, Claim System, and so on. After short-listing a few project of interest, a meeting with the lecturer who proposed the system is conducted. The purpose is to gather more information on the requirement, scope and specification of project.

3.1.1.2 ***Classifying and ranking projects***

Projects are classified to several categories such as web-based, theory reports, expert systems, Artificial Intelligence and Information Systems. After sufficient information is collected, each of the projects are analyzed and ranked, based on interest, practicality, familiarity and my own capability.

3.1.1.3 ***Selecting projects for development***

After weighing all factors, the Intranet Leave Application System was chosen. This is due to my own experience as managing web hosting and Intranet system during my industrial training in Shell Service International (SSI). I was also assigned a project to create a Leave Request System for SSI during my industrial training.

3.1.2 **Project Initiation and Planning**

The two major activities that occur in the first phase are the process of initiating the project and planning the system development project. Project initiation focuses on activities designed to assist in organizing the project development. These activities include defining the objective of the system and the system request.

Project planning is the second activity in the project initiation stage. It is a process of defining clear and discrete activities and the work needed to complete each activity within a single project. The tasks are carefully evaluated to make the best use of the resource. After identifying tasks for each development phase, it is charted into a Gantt Chart *(please refer to Appendix A)*, which contain all details pertaining to the project schedule.

The objective of project planning process is the development of a Baseline Project Plan (BPP) and the Statement of Work (SOW). The BPP contains all information collected and analyzed during project initiation and planning. The plan reflects the best estimate of the project's scope, benefits, costs, risks, and resource requirement given the current understanding of the project. The SOW is a short document prepared for the user that described what the project would deliver and outlined all works required to complete the project.

3.1.3 **Analysis**

Analysis is the first system development life cycle phase where I begin to understand, in depth, the need of the system. It mainly consists of capturing and restructuring information. The fact-finding techniques are used to learn about the current system (use as a model), the organization (that plans to implement the new system), and the user requirements of the proposed system. The objective of this stage is to determine what information and information processing services are needed to support the proposed system.

The analysis stage consists of two steps:

- Requirements Determination

  Gathering the information about both current and the proposed system. One of the best ways of getting information is through literature review, questionnaires and interview.

- Requirements Structuring

  This activity creates a thorough and a clear description of the current situation of how COE staffs applying leave by the current manual leave application system. Besides that a clear description of the newly proposed online leave application system also being stated in this stage.

3.1.3.1 **Requirements determination**

There are a few information resources in gathering data and information:

- Research papers, journals, and articles available from Internet
- Research on Existing System
- Interview
- Questionnaires
- References books

One of the best ways to get information is through the literature review and interview. (*Please refer to Chapter 4 about the Literature Review that has been gone through).*

3.2.3.2 **Requirements Structuring**

Requirement determination had generated a lot of ideas about how the systems work and what is the criteria that user like most. All the information is then being process and structure into a logical form called, Data Flow Diagram. Data flow diagrams are versatile diagramming tools use to depicting logical information flow.

There are two different standard sets of data flow diagram symbols. Each set consists of four symbols that represent the same thing: data flow, data stores, process, and sources. The set of symbols that are used for this report was devised by Gane and Sarson (1979).

3.1.4 **Logical Design**

This is the stage where the conversion between user and computer is done. All the information gathered in the analysis stage is restructured to create a concrete understanding on how the system will operate. The logical design defines the "look and feel" of all system inputs, outputs, interfaces and dialogues. The conceptual data model is supplemented with new data requirements to produce the relational database model.

Logical design includes the following steps:

- Designing forms and reports. It describes how the data will appear to user in system input and outputs.
- Designing interfaces and dialogues.
- Designing logical databases. It describes the standard structure of the database. This structure is easily implemented in any database technology. The key deliverables of logical design include functional specifications for system inputs, outputs, interfaces, dialogues and databases. However, there is no specific milestone for the logical design as it can sometimes overlap with the physical design.

3.1.5 **Physical Design**

This stage carries the logical design to the actual specifications of the system. The purpose of physical design is to specify all technological characteristics so that those involved in the implementation phase can concentrate on building the system (instead of deciding how the system is to be built). Physical design specifies the structures for data and programs that will make the system work efficiently and securely, including the considerations for the logical of data and data processing on the computer network.

Steps in physical design include:

- Designing physical files and databases – describe how the data will be stored and accessed in secondary computer memory and how the quality of data will be ensured.
- Designing system and program structures – describe the various programs and program modules that correspond to the data flow diagrams and other documentation developed in prior life cycle stages
- Designing distributed processing strategies – describe how the data and processing available to users on computer networks within the capabilities of existing computer networks.

3.1.6 **Implementation**

The purpose of implementation is to convert the final physical system specifications into a working and reliable software and hardware, document the work that has been done and provide help for users and caretakers of the system. Implementation also includes finalizing all system and user documentation, thoroughly training users and others to effectively use the system, and preparing support systems to assist users as they encounter difficulties.

Once built, the system must be installed. The existing system would need to be 'plugged out' and the new system to be 'plugged in'.

Implementation consists of the following steps:

- Writing computer software
- Testing software
- Converting from the old to the new system
- Documenting the system
- Training users and others
- Designing support procedures

3.1.7 **Maintenance**

The purpose of maintenance is to fix and enhance the system to respond to problems and changing business conditions. Maintenance also includes activities from all system development phases, and usually leads to a decision to abandon a system and start the cycle all over again.

Maintenance includes the following procedures (Hoffer):

- Responding to requests to change the system
- Transforming requests into changes
- Designing the changes
- Implementing changes

3.2 Technology Used

3.2.1 Hardware and Software Requirement

The only required hardware that is needed is computers with Internet and Intranet accesses. A considerable amount of hard disk might be needed. A back-up system might be needed in case of fire or shortage of power supply.

Microsoft's software such as MS FrontPage 2000, MS Access 2000, and MS IIS 5.0 might be used to create the system with a combination of other software. Programming language such as Active Server Pages (ASP) will be used too.

3.2.2 **Scripting Language**

3.2.2.1 **Microsoft FrontPage 2000**

The Microsoft FrontPage 2000 Web Site's creation and management tool can be used to create and manage exactly the site a user wants, either for creating a personal Web page or a corporate Internet or Intranet site. And because it works like the rest of Microsoft Office Premium, FrontPage 2000 is easy to learn and use.

These are the tasks that can be performed by using MS FrontPage 2000:

(i) **Create, Edit, Manage, and Update The Site**

- The multiple views in FrontPage 2000 allow me to see all the files in the my Web, run reports to find slow pages and older files, set up my site's navigational structure, and keep track of Web tasks
- For example, I can create and edit Web pages in Page View, use Folders View to see all the content on my Web, then set up how my pages link to one another in Navigation View

(ii) **Technical Articles**

- Learn about administration tools in FrontPage 2000 that enable me to set permissions and limit access to my Web sites

- Learn how to write a Microsoft Visual Basic for Applications (VBA) macro in FrontPage 2000, which adds a Save All command to my File menu so that I can save all my open files at once
- Learn how to write code that uses the FrontPage 2000 object model to change the properties of a theme

(iii) **Working with Databases and ASP**

- Learn how new and improved handling of Active Server Pages (ASP) enables me to safely create and edit .asp files in FrontPage
- Learn how to use FrontPage 2000 and ASP to remove a record from an Access database
- Learn how to use FrontPage 2000 and ASP to update database information in Access
- Learn how to set up automatic confirmation that my records have been saved successfully

(iv) **Web Resources on MSDN Online**

- Get the latest information about Internet technologies, including reference material and in-depth articles on all aspects of Web site design and development
- Read answers to users questions about a broad range of Web-related topics or send in own questions

(v) **Training and Certification**

- Visit the Microsoft Training & Certification site to find instructor-led classes on FrontPage 2000
- This instructor-led class teaches how to build dynamic, data-driven Web sites using HTML forms and Microsoft Access databases to collect and display information such as customer sales leads

(vi) **Using FrontPage to edit ASP pages**

- When an ASP page consists primarily of HTML, or contains large blocks of HTML, the FrontPage Normal View can be used to edit that HTML in the same way it can be used to edit any other HTML page
- When an ASP page consists primarily of program code, the page developer would likely need code editing features (such as syntax checking and coloring and statement completion) provided by rich program editors like Microsoft Script Editor or Visual Interdev
- When FrontPage loads an ASP page, it takes care to preserve the ASP code but it does not (and, logically, cannot) actually interpret the ASP code to determine what HTML it would generate
- When FrontPage loads any HTML page or ASP page, it needs to process the HTML into an in-memory data structure that represents the page. If the HTML tags are hidden within blocks of ASP, FrontPage cannot see the HTML tags and therefore cannot properly load the HTML

(vii) **Opens and Displays HTML files**

- When FrontPage opens a file, it reads the HTML tags in order to build an internal data structure representing the page and this process is called parsing
- During the parsing process, FrontPage will automatically handle common errors or omissions in the page (for example, multiple <BODY> tags or omitted end tags). This automatic error handling is designed to match the process that the browsers perform when reading the page. Some of this error handling is part of the HTML standard (for example, omit the </TR> tag before </TABLE>) and some is performed simply to be more robust in handling common errors that appear in hand-written HTML (such as multiple <BODY> tags)
- In all cases, FrontPage's behavior is designed to match the browser's behavior so that pages that can be loaded by the browser can also be properly loaded, displayed and edited by FrontPage
- Error handling during the parsing process is important when discussing ASP pages because some common ASP programming idioms can look like parsing errors. The presence of these errors in an ASP page may indicate problems in loading an ASP page into FrontPage's Normal View

3.2.2.2 **Microsoft Access 2000**

(i) **Creating Database**

1. Creating tables

   - Set the field name, data type, description, field properties, primary keys, etc.

2. Creating Queries

   - Set the field, which is needed.

   - Calculate the Amount of the field using criteria.

3. Creating Forms

   - Create and design the interface to let user to enter data.

   - Create the command button functionality to the user

4. Creating Reports

   - Set the outline what the user needed.

   - Design the print out of the data.

   - Calculate the amount of the field.

5. Creating Macros

   - Write the program to perform the command button.

   - Write the program to link the forms.

6. Creating Relationships

   - Link the table, which is related.

7. Creating Switchboard Manager

   - Creating the main interface for user to use

3.2.2.3 **Active Server Pages (ASP)**

Active Server Pages are components that allow web developers to create server-side scripted templates that generate dynamic, interactive web server applications. By embedding special programmatic codes in standard HTML pages, a user can access data in a database, interact with page objects such as Active-X or Java components, or create other types of dynamic output. The HTML output by an ASP is totally browser independent, which means that Microsoft Explorer, Netscape Navigator, or most other browsers can read it equally well.

ASP makes it easy for a developer to create everything from a web site that is customized to a viewer's taste, to complex database application that may access legacy data from a mainframe.

A very simple example of ASP:

```
<HTML>
<HEAD>
<TITLE>Sample Web Page</TITLE>
</HEAD>
<BODY>
<P>
Hello <%= Request.ServerVariables(“REMOTE_USER”) %>
The time here is <%= now %>
Your browser is <%= Request.ServerVariables(“http_user_agent”) %>
</BODY></HTML>
```

Active Server Pages can be used with any ODBC-compliant database such as FoxPro, Microsoft Access, Paradox, Microsoft SQL Server, etc.

Microsoft's Active Server Pages is a server-side scripting technology for building web pages that are both dynamic and interactive. An active server page itself is simply a text file script with the extension **.asp** containing HTML, client- and server-side script. The implementation behind it meanwhile was created by Microsoft and intended as an open technology server-side framework, giving web developers the freedom to develop dynamic web sites using information accessed from the many COM-compliant data sources available to them.

The syntax and grammar of ASP is easy to comprehend, and yet powerful enough to support some interaction between page user and server, allow web page access to databases and directory services, incorporate and make use of high-powered COM components.

A web application is really no more than a collection of ASP pages and Server Components and a website can contain several. The distinction between them is made by the definition of the application's starting-point or root directory within the site. All the content within this directory and the physical directory structure underneath it is considered to be part of the application until another application root directory is found. Thus the scope of an application is defined by the directories under the root not belonging to another application. The scope itself specifies the reach of changes to application state or to server related attributes.

Each application has its own set of variables and attributes that define its current state, and these are maintained throughout the lifetime of the application from the moment it is first run until the end of the last session is closed. As pointed out earlier, the problem with web-based applications is that the under-lying protocol of the web, HTTP, has less memory than a goldfish and retains no information from one client request to the next. ASP gets around this by using the **Application** and **Session** objects, to store information for the duration of the application or for the duration of a user's session respectively. Both application and session states are initialized and destroyed by code contained in the **global.asa** file. Global.asa can be found in the root directory of any web application. Code in Global.asa is triggered off at the initialization and termination of the application, and also at the beginning and end of each user session, governing the lifecycle of the state variables. Global.asa might also contain some 'global code' for use throughout the application.

3.2.2.4 **Microsoft Internet Information Server (IIS)**

IIS is a protocol server. It is implemented as a set of several system services that use the most common Internet protocols including HTTP, FTP, Network News Transfer Protocol (NNTP), and the Simple Mail Transfer Protocol (SMTP).

IIS also offers standard APIs for extending and customizing the server's capabilities. These APIs are known as the Common Gateway Interface (CGI) and the Internet Server

API (ISAPI). CGI is almost never used in IIS applications because it requires a new Win32 process for each HTTP request. Win32 process creation and destruction is relatively expensive, so this approach doesn't scale well in high-volume Web sites.

The ISAPI model was introduced to step in where CGI failed. ISAPI is based on the Win32 DLL architecture. Instead of loading a new process for each request, IIS loads an ISAPI-compliant DLL into its process and calls a well-known entry point to satisfy the HTTP request. Once the ISAPI DLL is loaded, it remains loaded until the IIS process is torn down. Hence, a single ISAPI DLL can satisfy all HTTP requests without any operating system overhead.

The downside to ISAPI is that it's difficult to implement correctly. Developing for ISAPI obviously has to use a language capable of creating ISAPI-compliant DLLs. While this is not directly possible from Visual Basic, creative programmers have developed mapping layers that allow Visual Basic code to participate in the ISAPI architecture. Nevertheless, most ISAPI developers stay with languages like C or C++ for better performance (no mapping layer) and more control over thread safety.

Because Microsoft didn't want to force Web developers to use these complex APIs directly, they introduced Active Server Pages (ASP) as a higher-productivity alternative. ASP is completely based on the ISAPI architecture. In fact, ASP is implemented as a system ISAPI DLL (asp.dll, also referred to as the ASP runtime). When a request for an

ASP page comes in, IIS makes sure that the ASP runtime is loaded and uses it to process the page.

ASP pages can contain both HTML and script from any scripting language that supports the ActiveX scripting interfaces, such as VBScript or JScript. Any HTML found in an ASP page is simply buffered back to the HTTP response. Script, however, is just-in-time compiled and executed using the appropriate scripting language engine. Because developers using Visual Basic can implement code for the ISAPI architecture with a familiar language like VBScript, ASP offers the best solution for high productivity and exceptional performance.