

ตัวแปลงภาษา HTML เป็น WML

HTML-to-WML Converter



นางสาววรรณิธร เทพสวัสดิ์
นางสาววรรณิตา วีระมนตรี

รพ.
๖๕๔๓๐๑
๕๕๔๓

เลขหมู่.....
เลขทะเบียน..... 42837
วัน, เดือน, ปี..... 10 มิ.ย. 2545

.b.....
.i.....

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๐๑๑๒๑๗๖๒๕

ตัวแปลงภาษา HTML เป็น WML

HTML-to-WML Converter

โดย

นางสาววรรณฉัตร เทพสวัสดิ์

นางสาววรรณิตา วีระมนตรี

อาจารย์ที่ปรึกษา

ดร.วิศิษฎ์ หิรัญกิตติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตัวแปลงภาษา HTML เป็น WML

HTML-to-WML Converter

ผู้จัดทำ

1. นางสาววรรณฉัตร เทพสวัสดิ์ รหัสประจำตัว 40010676
2. นางสาววรรณิตา วีระมนตรี รหัสประจำตัว 40010678



(ดร.วิศิษฎ์ หิรัญญิตติ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HTML-to-WML converter

Wannachat Thepsawat

Wannita Veeramontri

Dr. Visit Hirankitti Advisor

ABSTRACT

Main purpose of HTML to WML converter is to WAP devices can widely access information and services which are web services ,but WAP devices support only WML while almost information on World Wide Web is a format of HTML.

This thesis is concerned with the creation of fully automated converter from HTML to WML, that must have the knowledge about HTTP protocol, language structure of HTML, WML and XML, XSLT and Java language. The concept of conversion can be separated to 3 parts, the first part is Data Modification and Extraction from HTML, into XML by using Java application as a parser to convert an HTML document into a tree of XML elements. The second part is XML Manipulation using XSLT to access XML for defining XML display format. The last part is WML Generation and return a result page to browser that calls service

Developed conversion is implemented and compared advantage/disadvantage with other converters for improvement and extension.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดีหากไม่ได้รับความช่วยเหลือแลร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญาานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ วิศิษฐ์ หิรัญกิตติ อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณเพื่อนๆ ทุกคนที่ได้สละเวลามาทำการทดสอบตัวแปลงภาษา HTML เป็น WML ด้วยความตั้งใจ รวมทั้งต้องขอบคุณห้อง ICL ที่เอื้อเฟื้อสถานที่ให้ทำปริญญาานิพนธ์นี้

สุดท้ายขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

วรรณฉัตร เทพสวัสดิ์

วรรณิตา วีระมนตรี

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูปภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2 ทฤษฎี	2
2.1 โพรโตคอล HTTP (HyperText Transfer Protocol)	2
2.1.1 ความหมายของ HTTP	2
2.1.2 โครงสร้าง HTTP	4
2.1.3 เมธอด	11
2.2 ภาษามาร์กอัพ (Markup Language)	13
2.2.1 HTML (HyperText Markup Language)	13
2.2.2 XML (Extensible Markup Language)	13
2.2.3 WML (Wireless Markup Language)	16
2.3 เซิร์ฟเลต (Servlet)	16
2.3.1 ความหมายของเซิร์ฟเลต	17
2.3.2 เซิร์ฟเลตเอนจิน (Servlet engine)	18
2.3.3 อินเตอร์เฟซ javax.servlet.Servlet	19
2.3.4 วงจรชีวิตของเซิร์ฟเลต	20
บทที่ 3 หลักการในการออกแบบและสร้างตัวแปลงภาษา HTML เป็น WML	22
3.1 บทนำ	22
3.2 วิธีการในการแปลงเอกสาร HTML	22
3.2.1 การแปลงอัตโนมัติแบบสมบูรณ์	22
3.2.2 ตัวแปลงที่ให้ผู้ใช้งานเซตคอนฟิกูเรชันได้	22
3.2.3 การแปลงโดยการแก้ไขเอกสารต้นฉบับ	23
3.3 เครื่องมือช่วยในการแปลง	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
3.3.1 เฟรมเวิร์ก อาปาเช่ โคลน (Apache Cocoon)	24
3.4 หลักการทำงานของตัวแปลงภาษา	24
3.4.1 การตัดแบ่งและตัดแปลงเอกสาร	24
3.4.2 การจัดการกับ XML	29
3.4.3 การสร้างโค้ดภาษา WML	34
บทที่ 4 อิมพลีเมนเตชัน	36
4.1 การทำงานทั่วไป	36
4.2 การทำงานที่มีแท็กลิงก์หรือเฟรม	37
4.3 การทำงานที่มีแท็กฟอร์ม	38
บทที่ 5 ผลการทดลองและการเปรียบเทียบ	40
5.1 การใช้งาน	40
5.2 ตัวอย่างการแปลง	40
5.3 การเปรียบเทียบ	41
5.3.1 Phone.com WAP Gateway	41
5.3.2 Argo ActiGate	42
บทที่ 6 บทวิเคราะห์และสรุป	45
6.1 ข้อดี	45
6.2 ข้อเสีย	46
6.3 ข้อจำกัด	46
6.4 สรุป	46
6.5 แนวทางการพัฒนา	46
ภาคผนวก ก	48
ภาคผนวก ข	101
ภาคผนวก ค	121
บรรณานุกรม	122

สารบัญตาราง

	หน้าที่
ตารางที่ 3-1 แสดงการแมประหว่างแท็กภาษา XML เป็นภาษา WML	32
ตารางที่ 5-1 การเปรียบเทียบกับ Phone.com	42
ตารางที่ 5-2 การเปรียบเทียบกับ Argo ActiGate	43



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้าที่
รูปที่ 2-1 การติดต่อระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์	3
รูปที่ 2-2 ลักษณะข้อมูลที่รับส่งระหว่างไคลเอ็นต์และเซิร์ฟเวอร์	10
รูปที่ 2-3 หน้าที่และการทำงานของเซิร์ฟเลต	18
รูปที่ 2-4 เซิร์ฟเลตเอนจิน และเซิร์ฟเลตที่อยู่ภายใน	19
รูปที่ 2-5 วงจรชีวิตของเซิร์ฟเลต	20
รูปที่ 2-6 วิธีการจัดการกับการร้องขอแบบ GET และ POST ของเซิร์ฟเลต	21
รูปที่ 3-1 ขั้นตอนการแปลง HTML เป็น WML	22
รูปที่ 3-2 การแก้ไขเอกสาร HTML ดั้งฉบับ	23
รูปที่ 3-3 บางส่วนของเอกสาร HTML ดั้งฉบับ	25
รูปที่ 3-4 เอกสาร HTML หลังจากการตัดแบ่งและดัดแปลงให้ง่ายต่อการแปลงเป็น WML	26
รูปที่ 3-5 ตัวอย่างสตริงของเอกสาร HTML ดั้งฉบับ	27
รูปที่ 3-6 ตัวอย่างโครงสร้างต้นไม้ของ XML อลิเมนต์	28
รูปที่ 4-1 การทำงานต่างๆ ไปในการแปลง	36
รูปที่ 4-2 การร้องขอจากเว็บไคลเอ็นต์ไปยังเว็บเซิร์ฟเวอร์	37
รูปที่ 4-3 การทำงานในการแปลงแท็กฟอร์ม	38
รูปที่ 5-1 โสมเพจของ [HT → W]ML Converter	30
รูปที่ 5-2 การแสดงผลเว็บเพจ ภาษา HTML	30
รูปที่ 5-3 การแสดงผลเว็บเพจ ภาษา WML โดยการแปลงจาก HTML	31
รูปที่ 5-4 ตัวอย่างเปรียบเทียบการแสดงผลการแปลงเฟรม	43
รูปที่ 5-5 ตัวอย่างเปรียบเทียบการแสดงผลการแปลงตาราง	44
รูปที่ 5-6 ตัวอย่างเปรียบเทียบการแสดงผลการแปลงอินพุตฟอร์ม	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

2.1 โพรโทคอล HTTP (HyperText Transfer Protocol)

เป็นโพรโทคอลที่ใช้ในการติดต่อสื่อสารระหว่างเว็บเบราว์เซอร์และเว็บเซิร์ฟเวอร์ สำหรับเครือข่ายเวิลด์ไวด์เว็บ โดยใช้คอนเซ็ปต์ของไคลเอนต์-เซิร์ฟเวอร์เป็นหลัก หลักการทำงานพื้นฐานของ HTTP คือการส่งข้อความจากไคลเอนต์ไปยังเซิร์ฟเวอร์เพื่อทำการตีความ หลังจากนั้นเซิร์ฟเวอร์จะทำการส่งข้อความซึ่งเป็นผลจากการตีความกลับมายังไคลเอนต์โดยทั่วไปการส่งข้อความระหว่างไคลเอนต์และเซิร์ฟเวอร์มักใช้ ไบต์สตรีมเป็นหลัก แต่ HTTP เป็นโพรโทคอลที่ถูกออกแบบมาเพื่อให้ใช้งานง่ายแต่มีประสิทธิภาพ ดังนั้นจึงมีการใช้การส่งข้อมูลแบบเท็กซ์สตรีมแทน ความเข้าใจในหลักการทำงานขั้นพื้นฐานของ HTTP จะช่วยให้สามารถเขียนแอปพลิเคชันฝั่งเซิร์ฟเวอร์ เช่น Servlet, JSP, CGI, ASP, PHP, Cold Fusion ฯลฯ ได้อย่างมีประสิทธิภาพมากขึ้น อย่างไรก็ตามสำหรับผู้ที่อยากเขียน เว็บเซิร์ฟเวอร์, เว็บเซิร์ฟเวอร์หรือ Web Crawler อย่างง่าย ๆ

2.1.1 ความหมายของ HTTP

HTTP มาจากคำว่า Hypertext Transfer Protocol ซึ่งเป็นโพรโทคอลที่ใช้ในการส่งข้อมูลต่าง ๆ ในโลกของ เวิลด์ไวด์เว็บ ข้อมูลต่าง ๆ เหล่านี้โดยทั่วไปมักจะถูกเรียกว่า *Resource* โดย Resource เหล่านี้อาจจะเป็นไฟล์ เช่น ไฟล์ HTML, ไฟล์รูปภาพ หรือคำสั่งต่าง ๆ (Query String) เช่น คำสั่งที่ส่งไปที่ cgi ของโปรแกรมหรืออาจจะเป็นไบนารีสตรีมในกรณีของการดาวน์โหลดหรืออัปโหลดไฟล์ หรืออาจจะเป็นสิ่งอื่น ๆ อีกมากมายตามแต่จะกำหนดขึ้น

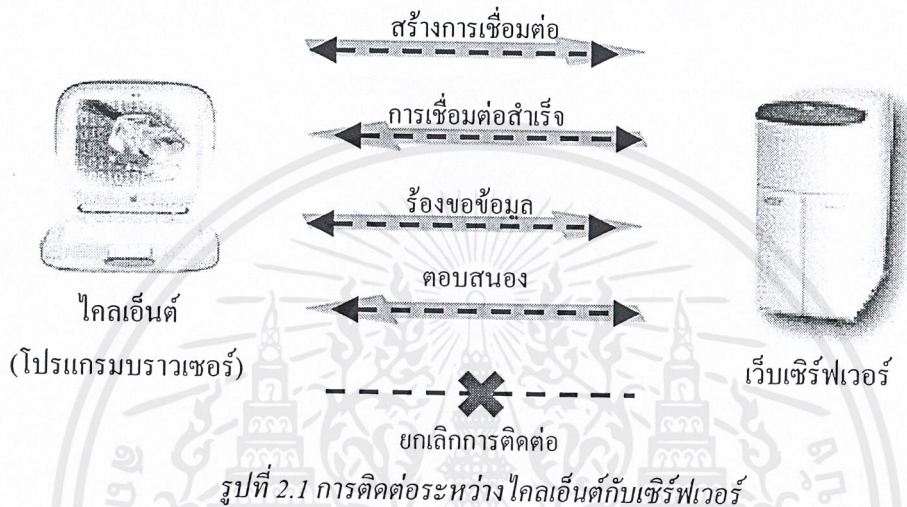
HTTP เป็นโพรโทคอลที่อยู่ในส่วนของเลเยอร์แอปพลิเคชัน ในโพรโทคอลสแต็กโดยข้อมูลต่าง ๆ จากเลเยอร์นี้จะถูกส่งผ่านไปยังเลเยอร์อื่น ๆ ที่ต่ำกว่าซึ่งส่วนหนึ่งในนั้นก็คือโพรโทคอล TCP/IP นั้นเอง (จะไม่กล่าวถึงรายละเอียดของโพรโทคอลสแต็กในที่นี้)

HTTP เป็นเน็ตเวิร์กโพรโทคอลที่ใช้หลักการของโมเดลไคลเอนต์-เซิร์ฟเวอร์ ในการติดต่อสื่อสารซึ่งหลักการทำงานอย่างคร่าว ๆ มีดังนี้

- HTTP ไคลเอนต์ จะทำการสร้างคอนเนกชันไปหา HTTP เซิร์ฟเวอร์ซึ่งโดยทั่วไปจะผ่านทางซ็อกเก็ตของ TCP/IP
- หลังจากนั้น HTTP ไคลเอนต์ จะทำการส่งคำสั่งการร้องขอ (request) ซึ่งอยู่ในรูปของข้อความ (เมสเซจ) ไปให้ HTTP เซิร์ฟเวอร์เพื่อทวงถามถึง resource ที่ต้องการ
- HTTP เซิร์ฟเวอร์จะทำการตีความคำสั่งที่ได้และส่งผลตอบสนองกลับ (response) ซึ่งเป็น resource ที่ HTTP ไคลเอนต์ ต้องการกลับมา (ผลที่ส่งกลับมาจะเป็นลักษณะของเมสเซจคล้ายกับการร้องขอของ HTTP ไคลเอนต์ ที่ส่งมาให้ HTTP เซิร์ฟเวอร์)

- หลังจากที่มีการส่งการตอบรับเสร็จสิ้น HTTP เซิร์ฟเวอร์จะทำการปิดคอนเน็กชันที่มาจาก HTTP โคลเอ็นต์
- ในกรณีที่ HTTP โคลเอ็นต์ ต้องการ resource อื่น ๆ, HTTP โคลเอ็นต์ จะต้องทำการสร้างคอนเน็กชันใหม่และส่งคำสั่งไปหา HTTP เซิร์ฟเวอร์อีกครั้ง

จากหลักการข้างต้นจะเห็นว่าการติดต่อสื่อสารระหว่างโคลเอ็นต์และเซิร์ฟเวอร์จะเป็นลักษณะครั้งต่อครั้ง ในทางเครือข่ายเรียกการติดต่อสื่อสารแบบนี้ว่า *Stateless Protocol*



ในการทำงานเว็บเบราว์เซอร์ก็คือ HTTP โคลเอ็นต์ นั่นเอง เหตุผลคือเว็บเบราว์เซอร์ถูกใช้เป็นตัวส่งการร้องขอไปที่เว็บเซิร์ฟเวอร์ (หรือ HTTP เซิร์ฟเวอร์) เพื่อจะรับ resource ที่ต้องการกลับมาโดยใช้โปรโตคอล HTTP โดยทั่วไป resource จะกระจายอยู่ตามโหนดของเครือข่ายต่าง ๆ ทั่วโลก สิ่งหนึ่งที่จะขาดไม่ได้ในการอ้างอิงถึง resource เหล่านี้คือ *URL (Universal Resource Locator)*. URL คือตัวที่ใช้ชี้ถึงแหล่งที่อยู่ของ resource ว่าอยู่ที่ไหน URL สามารถใช้เพื่ออ้างอิงถึง resource อะไรก็ได้ โดยใช้โปรโตคอลอะไรก็ได้ ไม่ใช่สำหรับ HTTP อย่างเดียวยกตัวอย่างเช่น

`http://161.246.5.113/index.html`

เป็น URL ของโปรโตคอล HTTP เพื่อใช้โหลดไฟล์ html

`http://161.246.5.113:80/coffee.jpg`

เป็น URL ของโปรโตคอล HTTP เพื่อใช้โหลดไฟล์รูปภาพ

`http://161.246.5.113/cgi-bin/sendMail.pl`

เป็น URL ของโปรโตคอล HTTP เพื่อใช้โหลดค่าที่ได้จากการประมวลผลของ cgi โปรแกรมชื่อ sendMail.pl

`ftp://ftp.jarticles.com/jspTutorial.doc`

เป็น URL ของ โปรโตคอล FTP เพื่อใช้ดาวโหลดไฟล์เอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างข้างต้น จะเห็นว่าในหนึ่ง URL จะประกอบไปด้วยชื่อของโพรโตคอล, ชื่อของเซิร์ฟเวอร์, พอร์ตที่ใช้และ resource ที่ต้องการ

2.1.2 โครงสร้าง HTTP (HTTP Structure)

จากหลักการการทำงานของ HTTP จะเห็นว่าสำหรับการติดต่อสื่อสารระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์แล้ว ตัวเมสเสจ (message) จะเป็นตัวกลางที่ใช้ในการติดต่อสื่อสารเสมอ และเพื่อให้ง่ายต่อการใช้งานรูปแบบของเมสเสจที่ใช้ในการร้องขอหรือตอบรับ จึงมีลักษณะคล้าย ๆ กัน โดยจะใช้เท็กซ์เป็นหลัก ซึ่งโครงสร้างของเมสเสจจะประกอบไปด้วย

1. บรรทัดเริ่มต้น (initial line)
2. เฮดเดอร์
3. บรรทัดว่าง (a blank line) ซึ่งก็คือ CRLF หรือการเว้นหนึ่งบรรทัด (วิธีหนึ่งที่ทำให้คือการกด Enter นั่นเอง)
4. เมสเสจบอดี ซึ่งอาจจะใช้บรรจุไฟล์, คำสั่ง (Query String) หรืออาจจะเป็นเอาต์พุตที่มาจากเซิร์ฟเวอร์โดยส่วนนี้จะมีหรือไม่มีก็ได้ขึ้นอยู่กับจุดประสงค์ของการใช้งาน

CRLF = Carriage return (\r) และ Line Feed (\n) ซึ่งก็คือ \u000A และ \u000D ใน ACSII นั่นเอง
รูปแบบโดยทั่วไปของเมสเสจจะเป็น

```
<initial line>
เฮดเดอร์1: value1
เฮดเดอร์2: value2
เฮดเดอร์3: value3
[blank line]
<เมสเสจ body, optional .....>
.....>
```

ตัวอย่างง่าย ๆ ของ เมสเสจ ที่ไคลเอ็นต์ใช้จะเป็น

```
GET /images/coffee.jpg <-- บรรทัดเริ่มต้น
```

```
From: soup@jarticles.com <-- เฮดเดอร์
```

```
User-Agent: Mozilla/4.72 <-- เฮดเดอร์
```

```
[blank line] <-- บรรทัดว่าง
```

หรือ

```
POST /servlet/searchEngine HTTP/1.0 <-- บรรทัดเริ่มต้น
```

```
From: webmaster@jarticles.com <-- เฮดเดอร์
```

```
User-Agent: Mozilla/4.72 <-- เฮดเดอร์
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[blank line] <-- บรรทัดว่าง
keyword=java&topic=jsp <-- เมสเซจบอดี
ตัวอย่างง่าย ๆ ของเมสเซจที่เซิร์ฟเวอร์ใช้อาจจะเป็น
HTTP 1.0 200 OK <-- บรรทัดเริ่มต้น
Date: Sunday, 23-July-00 04:01:12 GMT <-- เฮดเดอร์
Server: Apache/1.3.12(Unix) (Red Hat/Linux) PHP/3.0.15 mod_perl/1.21 <-- เฮดเดอร์
MIME-version: 1.0 <-- เฮดเดอร์
Content-type: text/html <-- เฮดเดอร์
Content-length: 115 <-- เฮดเดอร์
[blank line] <-- บรรทัดว่าง
<HTML><HEAD><TITLE>HTTTP Tutorial</TITLE></HEAD> <-- เมสเซจบอดี
<BODY>This is a tutorial, but please visit me again...</BODY> <-- เมสเซจบอดี(cont.)
<HTML> <-- เมสเซจบอดี(cont.)
```

2.1.2.1 บรรทัดเริ่มต้นในการร้องขอ (Initial Request Line)

บรรทัดเริ่มต้นของการร้องขอจะแตกต่างจาก บรรทัดเริ่มต้นของการตอบรับเล็กน้อยโดย บรรทัดเริ่มต้นของการร้องขอจะมีสามส่วนคือ

- ชื่อของเมธอด เช่น GET, POST, HEAD, TRACE
- โคลอเลพาธของ resource ที่ไคลเอ็นต์ต้องการ
- เวอร์ชันของ HTTP/x.x ที่ HTTP ไคลเอ็นต์ ใช้

สามส่วนนี้จะประกอบกันเป็นบรรทัดเริ่มต้น โดยแต่ละส่วนจะถูกแยกออกจากกันโดย

ใช้ช่องว่าง (space) ดังตัวอย่างข้างล่าง

```
GET /path/to/file/index.html HTTP/1.0
```

ตัวอย่างนี้ใช้ method ที่ชื่อ GET เพื่อขอ resource (ในที่นี้คือไฟล์) ชื่อ /path/to/file/index.html โดยใช้ HTTP/1.0 protocol

การใช้โคลอเลพาธ (local path)

ก่อนที่ HTTP ไคลเอ็นต์ จะส่งการร้องขอไปที่ HTTP เซิร์ฟเวอร์, HTTP ไคลเอ็นต์ จะสร้างคอนเน็กชันขึ้นมาอันหนึ่งก่อน โดยใช้ชื่อเรียก ใน การสร้างชื่อเรียกจะต้องทำการกำหนดชื่อของโฮสต์ที่จะติดต่อ ยกตัวอย่างเช่น ถ้า url เป็น http://161.246.5.113/path/to/file/index.html ชื่อของโฮสต์ก็จะเป็น 161.246.5.113 ดังนั้นการกำหนดเพียง โคลอเลพาธเพื่อบ่งบอกถึง resource ที่ต้องการในส่วน of บรรทัดเริ่มต้นในการร้องขอ ซึ่งก็คือ /path/to/file/index.html ก็เพียงพอแล้ว

API ที่ใช้ในแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์ มักอ้างถึงส่วนที่เป็นโลคอลพาทนี้ว่า *requested URI* (ตัว I มาจาก Identifier)

ข้อระวัง

- ชื่อของเมธอดจะต้องใช้ตัวใหญ่เสมอ
- เวอร์ชันของ HTTP จะอยู่ในรูป HTTP/x.x และจะต้องเป็นตัวใหญ่เสมอ
- บรรทัดเริ่มต้นในการตอบสนอง (Initial Response Line)

โดยทั่วไป บรรทัดเริ่มต้นในการตอบสนองมักจะเรียกว่า status line ซึ่งประกอบไปด้วยสามส่วนย่อย คือ

- เวอร์ชันของ HTTP/x.x ที่ เซิร์ฟเวอร์ใช้สำหรับส่งเมสเซจ
- response status code ซึ่งเป็นตัวบอกว่าผลของการร้องขอที่ไคลเอนต์ส่งมาเป็นอย่างไร
- reason phase เป็นตัวอธิบายความหมายของ response status code อีกทีหนึ่ง ยกตัวอย่างเช่น

HTTP/1.0 200 OK

หรือ

HTTP/1.0 404 Not Found

ข้อสังเกต

- เวอร์ชันของ HTTP จะต้องอยู่ในรูปของ HTTP/x.x และจะต้องเป็นตัวใหญ่เสมอ
- โค้ดสถานะการตอบสนอง (response status code) เป็นโค้ดที่ส่งมาให้คอมพิวเตอร์ฝั่งไคลเอนต์อ่านซึ่งจะมี reason phase เป็นตัวอธิบายให้มนุษย์ (Human Being) อ่านอีกทีหนึ่ง
- โค้ดสถานะการตอบสนองจะอยู่ในลักษณะของเลขสามหลัก โดยหลักแรกจะบอกถึงความหมายโดยทั่ว ๆ ไปของ response
- 1xx เป็น code ที่ใช้บอกถึงเหตุการณ์ต่าง ๆ ที่เกิดขึ้นในการสื่อสารระหว่างไคลเอนต์และ เซิร์ฟเวอร์
- 2xx เป็น code ที่บ่งบอกว่า request ที่ส่งจาก ไคลเอนต์ถูกทำให้ complete แล้ว
- 3xx เป็น code ที่บอกให้ไคลเอนต์ทำการ redirect ไปที่ url อื่นแทน
- 4xx เป็น code ที่บ่งบอกถึงความผิดพลาดที่เกิดขึ้นจากส่วนของไคลเอนต์
- 5xx เป็น code ที่บ่งบอกถึงความผิดพลาดที่เกิดขึ้นจากส่วนของ เซิร์ฟเวอร์โค้ดแสดงสถานะ (status code)

โค้ดสถานะที่พบเห็นโดยทั่วไป คือ

- 100 Continue หมายถึง เซิร์ฟเวอร์ได้รับ request บางส่วนจากไคลเอนต์แล้ว, ไคลเอนต์กรุณาส่งส่วนที่เหลือมาให้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 200 OK หมายถึง คำร้องขอที่มาจากไคลเอ็นต์ถูกต้องและ resource ที่ไคลเอ็นต์ต้องการอยู่ใน เมสเซจบอดีของ response นี้แล้ว
- 301 Moved Permanently หมายถึง resource ที่ไคลเอ็นต์ต้องการเคยอยู่ที่ เซิร์ฟเวอร์นี้แต่ถูกย้ายไปอยู่ที่อื่นแล้ว
- 302 Moved Temporarily หมายถึง resource ที่ไคลเอ็นต์ต้องการถูกย้ายไปอยู่ที่เซิร์ฟเวอร์อื่นหรือไม่สามารถเข้าถึงได้ชั่วคราว
- 400 Bad Request หมายถึง เซิร์ฟเวอร์ไม่สามารถเข้าใจการร้องขอที่ไคลเอ็นต์ส่งมา
- 403 Forbidden หมายถึง เซิร์ฟเวอร์เข้าใจการร้องขอที่ไคลเอ็นต์ส่งมาแต่ไม่ต้องการที่จะส่ง resource ที่ไคลเอ็นต์ต้องการกลับไปให้
- 404 Not Found หมายถึง เซิร์ฟเวอร์ไม่มี resource ที่ไคลเอ็นต์ต้องการ
- 500 เซิร์ฟเวอร์ Error หมายถึง เกิดข้อผิดพลาดขึ้นในส่วนของเซิร์ฟเวอร์ (โดยทั่วไปข้อผิดพลาดนี้จะเกิดขึ้นในกรณีที่การร้องขอส่งมาเรียกแอปพลิเคชันฝั่งเซิร์ฟเวอร์ที่รันบนเซิร์ฟเวอร์นี้แต่เกิดข้อผิดพลาดขึ้นระหว่างการประมวลผลซึ่งโดยทั่วไปจะเกิดขึ้นเพราะ โปรแกรมมีบั๊ก)

2.1.1.2.3 เฮดเดอร์ (header)

เฮดเดอร์จะเป็นส่วนที่บอกถึงรายละเอียดของการร้องขอ (ที่กำลังส่งไปให้เซิร์ฟเวอร์) หรือ การตอบรับ (ที่กำลังส่งกลับมายังไคลเอ็นต์) ยกตัวอย่างเช่น

- ไคลเอ็นต์จะส่ง เฮดเดอร์ ที่บอกถึงชนิดและขนาดของข้อมูลที่อยู่ข้างใน เมสเซจบอดี ในกรณีที่ไคลเอ็นต์ต้องการ upload ไฟล์ไปยัง เซิร์ฟเวอร์
- เซิร์ฟเวอร์อาจจะส่ง เฮดเดอร์ ที่เกี่ยวกับชนิดและขนาดของ resource ที่ไคลเอ็นต์กำลังจะได้รับโดยใช้ Content-Type และ Content-Length
- เฮดเดอร์จะมีรูปแบบเป็นเท็กซ์ โดยในหนึ่งบรรทัดจะถูกใช้สำหรับหนึ่งเฮดเดอร์ซึ่งจะอยู่ในรูปของ "เฮดเดอร์-Name: value" และจบด้วย CRLF ยกตัวอย่างเช่น

From: admin@ce.kmitl.ac.th

User-Agent: Mozilla/4.72

Content-Type: text/html

Content-Length: 250

ข้อสังเกต

- ชื่อของเฮดเดอร์สามารถใช้ตัวใหญ่หรือตัวเล็กก็ได้
- สามารถเว้นช่องว่างหรือ tab ระหว่าง ":" ของชื่อเฮดเดอร์และค่าของมัน กว้างเท่าไรก็ได้

- ใน HTTP1.0 จะมีเฮดเดอร์กำหนดไว้ 16 แบบ แต่ของ HTTP1.1 จะมีถึง 46 แบบด้วยกัน

เฮดเดอร์ ที่มักพบเห็นทั่วไปในส่วนของไคลเอ็นต์

From: เป็น เฮดเดอร์ ที่ใช้สำหรับเก็บอีเมลแอดเดรสของผู้ที่กำลังส่งการร้องขอ ยกตัวอย่างเช่น From: admin@ce.kmitl.ac.th

User-Agent: เป็น เฮดเดอร์ที่ใช้บ่งบอกถึงชื่อของโปรแกรมที่ใช้เป็น HTTP ไคลเอ็นต์ ยกตัวอย่างเช่นยูสเซอร์

Agent: Mozilla/4.72

เฮดเดอร์ที่มักพบเห็นทั่วไปในส่วนของ เซิร์ฟเวอร์

Server: เป็น เฮดเดอร์ ที่จะคล้ายกับ User-Agent แต่เป็นตัวบอกลิงชื่อของโปรแกรมที่ใช้เป็น HTTP เซิร์ฟเวอร์ยกตัวอย่างเช่น Server: Apache/1.93

Last-Modified: เป็น เฮดเดอร์ ที่ใช้บอกลิงเวลาครั้งสุดท้ายที่ resource ที่ไคลเอ็นต์ต้องการถูกเปลี่ยนแปลง (modify/update) โดยเวลาที่ใช้จะเทียบจาก GMT (Greenwich Mean Time) ยกตัวอย่างเช่น เวลาของเมืองไทยถือว่าเป็น GMT+7.00 ถ้าตอนนี้เวลาในเมืองไทยคือ Sun, 23 July 2000 20:39:56 เวลาที่เป็น GMT ก็คือ Last-Modified: Sun, 23 July 2000 13:39:56 GMT

2.1.2.4 เมสเซจบอดี (message body)

เมื่อไรก็ตามที่ไคลเอ็นต์หรือเซิร์ฟเวอร์ต้องการส่งข้อมูลไปกับเมสเซจ ส่วนที่เป็นเมสเซจบอดีจะเป็นส่วนที่ใช้สำหรับเก็บข้อมูลดังกล่าว ในกรณีของไคลเอ็นต์ตัวเมสเซจบอดีอาจใช้สำหรับบรรจุไฟล์ที่ต้องการอัปโหลดหรือใช้สำหรับเก็บข้อมูลที่มาจากอิลิเมนต์ต่าง ๆ ของฟอร์ม HTML แล้วส่งไปยังเซิร์ฟเวอร์ก็ได้ ในกรณีของเซิร์ฟเวอร์ตัวเมสเซจบอดีจะเป็นส่วนที่ใช้เก็บ resource ที่ไคลเอ็นต์ทวงถามหรืออาจใช้สำหรับเก็บคำอธิบายต่าง ๆ ในกรณีที่มีข้อผิดพลาดเกิดขึ้นก็ได้ ถ้าเมสเซจมีส่วนของเมสเซจบอดี ไคลเอ็นต์หรือเซิร์ฟเวอร์มักจะเพิ่มเฮดเดอร์ที่ช่วยบอกลิงรายละเอียดของเมสเซจบอดีดังกล่าวด้วย ยกตัวอย่างเช่นถ้า เซิร์ฟเวอร์ต้องการส่ง resource ที่เป็นไฟล์รูปภาพมาให้ไคลเอ็นต์ เฮดเดอร์ ที่ถูกเพิ่มขึ้นมาอีกจะเป็น Content-Type: image/gif เป็น เฮดเดอร์ ที่บ่งบอกถึง MIME type ของข้อมูลที่อยู่ใน เมสเซจบอดี Content-Length: 1026 เป็น เฮดเดอร์ที่บ่งบอกถึงขนาดของ เมสเซจบอดีในหน่วยไบต์

ตัวอย่างของการส่ง เมสเซจ ระหว่างไคลเอ็นต์และ เซิร์ฟเวอร์

<http://161.246.5.113/tutorials/basic/helloworld.html>

(Client = Web Browser, เซิร์ฟเวอร์= Web Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นแรกไคลเอ็นต์จะทำการเรียกใช้ชื่อเกิดเพื่อคอนเน็กไปที่เซิร์ฟเวอร์ชื่อ 161.246.5.113 ซึ่งโดยปกติจะติดต่อไปที่พอร์ต 80 ซึ่งเป็นพอร์ตดีฟอลต์ของโพรโตคอลHTTP (ในกรณีที่ต้องการคอนเน็กไปที่พอร์ตอื่นก็สามารถทำได้โดยการเพิ่มส่วนของพอร์ตเข้าไปหลังจากส่วนของโฮตส์เนม เช่น <http://161.246.5.113:8080/tutorials/advance/hellocorba.html> ไคลเอ็นต์จะคอนเน็กไปที่พอร์ต8080 แทน)

หลังจากนั้นไคลเอ็นต์จะทำการส่งเมสเสจร้องขอ

```
GET /tutorials/basic/helloworld.html HTTP/1.0
```

```
From: soup@jarticles.com
```

```
User-Agent: Mozilla/4.72
```

```
[blank line]
```

หลังจากที่เซิร์ฟเวอร์ได้รับเมสเสจข้างบนแล้ว เซิร์ฟเวอร์จะส่งการตอบรับพร้อมทั้ง resource ที่ไคลเอ็นต์ต้องการกลับมาดังนี้

```
HTTP/1.0 200 OK
```

```
Date: Sunday, 23-July-00 04:01:12 GMT
```

```
Content-type: text/html
```

```
Content-length: 1556
```

```
[blank line]
```

```
<HTML><HEAD><TITLE>HTTP Tutorial</TITLE></HEAD>
```

```
<BODY>This is HelloWorld tutorial, but please read it for fun : )
```

```
(more file contents)
```

```
...
```

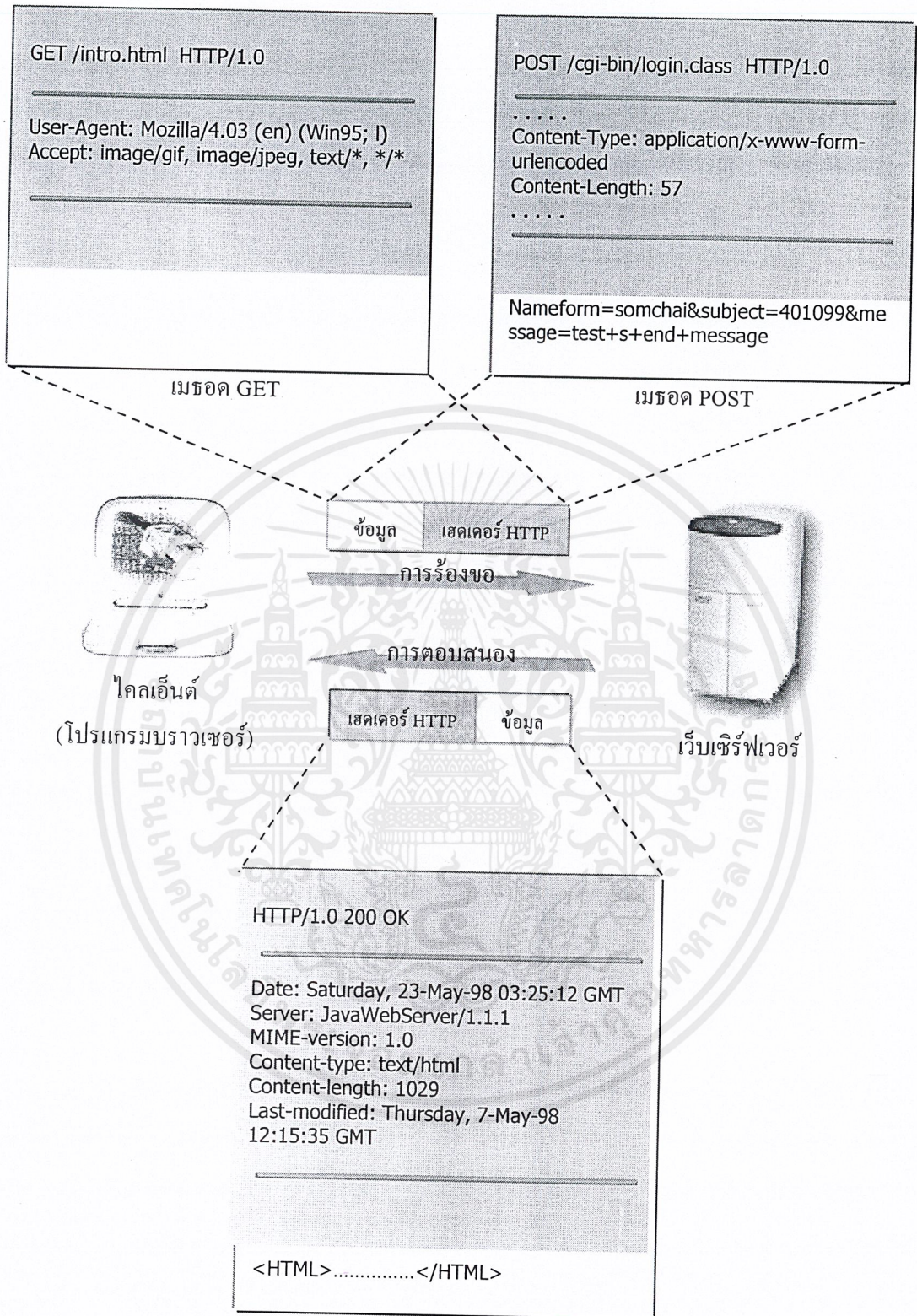
```
...
```

```
...
```

```
</BODY>
```

```
<HTML>
```

ท้ายสุด เซิร์ฟเวอร์จะทำการปิดคอนเน็กชันของไคลเอ็นต์



รูปที่ 2.2 ลักษณะข้อมูลที่รับส่งระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 เมธอด

2.1.3.1 เมธอด GET

ยกตัวอย่างเช่น <http://www.google.com/search.cgi?keyword=computer> เป็นการค้นหาที่ www.google.com โดยใช้คีย์เวิร์ด= "computer"

ถ้าพิจารณาส่วนที่อยู่ต่อจากส่วนของโฮสต์เนม(www.google.com) แล้วจะเห็นว่า URI ที่ถูกร้องขอ (/search.cgi?keyword=computer) ที่เห็นไม่ได้เป็นไฟล์ html ไฟล์อย่างทั่ว ๆ ไป แต่กลับกลายเป็นชื่อของโปรแกรม (search.cgi), เครื่องหมายคำถาม (?) และคีย์เวิร์ดที่ใช้ค้นหาแทน อย่างที่กล่าวมาแล้วใน ส่วนต้นของบทความว่า resource จริง ๆ แล้วก็คือตัวที่ใช้บ่งชี้ถึงข้อมูลที่อยู่ที่เซิร์ฟเวอร์ resource อาจเป็นไฟล์หรืออาจเป็น query string ที่ใช้ส่งไปเรียกโปรแกรมที่อยู่ที่ เซิร์ฟเวอร์ก็ได้ ซึ่งในกรณีของตัวอย่าง Search Engine ข้างต้นตัว resource ก็คือโปรแกรมที่ชื่อ search.cgi นั่นเอง โดยทั่วไป การตอบรับที่ได้กลับมาจาก เซิร์ฟเวอร์อาจจะอยู่ในรูปของไฟล์ถ้า resource ที่ร้องขอเป็นไฟล์ ยกตัวอย่างเช่น <http://161.246.5.113/tutorial/http.html> หรืออาจจะเป็นเอาต์พุตที่ได้จากการประมวลผลของโปรแกรมที่ผ่าน query string เข้าไป เช่นถ้าส่ง query string ไปที่ search engine ผลที่จะได้กลับมาก็คือ ลิงก์ต่าง ๆ ที่เกี่ยวข้องกับ คีย์เวิร์ดที่ใช้ค้นหานั้นเอง

โดยทั่วไปเว็บเบราว์เซอร์(ไคลเอ็นต์) จะใช้เมธอด GET ในการส่งเมสเสจการร้องขอไปที่เซิร์ฟเวอร์ในกรณีที่ resource ที่ต้องการเป็นไฟล์ อย่างไรก็ตาม GET ยังสามารถใช้ในการส่ง query string สั้น ๆ ไปยังโปรแกรมที่รันอยู่ที่ เซิร์ฟเวอร์ได้อีกด้วย ยกตัวอย่างเช่น ถ้าต้องการส่งข้อมูลชื่อคีย์เวิร์ดโดยมีค่าเท่ากับ computer ไปที่ เซิร์ฟเวอร์ชื่อ www.google.com โดยจะส่งไปที่ resource ซึ่งเป็น cgi โปรแกรม ชื่อ search.cgi วิธีการทำก็คือ

- กำหนด resource ลงไปในส่วนของ request URI ซึ่งจะอยู่หลังจากส่วนของโฮสต์เนมซึ่งจะได้ <http://www.google.com/search.cgi>
- ใส่ตัวเครื่องหมายคำถามเพื่อบอก เซิร์ฟเวอร์ว่าส่วนที่เหลือถัดไปจะเป็นส่วนของข้อมูลซึ่งจะได้ <http://www.google.com/search.cgi?>
- ทำการเข้ารหัสโดยวิธีการที่เรียกว่า *URL-encoding* ไปที่ชื่อและค่าของตัวแปรต่าง ๆ ซึ่งในกรณีของตัวแปรก็คือคีย์เวิร์ดซึ่งจะได้ <http://www.google.com/search.cgi?keyword=name>

ถ้าหากพิมพ์ URL ข้างบนเข้าไปในเว็บเบราว์เซอร์แล้วกดปุ่ม Enter ตัวเว็บเบราว์เซอร์จะตีความ URL ดังกล่าวซึ่งผลที่ได้ก็คือการใช้ GET ส่งเมสเสจร้องขอไปที่เซิร์ฟเวอร์ที่ชื่อ www.google.com โดยจะแนบคีย์เวิร์ด "computer" ไปด้วยกับการร้องขอซึ่งบรรทัดเริ่มต้นของเมสเสจการร้องขอจะเป็นอะไรคล้าย ๆ ข้างล่างนี้

```
GET /search.cgi?keyword=jarticles HTTP/1.0
```

```
From: soup@jarticles.com
```

User-Agent: Mozilla/4.72

[blank line]

หลังจากนั้นจะได้ลิงก์ต่าง ๆ ที่เกี่ยวกับ computer กลับมา สังเกตว่า URL จริง ๆ ที่ใช้กับ www.google.com จะต่างจาก URL ข้างบนเล็กน้อย วิธีการส่ง query string ไปกับ URL เหมาะสำหรับ query string ที่ไม่ยาวมากนัก โดยทั่วไปความยาวของ URL มักจะจำกัดอยู่ที่ 80 คาแร็กเตอร์ ซึ่งโดยทั่วไปแล้วตัวอักษรที่ตัดจากนั้นจะถูกตัดออกไปโดยอัตโนมัติ (จะตัดหรือไม่ตัดจริง ๆ ขึ้นอยู่กับการอิมพลีเม้นต์ของ เซิร์ฟเวอร์แต่ละตัว)

2.1.3.2 การเข้ารหัส URL (URL-Encoding)

ในการส่งข้อมูลไปที่ เซิร์ฟเวอร์โดยวิธีการ GET หรือวิธีการ POST (จะพูดถึงถัดไป) นั้นชื่อและค่าของตัวแปรต่าง ๆ ซึ่งอยู่หลังจากเครื่องหมายคำถามในกรณีของ GET หรืออยู่ในส่วนของ เมสเสจบอดีในกรณีของ POST จะต้องถูกผ่านการเข้ารหัสโดยวิธีการที่เรียกว่า *URL-Encoding* เสียก่อน โดยทั่วไปตัว URL เองจะมีคาแร็กเตอร์บางตัวที่ใช้สำหรับความหมายพิเศษ เช่น :, /, ~ & ? ซึ่งในบางครั้งชื่อและค่าของตัวแปรต่าง ๆ ที่ถูกส่งไปที่ เซิร์ฟเวอร์อาจจะมีคาแร็กเตอร์เหล่านี้ปะปนอยู่ด้วย เพื่อหลีกเลี่ยงความสับสนในการตีความของเซิร์ฟเวอร์ คาแร็กเตอร์ต่าง ๆ ที่เป็นที่ใช้ใน URL จะต้องถูกการเข้ารหัสเสียก่อน โดยมีหลักการดังต่อไปนี้

- ให้ทำการเปลี่ยน unsafe characters ต่าง ๆ ที่อยู่ชื่อและค่าของตัวแปรให้กลายเป็น "%xx" โดย "xx" คือค่าของแอสกีของคาแร็กเตอร์นั้น ๆ ในแบบเลขฐาน 16 ซึ่ง คาแร็กเตอร์ unsafe เหล่านี้รวมไปถึง =, &, %, +, และคาแร็กเตอร์ ต่าง ๆ ที่ไม่สามารถพิมพ์ได้หรือแม้กระทั่ง characters ที่ต้องการจะการเข้ารหัสเอง
- ให้ทำการเปลี่ยนช่องว่าง (space) ทั้งหมดให้กลายเป็นเครื่องหมายบวก (+)
- จับคู่ชื่อและค่าของตัวแปรต่าง ๆ ด้วย =
- ถ้ามีชื่อและค่าของตัวแปรมากกว่าหนึ่งตัว ให้นำชื่อและค่าของตัวแปรแต่ละคู่มาเชื่อมติดกันด้วยเครื่องหมาย &
- นำชื่อและค่าของตัวแปรที่เชื่อมติดกันทั้งหมดไปใส่ที่ URL โดยมีเครื่องหมาย ? อยู่ข้างหน้า(ในกรณีของ GET) หรือนำไปใส่ที่ เมสเสจบอดี(ในกรณีของ POST) ยกตัวอย่างเช่น ถ้ามีชื่อและค่า (name, value) = ("keyword", "jarticles") (name, value) = ("name", "Soup & friends") หลังจากทำการการเข้ารหัสจะได้ String ออกมาเป็น keyword=jarticles&name=Soup+%26+friends โดยสตริงนี้มีความยาวเท่ากับ 39 โดยที่ & = %26 in hex

2.1.3.3 เมธอด Post

บางครั้งหากไม่ต้องการส่งข้อมูลไปยังเซิร์ฟเวอร์ด้วยการติดข้อมูลเหล่านั้นไปกับ URL ด้วยเหตุผลที่ว่าข้อมูลเหล่านั้นอาจเกี่ยวข้องกับข้อมูลส่วนตัว ยกตัวอย่างเช่น login และ password ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตัวผ่านเข้าไปในเว็บไซด์ต่าง ๆ ซึ่งในกรณีที่ใช้ GET, URL หลังจากที่ทำกร login ไปแล้ว URL ที่อยู่ตรง Address Bar อาจจะออกมาเป็น `http://www.myserver.com/login.cgi?login=me&password=youandme` ซึ่งอาจคนเห็นและนำข้อมูลไปใช้อย่างผิดๆ ได้ อีกกรณีหนึ่งคือกรณีของข้อมูลที่ติดไปกับส่วนของ URI ที่ถูกร้องขอทำให้ URL มีความยาวมากกว่าความยาวของ URL ที่ เซิร์ฟเวอร์กำหนดไว้ ผลกระทบที่อาจเกิดขึ้นคือข้อมูลบางส่วนของอาจหายไปเนื่องจากการตัดทอนโดย เซิร์ฟเวอร์

วิธีการที่สามารถใช้เพื่อปิดบังข้อมูล (แบบง่าย ๆ) หรือเพื่อส่งข้อมูลที่มีความยาวมาก ๆ ก็คือการใช้ เมธอด POST โดยที่ POST ต่างกับ GET ตรงที่ POST จะไม่ทำการติดข้อมูลไปกับ URL แต่ POST จะทำการใส่ข้อมูลเข้าไปในส่วนของ เมสเซจบอดีของเมสเซอร์ร้องขอแทนซึ่งในกรณีนี้ URI ที่ถูกร้องขอจะมีปรากฏเพียงชื่อของโปรแกรมที่ต้องการส่งข้อมูลไปให้เท่านั้น ในการส่งข้อมูลไปในส่วนของ เมสเซจบอดี เพื่อให้ง่ายต่อการตีความโดยเซิร์ฟเวอร์ทางเมธอด POST จึงมีการบังคับให้ใส่เฮดเดอร์พิเศษเพื่อบรรยายรายละเอียดต่างๆ ของข้อมูลที่อยู่ในเมสเซจบอดีนั้นอีกด้วยซึ่งโดยทั่วไปเฮดเดอร์ที่มักจะถูกเพิ่มเข้าไปก็คือ Content-Type และ Content-Length การใช้ POST ที่มักพบเห็นโดยทั่วไปคือการใช้ POST ในการส่งข้อมูลที่มาจากรูปแบบ HTML สังเกตว่า ในกรณีของการส่งข้อมูลจากรูปแบบ HTML ไคลเอ็นต์จะทำการเซต Content-Type เป็น `application/x-www-form-urlencoded` และ Content-Length จะมีค่าเท่ากับ ความยาวของชื่อและค่าของอิตีเมนต์ฟอร์ม ต่าง ๆ ที่อยู่ในฟอร์มของ HTML รวมกันซึ่งจากตัวอย่างก็คืออิตีเมนต์ฟอร์มที่ชื่อ `userName` และ `password` นอกจากนี้ POST ยังสามารถใช้ในการส่งข้อมูลแบบอื่น ๆ ได้อีก โดยรูปแบบจะขึ้นอยู่กับวิธีการส่งที่ทางเว็บเบราว์เซอร์และ เว็บเซิร์ฟเวอร์ได้ทำการตกลงกันไว้ ยกตัวอย่างเช่น ในกรณีของไฟล์อัปโหลดตัว Content-Type ก็จะถูกเซตเป็น `Multipart/form-data` แทน

2.2 ภาษามาร์กอัพ (Markup Language)

2.2.1 HTML (HyperText Markup Language)

HTML เป็นส่วนหนึ่งของภาษา SGML (Standard Generalized Markup Language) ที่มีโครงสร้างการใช้งานแบบง่าย ๆ โดยภาษา HTML มีไว้สำหรับใช้สร้างเอกสารแบบไฮเพอร์เท็กซ์ของรูปแบบของเอกสารหรือข้อมูลส่วนต่างๆ ของเอกสารนั้นจะสนับสนุนทั้งที่เป็นข้อความ, ภาพ, เสียง, วิดีโอ และอื่น ๆ โดยการใช้งานนั้นสามารถเชื่อมโยงเอกสารหรือข้อมูลต่าง ๆ นี้ได้ โดยจุดที่แสดงการเชื่อมโยงนี้เองที่เรียกว่าการมาร์กอัพ

2.2.2 XML (Extensible Markup Language)

XML ถูกกำหนดโดยกลุ่มทำงาน XML ของสถาบัน World Wide Web Consortium (W3C) กลุ่มทำงานกลุ่มนี้ได้บรรยายถึงภาษา XML ไว้ว่า “Extensible Markup Language (XML) เป็น เซตย่อยของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGML จุดมุ่งหมายคือให้สามารถใช้ SGML ในการบริการรับข้อมูลและประมวลผลบนเครือข่ายอินเทอร์เน็ตซึ่งปัจจุบันสามารถทำได้ด้วยภาษา HTML เท่านั้น XML ถูกออกแบบมาเพื่อให้เป็นวิธีการที่ง่ายและสามารถประมวลผลร่วมกันทั้ง SGML และ HTML “

XML เป็นข้อความที่ถูกกำหนดขึ้นมาอย่างง่าย ๆ ด้วยตัวอักษรพิเศษ ดังเช่น ภาษา HTML หรือ WML แต่ข้อแตกต่างก็คือ HTML และ WML ต้องใช้อิเลเมนต์ที่ได้กำหนดความหมายไว้ล่วงหน้าแล้วโดย HTML เป็นแอปพลิเคชัน SGML ขณะที่ WML เป็นแอปพลิเคชันของ XML

SGML ย่อมาจาก Generalized Markup Language เป็นภาษาซับซ้อนขั้นสูงและยังเป็นต้นแบบภาษามาร์คอัพ (Markup) ทั้งหมด ทั้ง HTML และ XML ต่างก็สืบทอดมาจากภาษา SGML (ถึงแม้ว่าโดยพื้นฐานจะมีแนวทางที่แตกต่างกัน) SGML มีการจำกัดความไวยากรณ์พื้นฐาน แต่อนุญาตให้สร้างอิเลเมนต์เป็นของตัวเองได้ (เพราะฉะนั้น SGML จึงครอบคลุมทั่วไป) ดังนั้นการใช้ SGML เพื่ออธิบายเอกสารเฉพาะ ต้องออกแบบอิเลเมนต์และโครงสร้างเอกสารให้เหมาะสม

แม้ว่า SGML จะทำให้การสร้างประโยคของภาษาถูกกำหนดอย่างถูกต้องและสมบูรณ์ แต่อย่างไรก็ตาม มันไม่สามารถระบุสิ่งที่อิเลเมนต์นั้นหมายความว่าอย่างไรหรืออีกนัยหนึ่งคือ มันไม่สามารถบอกได้ว่าอิเลเมนต์จะถูกจัดการได้อย่างไรในอุปกรณ์ที่สนับสนุนภาษานั้น ยกตัวอย่างเช่น SGML สามารถทำให้กำหนดได้เองว่า ใน <table> จะประกอบด้วยอิเลเมนต์ <tr> และ <td> ในขณะที่ข้อกำหนดของ HTML จะบอกได้ว่าผลลัพธ์ที่เกิดขึ้นในการแสดงผลบนเว็บเบราว์เซอร์เป็น ตาราง ถ้า HTML ไม่สามารถระบุความหมายของแต่ละอิเลเมนต์ เว็บเบราว์เซอร์จะไม่สามารถรู้ได้เลยว่าจะต้องทำอย่างไรกับมัน

วัตถุประสงค์ทั่วไปของชุดอิเลเมนต์ที่ใช้ในการอธิบายรูปแบบเฉพาะของเอกสาร ถูกเรียกว่า โปรแกรมประยุกต์ SGML ซึ่งสามารถจำกัดความโปรแกรมประยุกต์ SGML เองได้ เพื่ออธิบายรูปแบบเฉพาะของเอกสารที่ใช้ทำงานด้วย หรือรูปร่างมาตรฐานที่สามารถจำกัดความโปรแกรมประยุกต์ SGML เพื่ออธิบายรูปแบบเอกสารที่ถูกใช้อย่างแพร่หลาย ตัวอย่างโปรแกรมประยุกต์ที่มีชื่อเสียงที่ใช้กันแพร่หลายก็คือ HTML ซึ่งเป็นโปรแกรมประยุกต์ SGML ที่ถูกพัฒนาขึ้นในปี 1991 เพื่ออธิบายเอกสารเว็บ SGML อาจจะเป็นภาษาที่ขยายการใช้งานมาร์คอัพได้อย่างสมบูรณ์เพื่ออธิบายเอกสารเว็บ อย่างไรก็ตาม สมาชิกของ W3C ได้พิจารณาเนื้อหาเหล่านี้และลงความเห็นว่า SGML ค่อนข้างยุ่งยากต่อการจัดส่งข้อมูลไปเก็บไว้บนเว็บ ความยืดหยุ่นและลักษณะเฉพาะที่เกินความต้องการจะทำให้ยากต่อการเขียนโปรแกรมเพื่อการประมวลผลและแสดงผลข้อมูล SGML ในโปรแกรมเบราว์เซอร์ จึงมีการปรับปรุงชุดย่อยของ SGML ที่ถูกออกแบบเฉพาะสำหรับจัดส่งข้อมูลไปไว้บนเว็บ ในปี 1996 กลุ่มทำงาน XML ของ W3C ได้ทำการพัฒนาชุดย่อยนั้นโดยตั้งชื่อว่า Extensible Markup Language ซึ่ง XML ถูกออกแบบมาเพื่อเป็นวิธีการที่ง่าย ซึ่งเป็นลักษณะเฉพาะที่ชัดเจนต่อการลดข้อบกพร่องของ SGML

เนื่องจากอิเลเมนต์ของ XML ไม่มีความหมายจริงๆ จึงเป็นสาเหตุที่ทำให้ XML เป็นภาษาที่กำลังมาเป็นภาษาที่มีความสำคัญ เพราะ XML เป็นซับเซตของภาษา SGML (มากกว่าที่จะเป็นแอปพลิเคชันของ SGML) และสามารถจะถูกใช้ได้ทั้งสำหรับเก็บข้อมูลและเป็นพื้นฐานสำหรับแอปพลิเคชันที่ใช้ XML เช่น XML Stylesheet Language (XSL) หรือ WML ที่เป็นแอปพลิเคชันของ XML อย่างแท้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสาร XML (XML document) สามารถเก็บข้อมูลโดยใช้อิลิเมนต์ที่มีความหมายอย่างที่ต้องการให้มันเป็นอย่างแท้จริง แม้ว่าตอนนี้แนวความคิดนี้จะยังค่อนข้างใหม่แต่นี้ก็เป็นข้อได้เปรียบประการหลักหากผู้ส่งและผู้รับ รู้ความหมายของอิลิเมนต์นั้น เอกสาร XML สามารถถูกใช้ในเการเก็บและโอนย้ายระหว่างข้อมูลในแพลตฟอร์ม, ระบบปฏิบัติการและอุปกรณ์ที่เป็นอิสระจากกันอย่างแท้จริง และเนื่องจากมันเป็นแค่เท็กซ์เท่านั้น มันจึงสามารถส่งผ่านเครือข่ายชนิดใดๆ ก็ตามได้อย่างง่ายดาย รวมถึงการส่งผ่านโพรโตคอล HTTP ในการติดต่อสื่อสารของเครือข่ายอินเทอร์เน็ตด้วย ด้วยเหตุนี้ XML จึงเป็นเวอร์ชันของ SGML ที่ใช้งานง่ายและเหมาะสมสำหรับเว็บ เช่นเดียวกับ SGML ตรงที่ XML .ให้คุณออกแบบชุดอิลิเมนต์ของคุณเอง เมื่อต้องการอธิบายเอกสารที่เป็นเรื่องเฉพาะเหมือนกับ SGML รูปแบบที่เป็นเรื่องเฉพาะซึ่งเป็นคำจำกัดความ โปรแกรมประยุกต์ XML (หรือ เรียกอีกอย่างว่า Vocabulary)

การแปลงรูปแบบจาก XML ให้อยู่ในรูปแบบต่างๆ (Transforming XML into Other Formats)

XML Stylesheet Language (XSL) เป็นภาษาที่ใช้ในการแปลงจากรูปแบบ (transformation) ของเอกสาร XML ไปอยู่ในรูปแบบอื่น ๆ และในขณะเดียวกันก็ทำการประยุกต์ใช้รูปแบบการจัดข้อความ (style) ด้วย XSL stylesheets (สไตล์ชีต) ถูกเขียนในรูปแบบที่เหมาะสมกับ XML (XML-compliant format) แต่ต้องทำการกำหนดลำดับของอิลิเมนต์ไว้ล่วงหน้าในการกำหนดสิ่งที่ต้องการให้กระทำในการแปลงรูปแบบ (อีกประการหนึ่งคือ เพื่อให้ทราบไว้ว่าจะเลือกใช้อันไหนทำงาน) โดยความหมายจะถูกกำหนดโดยตัวประมวลผล XSL เอง ซึ่งควรจะเป็นไปตามมาตรฐานของ W3C ดังนั้นจาก XSL ซึ่งเป็นแอปพลิเคชันหนึ่งของ XML และอยู่ในรูปแบบที่เหมาะสมกับ XML ดังนั้นจึงสามารถประมวลผล XSL สไตล์ชีตได้เหมือนกับเอกสาร XML

ครั้งแรก XSL เป็นข้อเสนอที่ถูกระบุทั้งทางที่การแปลงรูปแบบ XML และไวยากรณ์ในการจัดรูปแบบบนแพลตฟอร์มที่เป็นอิสระกัน (platform-independent styling grammar) แต่อย่างไรก็ตาม ในส่วนของการแปลงรูปแบบก้าวหน้าอย่างรวดเร็วและเป็นประโยชน์อย่างมาก ขณะที่ไม่มีแอปพลิเคชันในส่วนของการจัดรูปแบบมากนักและส่วนใหญ่จะกล่าวว่ามีประโยชน์ในด้านการทำงานแค่น้อยนิดเท่านั้น ดังนั้นจากในข้อเสมอจากในตอนแรก จึงสามารถแบ่งออกได้เป็น 2 ส่วนที่แตกต่างกันคือ ส่วนในการแปลงรูปแบบเอกสาร ซึ่งปัจจุบันถูกเรียกว่า XSLT (XSL Transformation Language) และในส่วนของการจัดรูปแบบจะถูกเรียกอย่างเป็นทางการว่าวัตถุในการจัดรูปแบบของ XSL (XSL Formatting Object) หรือ XSL-FO แต่ในที่นี้จะพิจารณาเฉพาะ ในส่วนของการแปลงรูปแบบข้อมูลเท่านั้นเพื่อแปลงรูปแบบจาก XML ไปในรูปแบบและโครงสร้างใหม่ ดังนั้น XSLT จึงถือเป็นเครื่องมือหลักในการทำงานกับ XML

The XSL Transformation Language (XSLT)

XSLT เป็นข้อเสนอแนะจาก W3C ในการกำหนดเซตอิลิเมนต์ของ XML ที่สามารถจะถูกใช้ในการสร้างสไตล์ชีตที่แปลงเอกสาร XML ให้อยู่ในรูปแบบใดๆ ซึ่งในปัจจุบันนี้ใช้ในการแปลง XML ไปเป็น HTML ที่เหมาะสมเพื่อใช้ในการแสดงผลบนเว็บเบราว์เซอร์ อย่างไรก็ตามสามารถที่จะนำมาใช้ในการแปลง XML ไปเป็น WML หรือ รูปแบบใดๆ ตามแต่จะระบุใน XSL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สไลด์ซีดีโดยการทำงาน จำทำโดยนำเอาเอกสาร XML และสไลด์ซีดี XSL ที่เหมาะสมกับ XML นั้น ไปทำการแปลงรูปแบบเพื่อสร้างเอกสารเอาต์พุต ซึ่งขึ้นอยู่กับคำสั่งและเนื้อหาภายในสไลด์ซีดี จึงสามารถสร้างรูปแบบเอาต์พุตใดๆ ก็ได้ตามที่ต้องการในการทำงาน ต้องสร้างเอกสาร XML กับ XSL สไลด์ซีดีที่เหมาะสม แล้วจึงทำการแปลงเพื่อสร้างเอกสารเอาต์พุต ซึ่งขึ้นกับตัวข้อมูลและคำสั่งภายในสไลด์ซีดี โดยที่สามารถสร้างรูปแบบของเอาต์พุตในรูปแบบใดๆ ที่ต้องการ

2.2.3 WML (Wireless Markup Language)

WML โดยถูกออกแบบเพื่อให้เหมาะสมกับข้อจำกัดทางด้านฮาร์ดแวร์ และแบนด์วิธที่แคบของอุปกรณ์ไร้สาย ซึ่งจำกัดความสามารถในการอินพุตและเอาต์พุต โดยเป็นภาษาที่เขียนในรูปแบบแท็ก (tag-based) เช่นเดียวกับภาษา HTML รวมทั้งโครงสร้างการเขียนและแท็กบางอย่างจึงคล้ายกับ HTML ด้วย ข้อสำคัญของการเขียนแท็ก WML คือจะเขียนด้วยตัวพิมพ์เล็กทั้งหมดและต้องเขียนให้ถูกต้องตามโครงสร้างที่กำหนดเอาไว้ จะเขียนแท็กขาดหรือเกินเหมือน HTML ไม่ได้

โครงสร้างของเอกสาร WML ถูกออกแบบมาให้มีการทำงานเป็นการ์ดซึ่งเปรียบได้กับเว็บเพจหน้าหนึ่งหน้าและเมื่อรวมการ์ดหรือเว็บเพจหลายๆ หน้าเข้าด้วยกันจะเรียกว่า เด็ค (deck) ซึ่งก็คือเว็บไซต์นั่นเอง ซึ่งการที่ WML ถูกออกแบบให้มีลักษณะหลายๆ การ์ดในหนึ่งเด็ค เพราะอุปกรณ์ WAP จะได้ไม่เสียเวลาในการส่งคำร้องขอไปยังเซิร์ฟเวอร์หลายครั้ง ซึ่งเป็นการช่วยลดเวลา และแบนด์วิธลง

ดังนั้น โครงสร้างที่ใหญ่ที่สุดของเอกสาร WML คือ เด็ค ซึ่งก็คือเนื้อหาทั้งหมดที่อยู่ในเอกสาร WML . ในแต่ละเด็คจะประกอบไปด้วยกลุ่มเอกสารย่อย ๆ ซึ่งเรียกว่า การ์ด (card) ในแต่ละการ์ดจะเป็นที่เก็บข้อมูลต่างๆ ที่ต้องการให้แสดงที่หน้าจอรวมไปถึงคำสั่งต่างๆ ที่ต้องการให้เอกสาร WML ตอบสนองต่อความต้องการของผู้ใช้ อุปกรณ์เว็บจะทำการแสดงได้ที่การ์ดเท่านั้น โดยการ์ดที่อยู่บนสุดจะถูกแสดงก่อน ส่วนการ์ดต่อๆ ไปจะถูกแสดงก็ต่อเมื่อถูกเรียกดูเท่านั้น ขึ้นอยู่กับว่าเลือกเมนูหรือลิงก์อย่างไรในการกดแต่ละใบจะมีข้อมูลอยู่ในตัวมันเอง ซึ่งข้อมูลเหล่านั้นสามารถเป็นได้ทั้งข้อมูลสำหรับอ่าน หรือเป็นจอให้ผู้ให้บริการกรอกข้อมูล หรือเป็นเมนูและลิงก์ให้เลือก พูดย่าง ๆ ก็คือ ทำหน้าที่เป็นส่วนติดต่อกับผู้ใช้ (user interface) บนจอให้อุปกรณ์ WAP นั้นเอง ผู้ใช้บริการในระบบเว็บ จะทำการส่งข้อมูลที่ละหนึ่งเด็คเท่านั้น ดังนั้นจึงจำเป็นอย่างยิ่งที่จะต้องออกแบบเด็คและการ์ดให้ดี ดังนั้นจึงไม่ควรเขียน WML . ให้แต่ละการ์ดมีขนาดใหญ่เกินไป คือให้แต่ละการ์ดสามารถถูกแสดงได้บนหน้าจอได้เพียงหน้าจอเดียวเท่านั้น ถ้าไม่สามารถทำได้ก็ควรจะให้เขียนเป็นการ์ดใหม่ ทั้งนี้เพื่อความง่ายต่อการใช้งาน และถ้าเด็คใหญ่เกินไปจะทำให้ใช้เวลานานในการแสดงผลบนจอของอุปกรณ์เว็บ แต่ถ้าเล็กเกินไปหรือจะต้องเรียกเด็คอื่นๆ ก็จะทำให้ต้องมีการส่งรับข้อมูลบ่อยขึ้น ทำให้การทำงานเป็นไปได้อย่างไม่ต่อเนื่องและรวดเร็วเท่าที่ควร

2.3 เซิร์ฟเลต

แม้ว่าโลกของอินเทอร์เน็ตจะเพิ่งเกิดขึ้นเพียงไม่กี่ปีก็ตาม เทคโนโลยีที่ใช้กับอินเทอร์เน็ตกลับมีการเปลี่ยนแปลงไปอย่างรวดเร็วมาก ในสมัยแรกๆ เพจต่างๆ ที่อยู่เว็บจะเป็นลักษณะของเพจสแตติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเพจที่ไม่มีการเปลี่ยนแปลงเนื้อหาที่ไม่ว่าจะนานเท่าไรหรือนอกเสียจากว่าผู้ดูแลเพจนั้นจะทำการอัปเดต เพจดังกล่าว เพจลักษณะนี้เป็นเพจที่นิยมใช้กันทั่วไปในอินเทอร์เน็ตสมัยแรกเพราะอินเทอร์เน็ตยังนิยมกัน อยู่ในวงแคบ โดยกลุ่มผู้ใช้จะเป็นกลุ่มบุคคลที่อยู่ในวงการศึกษานั้น ต่อมาจากนั้นไม่นานทางผู้ผลิต บราวเซอร์ได้ทำการเพิ่มความสามารถให้กับเพจโดยอนุญาตให้เพจสามารถแทรกสคริปต์เล็ก ๆ ลงไปรวม กับส่วนที่เป็น HTML ได้ซึ่งจุดนี้ก็คือจุดเริ่มต้นของจาวาสคริปต์ทางฝั่งไคลเอนต์นั่นเอง

แม้ว่าเพจจะเริ่มมีความสามารถในการโต้ตอบกับผู้ใช้โดยอิงความสามารถจากจาวาสคริปต์แล้วก็ตาม ถ้าพูดถึงในแง่ของส่วนเนื้อหาของตัวเพจจริง ๆ แล้วตัวเพจเองก็ยังคงเป็นเพจสแตติกอยู่เช่นเดิม เมื่อกลุ่มผู้ใช้อินเทอร์เน็ตเริ่มมีมากขึ้นความต้องการที่จะให้เพจสามารถทำการรับส่งข้อมูลรวมไปถึงเปลี่ยนแปลงเนื้อหาได้โดยอัตโนมัติก็เกิดขึ้น เทคโนโลยีที่เกิดขึ้นเพื่อรองรับความต้องการเหล่านี้ก็คือแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์ (Server Side Application) นั่นเอง

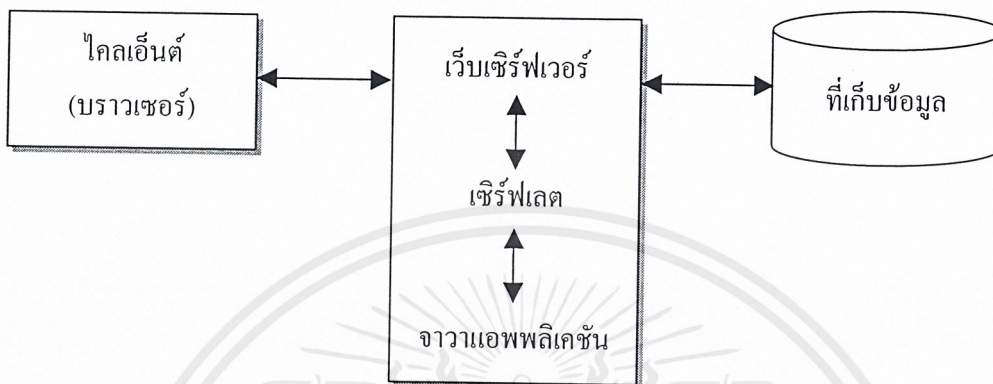
แอปพลิเคชันทางฝั่งเซิร์ฟเวอร์ ในระยะแรก ๆ มักถูกเขียนขึ้นด้วยคอนเซ็ปต์ของ CGI (Common Gateway Interface) โดยหลักการทำงานง่าย ๆ ก็คือเว็บเบราว์เซอร์จะทำการส่งข้อมูลที่เกิดจากเอ็กชันของผู้ใช้ เช่น การคลิกลิงก์หรือการกรอกแบบสอบถามไปยังเว็บเซิร์ฟเวอร์โดยแทนที่เว็บเซิร์ฟเวอร์จะทำการส่งเพจที่เป็นเพจสแตติกกลับมาเว็บเซิร์ฟเวอร์จะทำการส่งต่อข้อมูลดังกล่าวไปยังโปรแกรมซึ่งถูกจัดไว้ โปรแกรมดังกล่าวจะทำการประมวลผลข้อมูลที่ได้แล้วจะส่งผลกลับไปยังเว็บเซิร์ฟเวอร์ซึ่งทางเว็บเซิร์ฟเวอร์ก็จะส่งผลที่ได้นี้กลับไปยังเว็บเบราว์เซอร์อีกทีหนึ่ง หลาย ๆ คนคงเคยเห็นแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์ที่เป็นผลผลิตของ CGI ในรุ่นแรก ๆ ที่ยังหลงเหลืออยู่ในปัจจุบันนี้ ยกตัวอย่างเช่น ตัวนับ (Counter), แผนที่รูปภาพ (Image Map), สมุดเยี่ยม (Guessbook), ส่งเมล (SendMail) เป็นต้น จริง ๆ แล้วตัว CGI เองสามารถเขียนด้วยภาษาอะไรก็ได้ แต่ที่นิยมมากที่สุดเห็นจะเป็น C และ Perl อาจจะเป็นเพราะว่า CGI เป็นสิ่งที่มีมากับอินเทอร์เน็ตตั้งแต่ช่วงแรก ๆ ดังนั้นไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์ไหนก็ตามแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์พื้นฐานที่ทางเว็บเซิร์ฟเวอร์เหล่านั้นจะต้องสนับสนุนก็คือ CGI ซึ่งจุดนี้เองที่เป็นจุดเด่นทำให้ CGI เป็นที่นิยมใช้กันอย่างกว้างขวางจนกระทั่งปัจจุบันนี้

2.3.1 ความหมายของเซิร์ฟเลต

เซิร์ฟเลตเป็นแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์แบบหนึ่งซึ่งอ้างอิงคอนเซ็ปต์มาจาก CGI ข้อดีของเซิร์ฟเลตที่อยู่เหนือ CGI อย่างแรกก็คือตัวภาษาที่ใช้เขียนซึ่งก็คือจาวานั่นเอง จาวาเป็นภาษาที่ใช้คอนเซ็ปต์ของ Object Oriented ในการเขียน หลายคนที่เกี่ยวข้องกับการเขียนโปรแกรมสำหรับโปรเจกต์ใหญ่ ๆ จะทราบดีว่า Object Oriented สามารถลดความซับซ้อนของโครงสร้างโปรแกรมรวมถึงการอำนวยความสะดวกในการส่วนของโปรแกรมที่เขียนไว้แล้วมาใช้ใหม่เพียงไร นอกจากนี้จาวายังเป็นภาษาที่เป็นลักษณะแบบอิสระจากรูปแบบซึ่งจะช่วยให้สามารถที่จะทำการพัฒนาระบบโดยใช้ เอ็นไวรอนเมนต์อะไรก็ได้ซึ่งโดยทั่วไปมักนิยมใช้ เอ็นไวรอนเมนต์ของวินโดวส์โดยจะนำโปรแกรมที่เขียนเสร็จแล้วมารันบน เอ็นไวรอนเมนต์ของยูนิกซ์เพื่อเพิ่มประสิทธิภาพของโปรแกรมแทน

นอกจากนี้ เซิร์ฟเลตยังมีความเร็วที่สูงกว่า CGI เพราะ เซิร์ฟเลตใช้หลักการของ thread โดยจะทำการสร้าง 1 thread ต่อหนึ่งร้องขอที่มาจากไคลเอนต์ ซึ่งในทางกลับกัน CGI จะทำการสร้าง 1 โพรเซส ต่อ

หนึ่งการร้องขอ ซึ่งจะทำให้เปลืองทรัพยากรมากกว่าและประมวลผลในการรันก็จะช้ากว่าด้วย ท้ายที่สุดจุดเด่นที่สำคัญของเซิร์ฟเลตก็คือ API (Application Programming Interface) โดยระบบที่ทำการพัฒนาโดยใช้คอนเซ็ปต์ของเซิร์ฟเลตจะสามารถเรียกใช้ API ที่ทางจาวามีมาให้ (javax.servlet., javax.servlet.http.*) ซึ่งจะช่วยให้การพัฒนาาระบบดังกล่าวง่ายและเร็วยิ่งขึ้น



รูปที่ 2.3 หน้าี่และการทำงานของเซิร์ฟเลต

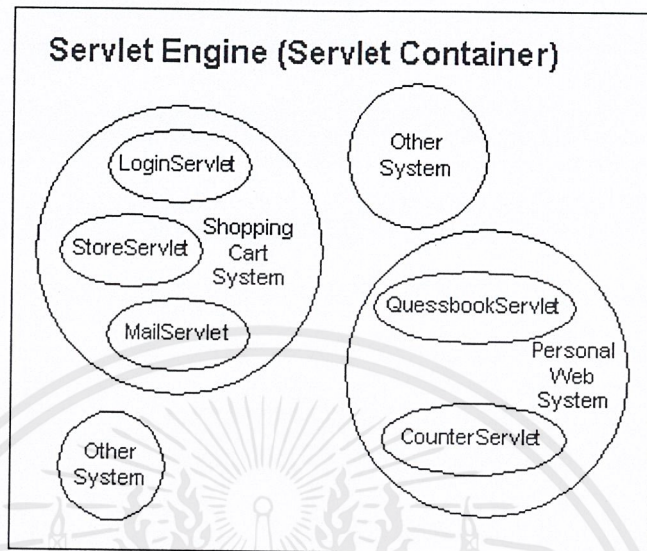
เมื่อการขอเรียกใช้บริการเซิร์ฟเลตมาถึงยังเซิร์ฟเวอร์ เซิร์ฟเลตเอ็นจินจะเรียกตัวโหลดคลาส (class loader) ตัวโหลดคลาสจะทำการตรวจสอบว่าคลาสดังกล่าวถูกโหลดขึ้นมาเรียบร้อยแล้วหรือไม่และตรวจสอบเวอร์ชันของคลาสที่ถูกโหลดว่ามีค่าไทม์แสตมป์ (time stamp) เท่ากับ .class ไฟล์ (.class file) ที่อยู่บนดิสก์ (disk) หรือไม่ ถ้าคลาสของเซิร์ฟเลตยังไม่ถูกโหลดหรือมีเวอร์ชันเก่ากว่าไฟล์ที่อยู่บนดิสก์ เซิร์ฟเลตเอ็นจินจะทำการโหลดคลาสไปเก็บไว้ใน JVM และสร้างอินสแตนซ์ (instance) ของคลาสดังกล่าว หลังจากนั้นวงจรชีวิต (life cycle) ของเซิร์ฟเลตจะเริ่มขึ้น

2.3.2 เซิร์ฟเลตเอ็นจิน (Servlet engine)

ในการรันระบบที่เขียนขึ้นโดยใช้หลักการของ เซิร์ฟเลตจะต้องนำระบบดังกล่าวมาบรรจุอยู่ในสิ่ง ๆ หนึ่งที่เรียกว่า เซิร์ฟเลตเอ็นจิน ให้นึกถึง เซิร์ฟเลตเอ็นจิน คล้าย ๆ กับกล่อง ๆ หนึ่งที่ใส่ลูกปิงปองไว้หลายลูก โดยลูกปิงปองแต่ละลูกก็คือระบบ ๆ หนึ่งนั่นเอง หลายคนอาจสงสัยทำไมถึงใช้คำว่าระบบ โดยทั่วไปแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์หนึ่ง ๆ ที่ถูกเขียนขึ้นโดยใช้ เซิร์ฟเลต API จะถูกเรียกว่า เซิร์ฟเลตในหนึ่งระบบอาจประกอบด้วย เซิร์ฟเลตหลายอัน ยกตัวอย่างเช่น ระบบที่เกี่ยวข้องกับ Shopping Cart อาจประกอบด้วย เซิร์ฟเลตที่ทำหน้าที่ในการเช็คล็อกอิน เซิร์ฟเลตที่ทำหน้าที่ในการเก็บข้อมูลสินค้า, เซิร์ฟเลตที่ทำหน้าที่ในการส่งเมลกลับไปยังลูกค้าเพื่อบอกว่าได้ทำการส่งของไปให้แล้ว เป็นต้น ดังนั้นถ้ามองโดยรวมแล้ว เซิร์ฟเลตเอ็นจิน ก็คือที่รวมของระบบตั้งแต่หนึ่งระบบถึงหลายระบบ โดยแต่ละระบบจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วย เซิร์ฟเลตหนึ่งอันหรือมากกว่า ระบบ ในที่นี้อาจจะหมายถึง Zone (Apache JServ) หรือแอปพลิเคชันของเว็บ (Tomcat) ก็ได้



รูปที่ 2.4 เซิร์ฟเลตเอนจิน และเซิร์ฟเลตที่อยู่ภายใน

เซิร์ฟเลตเอนจินเป็นเพียงกล่อง ๆ หนึ่งที่ใช้บรรจุและรันกลุ่มของเซิร์ฟเลตเท่านั้น ในการที่จะทำการติดต่อสื่อสารกับไคลเอนต์ตัว เซิร์ฟเลตเอนจินนี้จะต้องทำงานร่วมกับเว็บเซิร์ฟเวอร์ซึ่งเปรียบเสมือนฉากหน้าที่ติดต่อกับไคลเอนต์อีกทีหนึ่ง เมื่อใดก็ตามที่มีร้องขอ ส่งมาจากไคลเอนต์ถ้าร้องขอ นั้นจะจงมาที่ตัวเซิร์ฟเลตทางเว็บเซิร์ฟเวอร์ก็จะทำการ forward ตัวร้องขอนั้นมาให้ เซิร์ฟเลตเอนจินซึ่งทาง เซิร์ฟเลตเอนจินก็จะทำการเรียกเซิร์ฟเลตที่ไคลเอนต์ต้องการขึ้นมาทำการประมวลผลร้องขอ นั้น โดยท้ายสุดเซิร์ฟเลตจะส่งผลกลับไปให้เซิร์ฟเลตเอนจิน, เซิร์ฟเลตเอนจินก็จะส่งต่อผลที่ได้กลับไปให้เว็บเซิร์ฟเวอร์ซึ่งเว็บเซิร์ฟเวอร์ก็จะส่งผลกลับไปให้ไคลเอนต์

เซิร์ฟเลตเอนจินอาจจะเป็นส่วนที่ติดมากับเว็บเซิร์ฟเวอร์อยู่แล้วยกตัวอย่างเช่น เซิร์ฟเลตเอนจินที่อยู่ใน Netscape Enterprise Server, IBM WebSphere หรืออาจจะเป็นส่วนที่เป็น Add-on ให้กับเว็บเซิร์ฟเวอร์ก็ได้เช่น Apache Jserv, Tomcat, JRun หรือแม้กระทั่งเป็นส่วนหนึ่งที่อยู่ในเว็บแอปพลิเคชัน เช่น BEA Weblogic เป็นต้น ทั้งนี้การเลือกใช้ เซิร์ฟเลตเอนจินแต่ละชนิดก็มักขึ้นอยู่กับปัจจัยหลายอย่าง เช่น ความสะดวกในการรวมระบบที่จะสร้างขึ้นใหม่กับระบบที่มีอยู่แล้ว, งบประมาณที่มีอยู่สำหรับโครงการหรืออาจจะรวมไปถึงทักษะและประสบการณ์ส่วนตัวของนักพัฒนาแต่ละคน

2.3.3 อินเทอร์เฟซ javax.servlet.Servlet

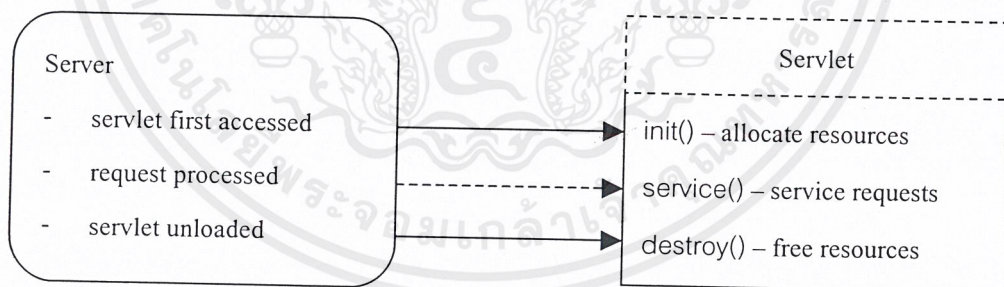
หัวใจของเซิร์ฟเลตจริง ๆ อยู่ที่อินเทอร์เฟซที่ชื่อ javax.servlet.Servlet* โดยทุกเซิร์ฟเลตที่ถูกเขียนขึ้นจะต้องทำการอิมพลีเมนต์ตัวอินเทอร์เฟซนี้ไม่ทางตรงก็ทางอ้อม (ทางตรงก็คือการอิมพลี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนต์ตัวอินเทอร์เฟซนี้เลย ส่วนทางอ้อมก็คือการให้เซิร์ฟเลตทำการย่อยคลาสบางคลาสที่ได้ทำการอิมพลิเมนต์ตัวอินเทอร์เฟซนี้ไว้แล้ว) เหตุผลว่าทำไมจึงต้องอิมพลิเมนต์ตัวอินเทอร์เฟซนี้เพราะว่าเมื่อไรก็ตามที่มีร้องขอ จากไคลเอนต์เข้ามายังเซิร์ฟเลตเอนจินตัว เซิร์ฟเลตเอนจินจะทำการหาเซิร์ฟเลตที่ร้องขอ ดังกล่าวอ้างถึง หลังจากนั้น เซิร์ฟเลตเอนจินจะทำการเรียกฟังก์ชันต่าง ๆ ที่อยู่ในเซิร์ฟเลตเพื่อทำการประมวลผลร้องขอ ของไคลเอนต์โดยฟังก์ชันที่ เซิร์ฟเลตเอนจินจะทำการเรียกก็คือฟังก์ชันที่เซิร์ฟเลตได้ทำการอิมพลิเมนต์ซึ่งเป็นฟังก์ชันที่ถูกกำหนดอยู่ใน `javax.servlet.Servlet` อินเทอร์เฟซนั่นเอง อาจมีคำถามว่าทำไม เซิร์ฟเลตเอนจินถึงเรียกฟังก์ชันที่ถูกกำหนดอยู่ในอินเทอร์เฟซนี้ เหตุผลง่าย ๆ ก็คือเซิร์ฟเลตเป็นส่วนที่ถูกโหลดเข้าไปใน เซิร์ฟเลตเอนจินในช่วงรันไทม์ ตัวเซิร์ฟเลตเอนจินเองไม่สามารถทราบได้ว่าเซิร์ฟเลตต่าง ๆ มีฟังก์ชันอะไรประกอบอยู่บ้างนอกเสียจากว่าเซิร์ฟเลตนั้นได้ทำการอิมพลิเมนต์ฟังก์ชันที่เป็นมาตรฐานที่ เซิร์ฟเลตเอนจินรับรู้ซึ่งก็คือเหตุผลว่าทำไมทุกเซิร์ฟเลตจะต้องทำการอิมพลิเมนต์ตัว `javax.servlet.Servlet` อินเทอร์เฟซอย่างไรก็ตามสามารถให้ เซิร์ฟเลตเอนจินเรียกฟังก์ชันอื่น ๆ ที่อยู่ในเซิร์ฟเลตได้ซึ่งวิธีการก็คือการใส่ฟังก์ชันดังกล่าวเข้าไปในส่วนอิมพลิเมนต์ของฟังก์ชันต่าง ๆ ที่ถูกกำหนดอยู่ใน `javax.servletServlet` อินเทอร์เฟซนั่นเอง

2.3.4 วงจรชีวิตของเซิร์ฟเลต

วงจรชีวิตของเซิร์ฟเลตประกอบด้วย 3 สถานะคือ สถานะเริ่มต้น (`init()`) สถานะให้บริการ (`service()`) และสถานะทำลาย (`destroy()`) เมธอดเหล่านี้เป็น “callback method” หมายถึงเมธอดจะไม่ถูกเรียกโดยตัวเซิร์ฟเลตเอง แต่จะถูกเรียกโดยเซิร์ฟเลตเอนจิน (servlet engine) ในการตอบสนองกับอีเวนต์ (event) ที่เกิดขึ้น



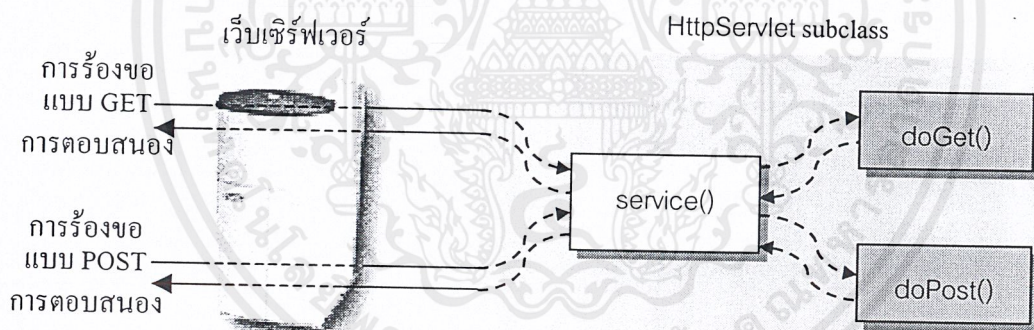
รูปที่ 2.5 วงจรชีวิตของเซิร์ฟเลต

1. สถานะเริ่มต้น เป็นการใช้เมธอด `init()` เมธอดจะถูกเรียกโดยเซิร์ฟเวอร์ทันทีที่อินสแตนซ์ถูกสร้างเสร็จเรียบร้อยแล้ว โดยปกติเมธอดนี้จะถูกใช้เพื่อทำการเริ่มต้นเซิร์ฟเลต สาเหตุที่ไม่ใช้คอนสตรัคเตอร์ (constructor) เพราะคอนสตรัคเตอร์ของ JDK 1.0 (เริ่มมีเซิร์ฟเลต) ไม่สามารถรับอาร์กิวเมนต์ (argument) ในกรณีที่คลาสถูกโหลดแบบไดนามิก (dynamic) ได้ ดังนั้นการที่จะจัดหาข้อมูลเกี่ยวกับตัวมันเองและสิ่งแวดล้อมให้แก่เซิร์ฟเลตที่ถูกโหลดเป็นครั้งแรก เซิร์ฟเวอร์จำเป็นต้องเรียกเมธอด `init()` และส่งผ่านออบเจกต์ (object) ที่ใช้อินเทอร์เฟซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ServletConfig นอกจากนี้ภาษาจาวาไม่อนุญาตให้อินเทอร์เฟซมีการประกาศคอนสตรัคเตอร์ นั่นหมายถึงอินเทอร์เฟซของแพ็คเกจ (package) javax.servlet.Servlet ก็ไม่สามารถประกาศคอนสตรัคเตอร์ที่รับค่าพารามิเตอร์ (parameter) ServletConfig ได้เช่นกัน จึงต้องประกาศเมธอดชื่ออื่นแทนเช่น init() แต่ยังคงมีทางเป็นไปได้ที่จะประกาศคอนสตรัคเตอร์ให้แก่เซิร์ฟเล็ต แต่ในคอนสตรัคเตอร์ดังกล่าวห้ามมีการเข้าถึงออบเจ็กต์ ServletConfig และไม่สามารถโยน (throw) ServletException ได้ ออบเจ็กต์ ServletConfig จะส่งพารามิเตอร์ที่จำเป็นต้องใช้ในการเริ่มต้นการทำงานให้แก่เซิร์ฟเล็ต พารามิเตอร์ดังกล่าวจะถูกป้อนให้แก่เซิร์ฟเล็ตโดยไม่เกี่ยวข้องกับการร้องขอที่มีเข้ามา

- สถานะให้บริการจะถูกเรียกจากเซิร์ฟเวอร์ทุกครั้งที่ไคลเอ็นต์ทำการร้องขอใช้บริการเซิร์ฟเล็ต เมธอด service() จะรับค่าพารามิเตอร์ 2 ค่า ได้แก่ออบเจ็กต์การร้องขอและออบเจ็กต์การตอบสนอง ออบเจ็กต์การร้องขอบอกเซิร์ฟเล็ตเกี่ยวกับการขอใช้บริการที่มีเข้ามา ในขณะที่ออบเจ็กต์การตอบสนองจะถูกใช้ในการส่งการตอบรับคืนกลับให้แก่เซิร์ฟเวอร์ โดยปกติแล้วเมธอด service() จะไม่ถูกโอเวอร์ไรด์ (override) แต่จะทำการโอเวอร์ไรด์เมธอด doGet() เพื่อใช้จัดการการร้องขอแบบ get และเมธอด doPost สำหรับจัดการกับการร้องขอแบบ post แทน ดังตัวอย่างในรูป



รูปที่ 2.6 วิธีการจัดการกับการร้องขอแบบ GET และ POST ของเซิร์ฟเล็ต

- สถานะทำลาย (destroy()) เซิร์ฟเวอร์จะเรียกเมธอด destroy() ของเซิร์ฟเล็ต เมื่อเซิร์ฟเล็ตกำลังจะถูกอันโหลด (unload) ในเมธอด destroy() ควรมีการคืนทรัพยากรต่าง ๆ ที่ได้มาให้แก่ระบบ เพื่อที่โปรแกรมอื่นๆ จะได้นำทรัพยากรเหล่านั้นไปใช้ได้ นอกจากนี้เมธอดยังให้โอกาสเซิร์ฟเล็ตในการบันทึกสถานะและข้อมูลต่างๆ ที่ถูกเรียกใช้เป็นประจำจากการเรียกเมธอด init() ในครั้งต่อไป

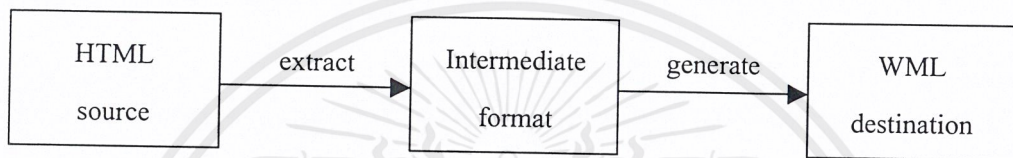
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการในการออกแบบและสร้างตัวแปลงภาษา HTML เป็น WML

3.1 บทนำ

ตัวแปลงทำงานโดยการตัดแบ่งข้อความจากหน้า HTML ดันฉบับและทำการจัดรูปแบบใหม่ให้เป็นภาษา WML ซึ่งใช้หลักการออกแบบเอ็นทีเยร์ (n-tier) ดังรูป ตัวแปลงจะทำการแปลงรูปแบบข้อมูลให้เหลือแต่เนื้อหาข้อมูลจริงๆ (pure data) หลังจากนั้นจึงทำการจัดรูปแบบให้เป็น WML



รูปที่ 3.1 ขั้นตอนการแปลง HTML เป็น WML

ภาษากลางที่แปลไปนั้น (intermediate data) สามารถถูกนำไปจัดการต่อได้โดยไม่ขึ้นกับกระบวนการที่แบ็กเอนด์และฟรอนต์เอนด์

3.2 วิธีการในการแปลงเอกสาร HTML

วิธีที่ใช้ในการแปลงเอกสาร HTML สามารถแบ่งเป็น 3 ประเภทใหญ่ๆ ได้ดังนี้

3.2.1 การแปลงอัตโนมัติแบบสมบูรณ์

เป็นตัวแปลงไม่ยินยอมให้ผู้ใช้เซตคอนฟิกูเรชันได้ ซึ่งจะมีประโยชน์อย่างยิ่งในการแปลงเอกสาร HTML ใดๆ เอกสาร HTML ทุกหน้าจะถูกแปลงด้วยกฎในการแปลง (conversion rules) อันเดียวกัน ปัญหาที่ตามมาในการแปลงด้วยวิธีนี้คือไม่สามารถควบคุมคอนเท้นต์ของเอกสาร HTML ดันฉบับได้นั้นคือลิงก์ อินพุตฟอร์ม รูปภาพ เฟรม ฯลฯ ที่มีอยู่ภายในเอกสาร HTML จะต้องถูกแปลงทั้งหมดและปรากฏให้ผู้ใช้เห็นอยู่บนอุปกรณ์เว็บ รวมไปถึงขนาดของคอนเท้นต์ HTML บางหน้ามีขนาดใหญ่จนเกินความสามารถที่อุปกรณ์เว็บจะสามารถรองรับได้ ดังนั้นขั้นตอนและรูปแบบในการแปลงด้วยวิธีนี้จะมี ความซับซ้อนมากกว่าแบบที่ผู้ใช้สามารถเซตคอนฟิกูเรชันได้

3.2.2 ตัวแปลงที่ผู้ใช้สามารถเซตคอนฟิกูเรชันได้

เป็นการแปลงที่ผู้ใช้สามารถเลือกได้ว่าต้องการแปลงส่วนใดของเอกสาร HTML ให้มาปรากฏบนโทรศัพท์เว็บเช่น ผู้ใช้อาจต้องการเพียงลิงก์ต่างๆ เพื่อค้นหาข้อมูล อาจต้องการอินพุตฟอร์มเพื่อใช้ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การล็อกอินเข้าสู่ระบบ หรือต้องการดูเพียงแค่ว่าตารางข้อมูลที่มีอยู่ในหน้า HTML ดันฉบับเท่านั้น ดังนั้น การแปลงด้วยวิธีนี้ จะไม่ยุ่งยากซับซ้อนเท่ากับวิธีแรกและไม่สามารถรองรับข้อมูลได้ทุกประเภทในแต่ละครั้งที่ทำการแปลง แต่มีข้อได้เปรียบคือได้ข้อมูลที่ตรงตามความต้องการของผู้ใช้

3.2.3 การแปลงโดยการแก้ไขเอกสารต้นฉบับ

เป็นการแปลงที่ต้องอาศัยการแก้ไขเอกสาร HTML ดันฉบับ ด้วยวิธีนี้จะมีการเพิ่มแท็กที่ใช้เป็น สัญลักษณ์ในการระบุส่วนของเอกสาร HTML ที่มีไว้สำหรับการแปลงให้เป็น WML ดังตัวอย่างในภาพ

```

.
.
.
<!-- WAP_START -->
    this is the part of the HTML-document you want to display
<!-- WAP_END -->
.
.
.

```

รูปที่ 3.2 การแก้ไขเอกสาร HTML ดันฉบับ

ซึ่งวิธีดังกล่าวจะทำให้กระบวนการในการตัดแบ่งและแปลงข้อมูลมีความยุ่งยากน้อยลง สามารถสร้างตัวแปลงได้ง่ายขึ้น

ในช่วงเริ่มแรกดูเหมือนว่าวิธีนี้จะเป็นวิธีที่น่าสนใจและเป็นหนทางในการแก้ปัญหาที่ไม่สลับซับซ้อน แต่หากต้องทำการแก้ไขเอกสาร HTML ดันฉบับทุก ๆ หน้าก็เหมือนกับการที่ต้องทำการสร้างเอกสาร WML สำหรับ HTML แต่ละหน้าด้วยตนเอง วิธีนี้จึงไม่เป็นที่นิยมมากนัก

3.3 เครื่องมือช่วยในการแปลง

หลังจากศึกษาหลักการและวิธีการต่าง ๆ ในการแปลงแล้ว ได้ทำการสร้างตัวแปลงซึ่งเป็นแบบการแปลงอัตโนมัติแบบสมบูรณ์ขึ้น เนื่องจากเห็นว่าจะสามารถนำไปประยุกต์ใช้ได้อย่างกว้างขวางมากขึ้นในอนาคตเพราะมีความยืดหยุ่นสูงสามารถแปลงเอกสาร HTML ที่มีอยู่ทั่วไปได้ไม่เฉพาะเจาะจง ตัวแปลงที่สร้างขึ้นนี้สามารถแบ่งออกได้เป็น 3 ส่วน (3-tier) ดังรูปที่ 2.1 ในส่วนที่ทำการตัดแบ่งเอกสาร HTML จะมีการเฟตช์ (fetch) คอนเท้นต์ของ HTML ดันฉบับจากโฮสต์ของเอกสารทุกครั้งที่ทำการแปลง นั่นคือผลลัพธ์ WML ที่ได้จะตรงกับหน้า HTML ปัจจุบัน ข้อมูลที่ถูกตัดแบ่งมาจะอยู่ในรูปโครงสร้างต้นไม้ของ XML อิลิเมนต์ ซึ่งจะถูกรูปแปลงต่อไปโดย XSL สไตลชีตเพื่อสร้างผลลัพธ์ WML ต่อไป วิธีการแปลงถูกแบ่งเป็น 3 ส่วนได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัดแปลงและตัดแบ่งข้อมูลจาก HTML ให้เป็น XML
- จัดสรร XML
- สร้าง WML

การสร้าง WML จาก XML ในขั้นที่ 3 มีการใช้เฟรมเวิร์กที่ชื่อ โคคอน (cocoon) เข้ามาช่วย WML เป็นภาษามาร์กอัป (markup language) ที่สอดคล้องกับ XML ดังนั้น XML จึงถูกเลือกให้เป็นภาษากลางในการแปลง เมื่อได้เอกสาร XML (ซึ่งถูกสร้างมาจากเอกสาร HTML ต้นฉบับ) มาแล้ว เอกสาร XML จะถูกส่งต่อไปยังฟัลซิงเฟรมเวิร์ก (publishing framework) ซึ่งจะส่งผลลัพธ์ซึ่งเป็นเอกสาร WML ออกมา โดยขั้นตอนนี้จะมี XSL สไตลชีตเป็นส่วนร่วมในการแปลง XSL จะแปลงข้อมูลดิบ (ในที่นี้คือ XML) ให้เป็นคอนเทนต์ที่ถูกจัดรูปแบบไว้เรียบร้อยแล้ว ซึ่งคอนเทนต์ดังกล่าวจะถูกทำขึ้นเฉพาะสำหรับแต่ละอุปกรณ์เอาต์พุตหรือสำหรับแต่ละภาษามาร์กอัป ในที่นี้คอนเทนต์ถูกทำขึ้นโดยเฉพาะสำหรับการแปลงเป็นภาษา WML บนอุปกรณ์เว็บ

3.3.1 เฟรมเวิร์ก อาปาเช่ โคคอน (Apache Cocoon)

ฟัลซิงเฟรมเวิร์กที่นำมาใช้ในการสร้างตัวแปลงได้แก่ อาปาเช่ โคคอน เป็นเฟรมเวิร์กที่ถูกสร้างจากภาษาจาวาเพียงภาษาเดียวและอาศัยเทคโนโลยีใหม่ที่สร้างขึ้นโดย W3C เพื่อใช้ในการจัดหาคอนเทนต์ของเว็บ (เช่น DOM XML และ XSL) โคคอนรองรับการสร้างเอกสาร HTML และ XML โดยเฉพาะ XML ซึ่งจะถูกแปลงให้เป็นภาษามาร์กอัปอื่น ๆ ที่มีรูปแบบสอดคล้องกับ XML ต่อไป

โคคอนถูกสร้างขึ้นด้วยภาษาจาวาและมีพื้นฐานอยู่บนเซิร์ฟเลต (servlet) ซึ่งทำให้มีประโยชน์อย่างยิ่งเพราะจะไม่ขึ้นกับแพลตฟอร์มและทูล สามารถนำไปใช้กับระบบปฏิบัติการ เว็บเซิร์ฟเวอร์ หรือ JVMs ตัวใดก็ได้ โคคอนสามารถทำงานได้ทั้งแบบสแตติก (นำเอาเอกสาร XML ที่มีอยู่บนระบบไฟล์หรือจากที่เก็บข้อมูลที่ตายตัวมาใช้) และไดนามิก (จะสร้างเอกสาร XML ขึ้นใหม่ทุกครั้งที่ทำงาน) ตัวแปลงที่สร้างขึ้นนี้ใช้ประโยชน์จากส่วนที่เป็นไดนามิกเพื่อให้ได้ผลลัพธ์ WML ที่ตรงกับหน้าของ HTML ปัจจุบันมากที่สุด นั่นคือจะมีการตัดแบ่งข้อมูลทุกครั้งที่ใช้ขอเรียกใช้บริการ นอกจากนี้โคคอนยังมีระบบแคชซิง (caching system) ที่มีความสามารถเพื่อเพิ่มประสิทธิภาพในการให้บริการ เนื่องจากมีความเป็นไปได้ที่การประมวลผล XML ต้องการทรัพยากรมากเพราะอาจมีออบเจกต์จำนวนมากที่เป็นที่ต้องการในการแสดงผลเอกสาร ซึ่งต้องการเวลาในการประมวลผล รวมไปถึงการอลโลเคทหน่วยความจำ (memory allocation) โอเวอร์เฮดที่เกิดจากการบริหาร (management overhead)

3.4 หลักการทำงานของตัวแปลงภาษา

3.4.1 การตัดแบ่งและตัดแปลงเอกสาร

อ่านไฟล์ HTML ต้นฉบับครั้งละ 1 บรรทัด แล้วนำแต่ละบรรทัดมาเรียงต่อกันเพื่อให้กลายเป็นค่าคงที่สตริง (String) 1 ค่าเท่านั้นหรือกล่าวง่าย ๆ คือเก็บไฟล์ HTML ต้นฉบับทั้งไฟล์ไว้ภายในสตริง 1 ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้ง่ายต่อการตัดแบ่งโดยใช้คลาส String ของภาษาจาวา ในการแปลงแบบอัตโนมัตินั้นจะตัดมาเฉพาะส่วนที่เป็นบอดีของเอกสาร HTML (อยู่ในแท็ก <body>) แล้วทำการแปลงข้อมูลและแท็กทุกแท็กที่อยู่ภายใน นอกจากนี้ยังมีการแทรกแท็กพิเศษที่กำหนดขึ้นเองเพื่อให้ง่ายต่อการแปลงเป็น WML ได้แก่แท็ก <checkbox> <radio> และ <text> ในกรณีแท็กอินพุตมีชนิดเป็น “checkbox” หรือ “radio” (type=checkbox หรือ type=radio) จะมีการใส่แท็ก <checkbox> และแท็ก <radio> ครอบไว้ตามลำดับ ส่วนอักขระต่างๆ จะใช้แท็ก <text> ครอบไว้ดังตัวอย่าง

```
<body>
.
.
.
status :
<input type=radio name=status value=s> single
<input type=radio name=status value=m> married
<input type=radio name=status value=d> divorced
Countries visited :
<input type=checkbox name=US value=yes> United State
<input type=checkbox name=TH value=yes> Thailand
hobbies :
<input type=checkbox name=hobby value=movie> movie
<input type=checkbox name=hobby value=magazine> magazine
<input type=checkbox name=hobby value=travelling> travelling
.
.
.
</body>
```

รูปที่ 3.3 บางส่วนของเอกสาร HTML ต้นฉบับ

```
<body>
.
.
.
<text>
status :
</text>
<radio>
<input type=radio name=status value=s> <text> single </text>
<input type=radio name=status value=m> <text> married </text>
<input type=radio name=status value=d> <text> divorced </text>
</radio>
<text>
Countries visited :
</text>
<checkbox>
<input type=checkbox name=US value=yes>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        <text> United State </text>
    </checkbox>
    <checkbox>
        <input type=checkbox name=TH value=yes>
        <text> Thailand </text>
    </checkbox>
    <text>
        hobbies :
    </text>
    <checkbox>
        <input type=checkbox name=hobby value=movie>
        <text> movie </text>
        <input type=checkbox name=hobby value=magazine>
        <text> magazine </text>
        <input type=checkbox name=hobby value=travelling>
        <text> travelling </text>
    </checkbox>
    .
    .
    .
</body>

```

รูปที่ 3.4 เอกสาร HTML หลังจากการตัดแบ่งและดัดแปลงให้ง่ายต่อการแปลงเป็น WML

แท็กที่ทำเป็นตัวเอียงเป็นแท็กที่กำหนดขึ้นเองเพื่อให้ง่ายต่อการแปลงเป็น WML จะเห็นได้ว่าแท็ก <checkbox> และแท็ก <radio> จะครอบคลุมอินพุตที่มีค่าแอตทริบิวต์เนม (name) เป็นค่าเดียวกัน เมื่อได้สตริงของ HTML ต้นฉบับที่เหมาะสมแล้วจึงทำการสร้างโครงสร้างต้นไม้ของ XML อิติเมนต์โดยใช้คลาส XmlDocument และ Element ของโคธอนเข้ามาช่วย ตัวอย่างการแปลงสตริงของ HTML ต้นฉบับเป็นโครงสร้างต้นไม้ของ XML อิติเมนต์

```

<body bgcolor="#66CCFF">
  <br>
  <form action=/servlet/CFfDC_loginhandler method=POST>
    <center>
      <text>Please enter your UserName</text>
      <br>
      <text>and Passwords</text>
      <table border=0>
        <tr>
          <td align=right>
            <b>
              <text>User Name : </text>
            </b>
          </td>
          <td>
            <input type=text name="user_name" value="" size=20>
            <font color="red">

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

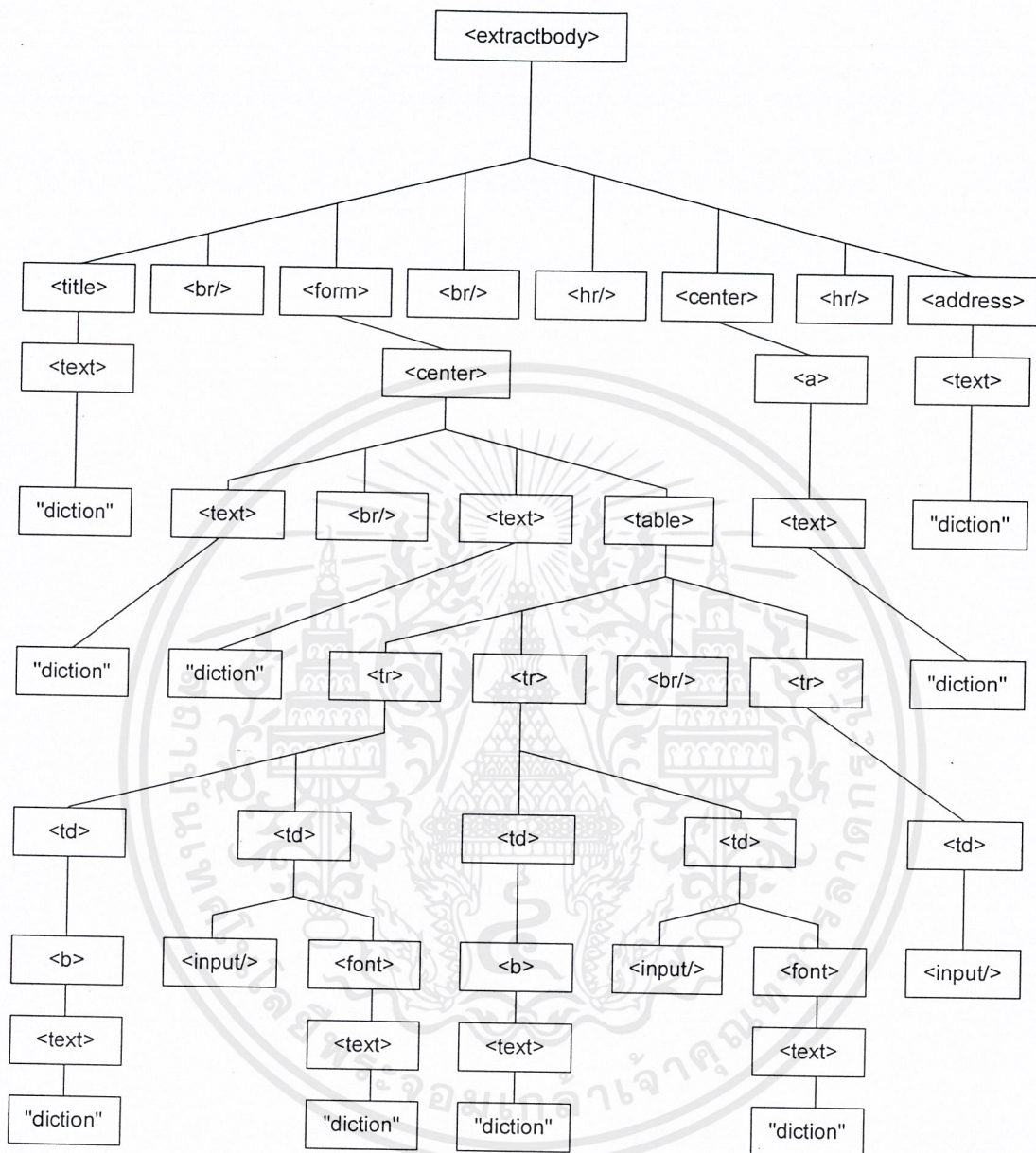
```

        <text>*</text>
    </font>
</td>
</tr>
<tr>
    <td align=right>
        <b>
            <text>User Password : </text>
        </b>
    </td>
    <td>
        <input type=password name="user_password" size=10>
        <font color="red">
            <text>*</text>
        </font>
    </td>
</tr>
<br>
<tr align=center>
    <td colspan=2>
        <input type=submit value=" Submit ">
    </td>
</tr>
</table>
</center>
</form>
<br>
<hr>
<center>
    <a href="/main.html">
        <text>HOME</text>
    </a>
</center>
<hr>
<address>
    <text>Cooperative Framework for Document Composition</text>
</address>
</body>

```

รูปที่ 3.5 ตัวอย่างสตริงของเอกสาร HTML ต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 โครงสร้างต้นไม้ของ XML อิลิเมนต์ที่แปลงมาจากสตรีมในรูปที่ 3.5

จากภาพโครงสร้างต้นไม้สังเกตเห็นได้ว่าที่โหนด (node) ทุกโหนดที่อยู่ปลายสุดของโครงสร้างจะเป็น "diction" และแท็กที่มีเครื่องหมาย "/" ปิดท้าย "diction" หมายถึงข้อความต่างๆ ที่อยู่ในเอกสาร HTML ซึ่งเมื่อเปิดเอกสารดูด้วยบราวเซอร์แล้ว จะปรากฏข้อความนั้นๆ ให้เห็น ส่วนแท็กที่มีเครื่องหมาย "/" ปิดท้ายนั้น หมายถึงแท็กซึ่งไม่มีแท็กปิดและเมื่อไม่มีแท็กปิดจึงหมายความว่าแท็กดังกล่าวไม่สามารถมีโหนดลูก (child node) ต่อไปได้ หลังจากได้โครงสร้างต้นไม้ของ XML อิลิเมนต์แล้ว นำโครงสร้างที่ได้ไปจัดการต่อโดยใช้ XSLT เข้ามาช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การจัดการกับ XML

ในขั้นตอนนี้เป็นส่วนกำหนดกฎและเทคนิคในการโปรแกรม XSLT เพื่อแปลงโครงสร้างและข้อมูลจากภาษา XML (ซึ่งถูกแปลงมาจาก HTML โดยพยายามคงโครงสร้างให้เหมือนกับ HTML มากที่สุด) ให้อยู่ในรูปแบบของภาษา WML แต่ในบางแท็ก HTML ก็ไม่สามารถหาแท็ก WML ใดมาเสมือนได้จึงจำเป็นต้องตัดออกไป เนื่องจากอาจเกิดข้อผิดพลาดได้ แม้ว่าบางอติเม้นต์จะได้รับการสนับสนุนและเป็นส่วนหนึ่งของสเปก WML แต่ไม่จำเป็นว่าโทรศัพท์ทุกเครื่องจะรองรับได้ เช่นในโทรศัพท์รุ่นแรกๆ รองรับได้เพียงฟอนต์เดียวเท่านั้น ดังนั้นตัวอักษรแบบต่าง ๆ เช่นตัวหนา ตัวเอียง ตัวเล็ก จะไม่สามารถทำงานได้ในหลาย ๆ โทรศัพท์ และส่วนมากยังไม่รองรับรูปแบบอีกด้วย

XML	WML	หน้าที่
*<a href	<do><go><postfield> <fieldset><input> 	ใช้สร้างลิงก์หรือสร้างจุดเชื่อมต่อที่จะอ้างไปยังเว็บเพจอื่น
<address>	<i>	ใช้ครอบข้อความที่เป็นอีเมลแอดเดรส โดยจะแสดงเป็นตัวเอียง
		ใช้ครอบข้อความเพื่อทำให้ข้อความนั้นเป็นตัวหนา
<blockquote>	 “ “	ใช้ครอบข้อความที่เป็นคำพูด โดยแยกข้อความนั้นออกจากข้อความส่วนอื่น แล้วนำมาขึ้นย่อหน้าใหม่
<big>	<big>	ใช้เน้นข้อความที่ต้องการให้มีขนาดใหญ่กว่าข้อความขนาดมาตรฐาน
 	 	กำหนดให้บรรทัดขึ้นบรรทัดใหม่
<cite>	<i>	ใช้ครอบข้อความที่ต้องการสื่อว่าเป็นคำที่ขี้มมาใช้หรือเอกสารที่ใช้อ้างอิง มักแสดงเป็นตัวเอียง
<dd>	 	ใช้แสดงส่วนที่เป็นคำอธิบายของแต่ละรายการย่อย
	<do> type label Name	ใช้ให้ผู้ใช้สามารถเข้าถึงและเชื่อมโยงจากการ์ดที่แสดงอยู่ไปสู่การ์ดหรือURI อื่นๆ ได้
<dt>	 	ใช้กำหนดแต่ละรายการย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ใช้ครอบข้อความที่ต้องการเน้นเป็นพิเศษ
	<fieldset>	การจัดกลุ่มให้กับส่วนรับข้อมูล
*<frame> src name	<do><go><postfield> <fieldset><input> 	ใช้กำหนดข้อมูลที่จะถูกแสดงไว้ในเฟรม
*<form> action method	<do> <go> <postfield>	ใช้ครอบข้อมูลและแท็กที่ใช้สร้างฟิลด์รับข้อมูลที่ประกอบกันเป็นฟอร์ม และยังเป็นตัวกำหนดว่าจะส่งข้อมูลให้เว็บเซิร์ฟเวอร์อย่างไรและเรียกใช้โปรแกรมตัวใดทำงานด้วย
	<go> href method accept-charset	ใช้กำหนดให้บราวเซอร์เว็บเปลี่ยนหน้าของเว็บเพจไปตามตำแหน่งที่กำหนด
<h1> - <h6>	<big>	ใช้จัดข้อความที่เป็นหัวข้อให้มีขนาดตัวอักษรใหญ่และหนากว่าปกติ
<hr>	 ----- 	สร้างเส้นคั่นเป็นแนวนอน เพื่อแบ่งข้อมูลออกเป็นส่วนๆอย่างชัดเจน
<i>	<i>	ใช้ครอบข้อความที่ต้องการเน้นให้เป็นตัวเอียง
 src width height alt align [top middle bottom] hspace vspace	 src width height alt align [top middle bottom] hspace vspace	ใช้วางรูป โดยไฟล์ภาพอาจเป็น .gif หรือ .jpeg ก็ได้
<input> name value type = checkbox	<select> name multiple = true <option> value	ใช้สร้างฟิลด์รับข้อมูลแบบ ให้ผู้ใช้ต้องคลิกเลือกหรือไม่เลือกรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<input type="radio"/> name value type = radio	<select> name multiple = false <option> value	ใช้สร้างฟิลด์รับข้อมูลแบบมีตัวเลือกหลายตัว แต่ผู้ใช้จะเลือกได้เพียงตัวเดียวเท่านั้น
<input type="text"/> name value type [text password] length maxlength	<input type="text"/> name value type [text password] size maxlength	ใช้สร้างฟิลด์รับข้อมูลชนิด text : รับข้อความบรรทัดเดียว password : รับข้อมูลแบบ text แต่จะแสดงข้อความหรือรหัสผ่านเป็นเครื่องหมาย *
	 -	ใช้แสดงรายการย่อย โดยแสดงหนึ่งบรรทัดต่อหนึ่งรายการ
<option value="...">	<option value="...">	สร้างตัวเลือกในฟิลด์แบบ <select>
<optgroup label="...">	<optgroup title="...">	จัดกลุ่มให้กับตัวเลือก <option>
<p align="left right center">	<p align="left right center">	ใช้ครอบข้อความเพื่อสร้างย่อหน้าใหม่
	<postfield name="..." value="...">	ใช้ในการระบุชื่อตัวแปรและค่าของมันในการส่งข้อมูล
<q>	“ ”	ใช้ครอบข้อความที่อ้างมาจากแหล่งอื่น โดยจะแสดงให้เครื่องหมายคำพูด
<select multiple="true">	<select multiple="true">	สร้างฟิลด์แบบมีหลายตัวเลือก โดยสามารถเลือกตัวเลือกเหล่านั้นได้ครั้งละ 1 หรือหลายตัวเลือกก็ได้
<select multiple="true"> <option value="...">	<select multiple="true"> <option value="...">	<input type="checkbox"> ใช้สร้างฟิลด์รับข้อมูลแบบ ให้ผู้ใช้ต้องคลิกเลือกหรือไม่เลือกรายการนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<select> name multiple = false <option> value	<select> name multiple = false <option> value	<input type="radio"> ใช้สร้างฟิลด์รับข้อมูลแบบมีตัวเลือกหลายตัว แต่ผู้ใช้จะเลือกได้เพียงตัวเดียวเท่านั้น
<table> cols	<table> columns	แท็กสร้างตาราง ใช้ครอบข้อมูลและแท็กทั้งหมดที่ใช้ตาราง
<td>	<td>	ใช้กำหนดเซลล์ข้อมูลของตาราง
<textarea> name	<input/> name type = text	ใช้สร้างฟิลด์รับข้อความ
<th>	<td> 	ใช้กำหนดเซลล์ที่เป็นหัวตาราง ซึ่งถูกแสดงเป็นตัวหนา
<tr>	<tr>	ใช้กำหนดแถวของตาราง
<u>	<u>	ใช้ครอบข้อความที่ต้องการให้มีเส้นขีดใต้ข้อความนั้น

ตารางที่ 3.1 แสดงการแมประหว่างแท็กภาษา XML เป็นภาษา WML

* คือแท็ก XML ที่จำเป็นต้องมีการแปลงเป็นพิเศษ เพื่อให้เหมือนกับแท็กของ WML

แท็กลิงก์ <a>

XML :

```
<a href="URL_OF_linked_document"> .. hyperlinked element goes here .. </a>
```

WML :

```
<do type="accept" label="Goto hyperlinked element" name="hyperlinked element">
  <go href="http://161.246.5.113:8080/test/servlet/Handlerwap" method="get">
    <postfield name="url" value="URL_OF_linked_document"/>
  </go>
</do>
<fieldset title="url">
  <input type=text name="hyperlinked element" emptyok=true value="hyperlinked" />
</fieldset>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แท็กเฟรม <frame>

XML :

```
<frame src="URL_of_document" name="frame_name">
```

WML :

```
<do type="accept" label="Goto frame_name" name="frame_name">
  <go href="http://161.246.5.113:8080/test/servlet/Handlerwap" method="get">
    <postfield name="url" value=" URL_of_document "/>
  </go>
</do>
<fieldset title="url">
  <input type="text" name="frame_name" emptyok=true value="frame_name" />
</fieldset>
<br/>
```

แท็กฟอร์ม <form>

XML :

```
<form action="http://161.246.5.113:8080/test/servlet/WapForm" method="get">
</form>
```

WML :

```
<do type="accept" label="Submit">
  <go href="http://161.246.5.113:8080/test/servlet/WapForm" method="get">
    <postfield name="variable_from_input" value="$value_from_input"/>
  </go>
</do>
```

ข้อจำกัดของ WML ที่แตกต่างจาก HTML

- ต้องปิดแท็ก WML ทุกครั้ง
- การเขียนค่าของแอตทริบิวต์ต้องมีเครื่องหมาย “ หรือ ‘ กำกับด้านหน้าและหลังเสมอ
- แท็กมีลักษณะเป็น case sensitive คือ ความหมายของตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่จะไม่เหมือนกัน
- ไม่สามารถใส่แท็กตารางซ้อนตารางได้ และแท็กตารางต้องมีแอตทริบิวต์ columns ทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การสร้างโค้ดภาษา WML

จากที่ได้อธิบายข้างต้น ในการสร้าง WML จาก XML นั้นจะถูกกระทำโดยโคจูนของอาปาเช่ (Apache's Cocoon) โดยที่โคจูนคือจาวาเซิร์ฟเลตตัวหนึ่ง ดังนั้นในการรันจึงต้องทำการเพิ่ม โคจูนเซิร์ฟเลตให้กับเว็บเซิร์ฟเวอร์ก่อน ในการทำงานเมื่อมีการร้องขอ HTTP มายังเว็บเซิร์ฟเวอร์ มันจะถูกส่งมาที่โคจูน ซึ่งมีหน้าที่รับการร้องขอ จากนั้นจะทำการประมวลผล และส่งหน้าผลลัพธ์กลับ โดยขึ้นกับชนิดของบราวเซอร์ การทำงานโดยละเอียดของอาปาเช่โคจูนประกอบด้วย 3 ส่วน ในการส่งกลับข้อมูล (content), ตรรกะ (logic) และ รูปแบบ (style)

3.4.3.1 Producer

เป็นตัวสร้างเอกสาร XML ซึ่งในการเขียนต้องสอดคล้องกับ API ของโปรคิวเซอร์โคจูน จากไฟล์ HTMLExtractor จะถูกมองว่าเป็นโปรคิวเซอร์ โปรคิวเซอร์ดีฟอลต์ที่สนับสนุนโคจูนคือ ProducerFromFile และ ProducerFromRequest โดยในการทำงาน ProducerFromFile จะส่งเอกสาร XML จากไฟล์ให้กับโคจูน โดยชื่อไฟล์จะถูกส่งเป็นพารามิเตอร์ของโปรคิวเซอร์ให้กับคลาส HttpServletRequest มาตรฐาน ส่วน ProducerFromRequest นั้น XML จะถูกฝัง (embed) ในตัว HttpServletRequest อยู่แล้ว อย่างเช่นถ้าต้องการสร้างโปรคิวเซอร์อันหนึ่งขึ้นมา เราต้องทำการกำหนดวิธี extract ข้อมูลจากไฟล์ HTML ให้กับโคจูนก่อน ซึ่งจะกล่าวถึงภายหลัง ในการเรียกใช้ โปรคิวเซอร์จะถูกระบุในการร้องขอ HTTP ไปยังโคจูนเซิร์ฟเลต เช่น <http://<hostname:port>/dummy.xml?producer=dummy> ดัมมี่ (dummy) เป็นตัวอย่างหนึ่งของโคจูนโปรคิวเซอร์ จาก URL ดังกล่าวจะกำหนดให้โคจูนเรียกใช้ดัมมี่โปรคิวเซอร์จัดการกับการร้องขอนี้และกำหนดวิธีในการแปล (interpret) ส่วนที่เหลือของ URL ในการระบุบริการที่จะใช้ในการแปล ซึ่งก็คือส่วน dummy.xml แต่ในกรณีดัมมี่โปรคิวเซอร์จะไม่ทำในส่วนนี้ และผู้เขียนโคจูนยังเสนอการใช้โปรคิวเซอร์อินเทอร์เฟซเพื่อส่ง XML ไปยังโคจูนหรืออาจกล่าวได้ว่าเป็นการส่งเอาต์พุตของเซิร์ฟเลตหนึ่งไปยังโคจูนเซิร์ฟเลต (หรือที่เรียกว่า servlet chaining)

3.4.3.2 Processor

Processor เป็นตัวในการประมวลผลต้นไม้ DOM (DOM tree) ซึ่งถูกสร้างขึ้นมาจากเอกสาร XML ที่สร้างโดยโปรคิวเซอร์ มีตัวประมวลผลมากมายที่ใช้ร่วมกับโคจูน รวมถึงตัวประมวลผล SQL ด้วย ในแต่ละหน้าจะระบุว่าใช้ตัวประมวลผลใดโดยคำสั่งประมวลผล (Processing Instruction) ของ XML เช่น ระบุให้ตัวประมวลผลคือ XSLT processor :

```
<?cocoon-process type="xslt"?>
```

ตัวประมวลผล XSL มาตรฐานที่สนับสนุนโคจูน คือ Xalan ซึ่งในการใช้นั้นต้องทำการเพิ่มบรรทัดข้างต้น เพราะในการสร้างเอกสาร XML ต้องทำการระบุ XSL สไตลชีตที่จะให้ตัวประมวลผล XSLT ทำงานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<?xml-stylesheet href="portfolio-wml.xml" type="text/xsl" media="wap"?>
```

เอกสาร XSL คือ portfolio-wml.xml ซึ่งเมธอด Service.createDOM() จะอ่านคำสั่งประมวลผลเหล่านี้จากไฟล์คุณสมบัติ (properties file) ของ HTMLExtractor ข้างล่างเป็นส่วน extract ที่สำคัญในไฟล์คุณสมบัติ ซึ่งจะเป็นตัวกำหนดสไตล์ชีตและตัวประมวลผล

...

```
distributors.pi1.target=xml-stylesheet
```

```
distributors.pi1.data=href=file:///c:/Cocoon/samples/wap/contact-wml.xml
```

```
type="text/xsl" media="wap"
```

...

```
distributors.pi3.target=cocoon-process
```

```
distributors.pi1.data=type="xslt"
```

แอตทริบิวต์ของมีเดีย (media) คือเว็บ ซึ่งแอตทริบิวต์นี้จะเป็นตัวบอกโคจูนในการจับคู่ XSL สไตล์ชีตให้เหมาะสมกับ HTTP ยูสเซอร์เอเจนต์ (user agent) ในการแมปนี้จะถูกระบุในไฟล์คอนฟิกูเรชันของโคจูน (cocoon configuration file) ชื่อ conf.properties ในกรณีของโคจูนนั้น wap จะถูกแมปกับส่วนหัวของ HTTP ยูสเซอร์เอเจนต์ (user-agent HTTP header) ของ Nokia-WAP-Toolkit

3.4.3.3 Formatter

Formatter เป็นตัวกำหนดรูปแบบของเอกสาร แต่เนื่องจาก XSL สไตล์ชีต ที่ระบุโดยตัวประมวลผล XSLT ทำงานที่จำเป็นทั้งหมดแล้ว ดังนั้นในกรณีนี้จึงไม่จำเป็นต้องใช้ในส่วนของนี้ อย่างไรก็ตามออปชันนี้จะเป็ประโยชน์สำหรับเอาต์พุตที่เป็น PDF หรือรูปแบบอื่นๆ ที่ต้องการ โดยจะมีรูปแบบคล้ายกับรูปแบบของ คำสั่งประมวลผล (Processing Instruction)

```
<?cocoon-format type="yyy"?>
```

สำหรับการกำหนดรูปแบบ Formatter เป็นทางเลือกหนึ่งใน XML โดนที่ค่าดีฟอลต์ คือ text/html ดังนั้นในการสร้างเอาต์พุตเป็น WML จึงต้องระบุสิ่งต่อไปนี้ใน WML สไตล์ชีต ดังในตัวอย่างของ contact-wml.xml :

```
<xsl:processing-instruction name="cocoon-format">type="text/wml"</xsl:processing-instruction>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

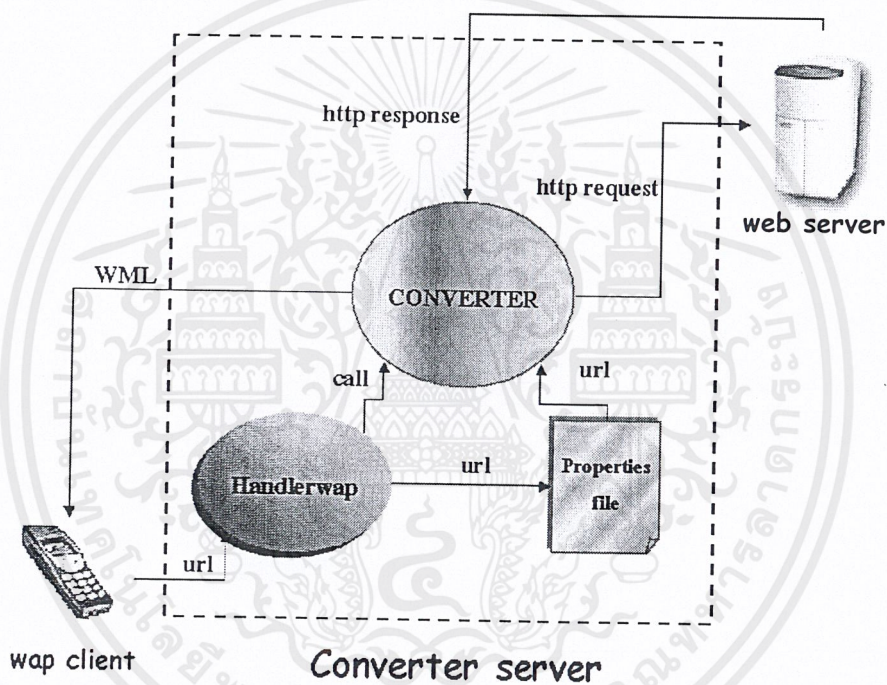
บทที่ 4

อิมพลีเม้นเตชัน

ในการทำงานจริงจะซับซ้อนกว่าทฤษฎีที่กล่าวไว้ในบทที่ 3 เนื่องจากจะมีการใช้แท็กลิงก์, เฟรม และฟอร์มจะต้องทำการแปลงแบบพิเศษ เพื่อให้การทำงานเป็นไปได้ถูกต้องและมีประสิทธิภาพ

4.1 การทำงานทั่วไป

ตัวแปลงที่สร้างขึ้นจะถูกฝังไว้ในเว็บเซิร์ฟเวอร์ซึ่งในที่นี้จะเรียกว่า “คอนเวอร์เตอร์เซิร์ฟเวอร์” (Converter Server) เพื่อไม่ให้สับสนกับเว็บเซิร์ฟเวอร์ซึ่งให้บริการเว็บเพจอยู่



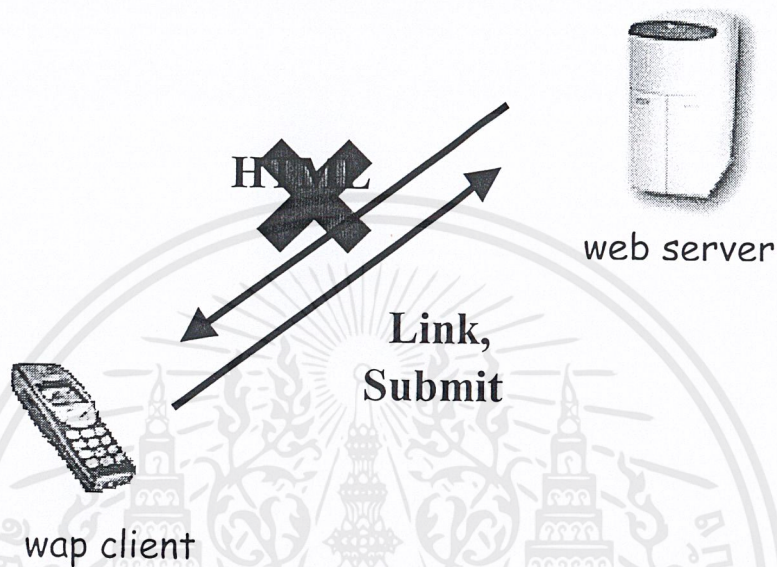
รูปที่ 4.1 การทำงานทั่วไปในการแปลง

ขั้นตอนการทำงาน

- 4.1.1 เมื่อผู้ใช้กรอกค่า URL ที่ต้องการแปลงผ่านเว็บเพจที่สร้างขึ้นเพื่อให้บริการการแปลงแล้วทำการ submit เว็บเบราว์เซอร์จะส่งการร้องขอ HTTP ซึ่งการร้องขอดังกล่าว จะมีการส่งค่า URL ที่ผู้ใช้กรอกมายังเซิร์ฟเลตที่ชื่อ Handlerwap
- 4.1.2 Handlerwap จะนำค่า URL ที่ได้ เขียนลงไฟล์คุณสมบัติ (properties file) จากนั้นจะเรียกตัวแปลง
- 4.1.3 ตัวแปลงจะดึงค่า URL ที่ต้องแปลงมาจากไฟล์คุณสมบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.1.4 หลังจากได้ค่า URL มาจากไฟล์คุณสมบัติแล้ว ตัวแปลงจะทำการส่งการร้องขอ HTTP ไปยังเว็บเซิร์ฟเวอร์
- 4.1.5 เว็บเซิร์ฟเวอร์ส่งการตอบรับกลับมาโดยอยู่ในรูปแบบ HTML
- 4.1.6 ตัวแปลงภาษาจะทำการแปลงจากผลลัพธ์ที่ได้จากการตอบสนองเป็น WML แล้วส่งผลลัพธ์ที่ได้กลับให้เว็บเบราว์เซอร์



รูปที่ 4.2 การร้องขอจากเว็บไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์

จากรูปที่ 4.2 การลิงก์หรือ submit จากเว็บเบราว์เซอร์โดยตรงไปยังเว็บเซิร์ฟเวอร์นั้น ไม่สามารถทำได้ เนื่องจากเว็บเซิร์ฟเวอร์จะส่งการตอบสนองซึ่งอยู่ในรูปแบบของเอกสาร HTML กลับมายังไคลเอนต์ตัวที่ทำการร้องขอไป ในกรณีนี้ไคลเอนต์ดังกล่าวได้แก่ เว็บเบราว์เซอร์ซึ่งไม่สามารถรองรับการตอบสนองในลักษณะดังกล่าวได้ จึงต้องมีการใช้เซิร์ฟเวทช่วยในการทำงาน

4.2 การทำงานที่มีแท็กลิงก์หรือเฟรม

เนื่องจากเว็บไคลเอนต์ไม่สามารถแสดงผลในรูปแบบของเฟรมได้ ดังนั้นเมื่อมีแท็ก <frame> ปรากฏอยู่ในเอกสาร HTML จะมีการใช้เทคนิคในการแปลงคือเปลี่ยนเฟรมต่าง ๆ ให้เป็นลิงก์ ผู้ใช้สามารถเลือกดูแต่ละเฟรมได้

ขั้นตอนการลิงก์นั้นเหมือนกับการทำงานทั่ว ๆ ไป ต่างกันที่การรับค่า URL ที่ต้องการแปลงนั้น ไม่ได้มาจากการกรอกของผู้ใช้ แต่ได้มาจากการแปลงแท็ก <a> และแท็ก <frame> ของเอกสาร HTML ให้เป็น WML ตามรูปแบบที่ได้อธิบายไว้ในบทที่ 3

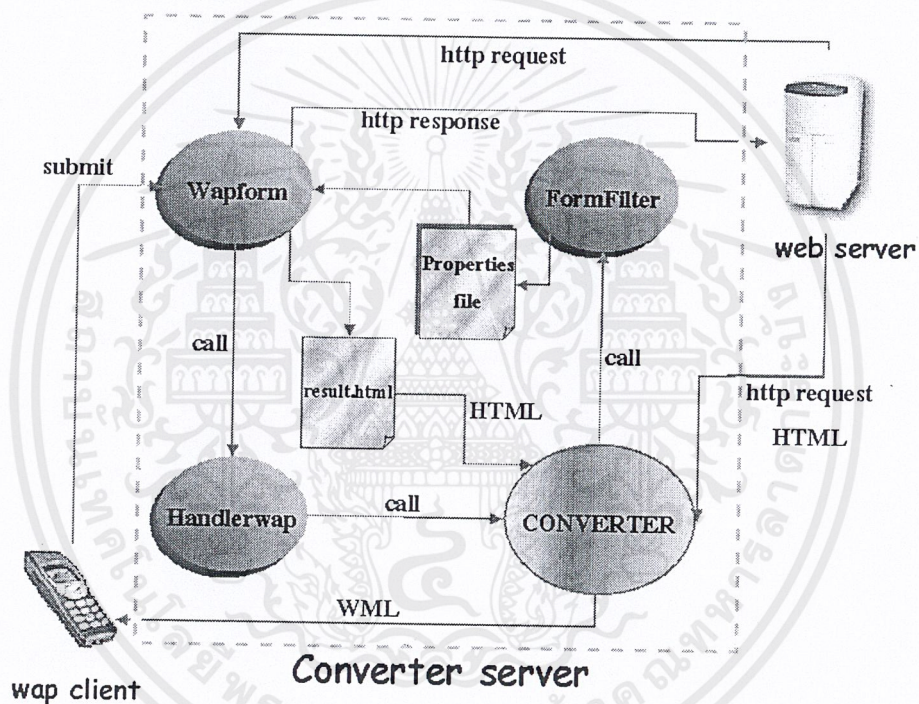
ขั้นตอนการทำงาน

- 4.2.1 เมื่อมีการเลือกลิงก์ เว็บเบราว์เซอร์ส่งการร้องขอ HTTP ซึ่งมีค่าตัวแปร URL เท่ากับ URL ของลิงก์นั้นให้กับ Handlerwap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.2.2 Handlerwap จะนำค่า URL ที่ได้ เขียนลงไฟล์คุณสมบัติ (properties file) จากนั้นจะเรียกตัวแปลง
- 4.2.3 ตัวแปลงจะดึงค่า URL ที่ต้องแปลงมาจากไฟล์คุณสมบัติ
- 4.2.4 หลังจากได้ค่า URL มาจากไฟล์คุณสมบัติแล้ว ตัวแปลงจะทำการส่งการร้องขอ HTTP ไปยังเว็บเซิร์ฟเวอร์
- 4.2.5 เว็บเซิร์ฟเวอร์ส่งการตอบรับกลับมาโดยอยู่ในรูปแบบ HTML
- 4.2.6 ตัวแปลงภาษาจะทำการแปลงจากผลลัพธ์ที่ได้จากการตอบสนองเป็น WML แล้วส่งผลลัพธ์ที่ได้กลับให้เว็บเบราว์เซอร์

4.3 การทำงานที่มีแท็กฟอร์ม



รูปที่ 4.3 การทำงานในการแปลงแท็กฟอร์ม

ขั้นตอนการทำงาน

- 4.3.1 เมื่อเว็บเซิร์ฟเวอร์ส่งการตอบรับกลับมา แล้วภายในเอกสาร HTML มีการใช้แท็ก <form> เพื่อรับอินพุตให้ CGI ประมวลผล ก่อนทำการแปลงตัวแปลงจะเรียกแอปพลิเคชันที่ชื่อ FormFilter เพื่อเก็บรายละเอียดต่าง ๆ ที่จำเป็นต่อการ submit ฟอร์ม เช่น ค่าแอตทริบิวต์ action, เมธอดที่ใช้, ตัวแปรต่าง ๆ ที่ต้องส่งให้ CGI ไว้ในไฟล์คุณสมบัติ
- 4.3.2 ตัวแปลงทำการแปลง HTML เป็น WML ส่งให้แก่เว็บไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.3.3 เมื่อมีการอินพุตฟอร์มแล้วทำการ submit เว็บเบราว์เซอร์ส่งการร้องขอ HTTP ไปยังเซิร์ฟเวอร์ที่ชื่อ Wapform ซึ่งจะรับค่าตัวแปรต่าง ๆ ไว้ แล้วดึงรายละเอียดที่ต้องใช้ในการสร้างการร้องขอ HTTP จากไฟล์คุณสมบัติ จากนั้นจึงสร้างการร้องขอ HTTP ไปยังเว็บเซิร์ฟเวอร์
- 4.3.4 Wapform จะรอการตอบสนองจากเว็บเซิร์ฟเวอร์ เมื่อได้รับเรียบร้อยแล้วจะบันทึกการตอบสนองที่ได้กลับมาลงไฟล์ที่ชื่อ result.html
- 4.3.5 จากนั้น Wapform ทำการเรียก Handlerwap เพื่อให้ Handlerwap เรียกใช้ตัวแปลง
- 4.3.6 ตัวแปลงแปลงการตอบสนองที่ได้จากเว็บเซิร์ฟเวอร์ด้วยการนำไฟล์ result.html ที่ Wapform ได้บันทึกไว้มาทำการแปลงแล้วจึงส่งผลลัพธ์ที่เป็น WML ให้กับเว็บไคลเอนต์



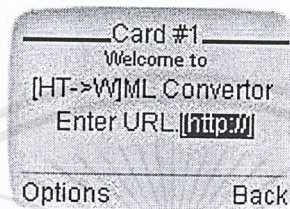
บทที่ 5

ผลการทดลองและการเปรียบเทียบ

5.1 การใช้งาน

ในการใช้งานสามารถเรียกผ่านบราวเซอร์ใด ๆ ผ่านทาง URL :

“ <http://161.246.5.113:8080/test/convert2wml.wml> “ โดยการพิมพ์ URL ที่ต้องการแปลงลงในฟอร์ม

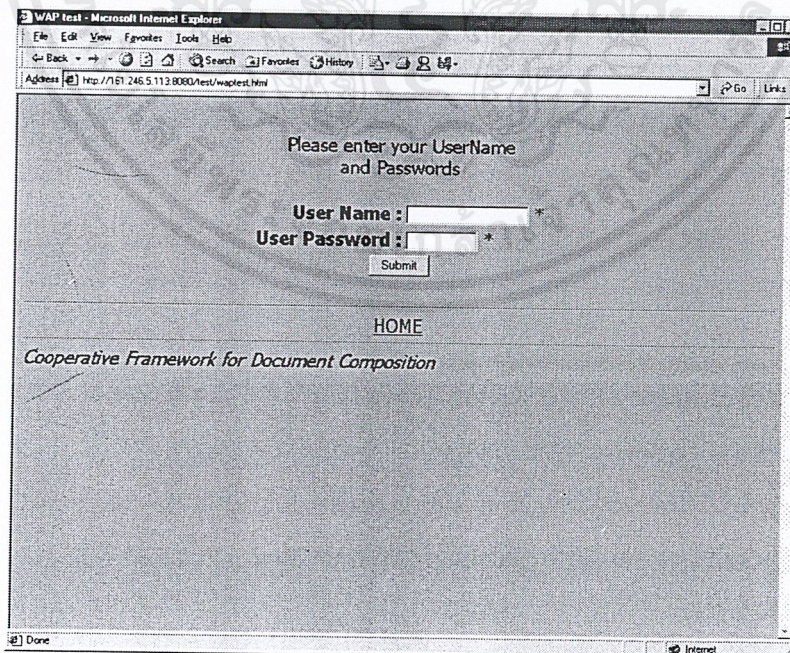


รูปที่ 5.1 โฮมเพจของ [HT --> W] Converter

5.2 ตัวอย่างการแปลง

แสดงผลโดยใช้บราวเซอร์ IE5.5 และ อีมีเลเตอร์โนเกีย ตามลำดับ

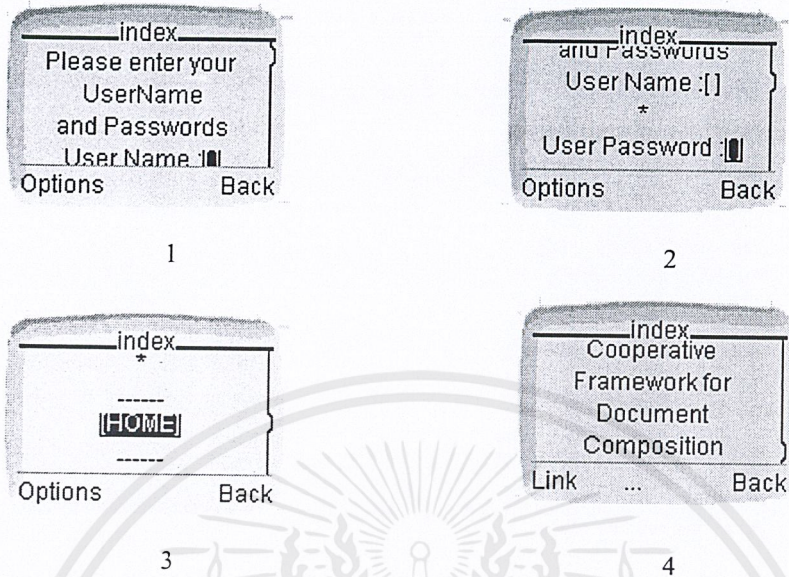
การแสดงผลหน้า HTML บนเว็บไคลเอ็นต์



รูปที่ 5.2 การแสดงผลเว็บเพจ ภาษา HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในข้อมูลเดียวกัน เมื่อแปลงเป็น WML เพื่อแสดงบนเว็บโคลเอนต์ ได้ดังต่อไปนี้



รูปที่ 5.3 การแสดงผลเว็บเพจ ภาษา WML โดยการแปลงจาก HTML

5.3 การเปรียบเทียบ

เป็นการเปรียบเทียบตัวแปลงภาษาแบบอัตโนมัติอย่างสมบูรณ์ระหว่างตัวแปลงภาษาของเรา ([HT -> W] ML Converter) กับตัวแปลงภาษาอื่นที่ปรากฏ ดังต่อไปนี้

5.3.1 Phone.com WAP Gateway (HTTP://www.phone.com)

Phone.com จะรวมตัวแปลงภาษาแบบอัตโนมัติไว้กับเว็บเกตเวย์ของตัวเอง ซึ่งจะทำการตัดแบ่งไตเติ้ล, ข้อความและลิงก์ทั้งหมด แล้วนำมาแสดงผลที่บนอุปกรณ์เว็บ ถ้าข้อความมีขนาดใหญ่กว่าเด็กจะสามารถรับได้ (~1400) จะทำการแบ่งผลลัพธ์ออกเป็นหลาย ๆ เด็กและลิงก์กันด้วยปุ่ม More นอกจากนี้ยังสามารถรองรับฟอร์มอินพุตและลิงก์ของ HTML แต่ไม่สามารถทำการแปลงเฟรมหรือการแปลงรูปภาพแบบไดนามิกได้

เกตเวย์จะยอมให้สามารถใส่เป็น URL แบบสั้นได้โดยจะทำการเติม HTTP:// ไว้ข้างหน้า และเติม .com ไว้ข้างหลัง แม้ว่ามันจะไม่ใช่ว่าฟังก์ชันการแปลงที่แท้จริง แต่ก็เป็นที่เอร์พื้นฐานที่ทุกตัวแปลงแบบอัตโนมัติจะมี เป็นการเพื่อเป็นการเสริมการใช้งานให้ง่ายขึ้น

น่าเสียดายที่ตัวแปลงภาษานี้ไม่สามารถที่จะแยกออกจากเกตเวย์ได้และเกตเวย์เป็นซอฟต์แวร์ที่มีราคาแพงมากกว่าที่จะซื้อเพราะต้องการแก้ตัวแปลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบ

(ในการเปรียบเทียบเฉพาะสิ่งที่แตกต่างกันเท่านั้น)

Phone.com	[HT -> W] ML Converter
<ul style="list-style-type: none"> - ไม่สามารถแปลงเฟรมได้ - สามารถแบ่งหน้าเพจที่มีขนาดใหญ่ให้มีขนาดเหมาะสมกับเค็ต์ได้ <p>การใช้ได้ต้องเรียกผ่านเกตเวย์ของ Phone.com เท่านั้น</p>	<ul style="list-style-type: none"> - ทำการแปลงเฟรมได้ - แสดงข้อมูลทั้งหมดใน 1 เค็ต์ - ใช้จากที่ใดก็ได้ผ่านทางเว็บเพจของเรา

ตารางที่ 5.1 การเปรียบเทียบกับ Phone.com

5.3.2 Argo ActiGate (HTTP://www.argogroup.com)

ActiGate ไม่ได้เป็นแค่ตัวแปลงภาษาของเว็บเท่านั้นแต่ยังสามารถแปลงจาก HTML เป็นได้ทั้ง WML, HDML และ XML สำหรับ PDA ที่มีหน้าจอและกำลังในการประมวลผลที่จำกัด ActiGate มีการพัฒนาไปมากกว่า Phone.com เล็กน้อยและสามารถแปลงรูปภาพเป็นแบบโมนโครมได้ จากไฟล์ .jpeg เป็น .gif และแปลงให้อยู่ในสเกลและสีที่เหมาะสม ซึ่งเป็นการช่วยเพิ่มความเร็วในการดาวน์โหลด แต่ถ้าอุปกรณ์ไม่สามารถแสดงรูปภาพได้จะทำการแสดงไต่เต้รูปภาพแทน ยังไม่ปรากฏว่าสามารถรองรับไฟล์ .wbmp ได้

ชื่อของลิงก์จะถูกแปลงเป็นไต่เต้ของเพจที่ต้องการลิงก์ เพื่อช่วยให้มันสามารถอ่านได้ ยกตัวอย่างเช่นลิงก์ที่มีข้อความที่ไม่เหมาะสมอยู่ด้วย เช่น './test/index.htm' ในกรณีนี้ Actigate จะทำการอ่านหน้าเพจของ test/index.htm และใช้ไต่เต้นี้เป็นข้อความในการลิงก์แทน นอกจากนี้ จะทำการแปลงลิงก์จาวาสคริปต์ (แต่ไม่ใช่ตัว JavaScript) และเมื่อพบว่ามีลิงก์ซ้ำจะเอาออกไป

เพจที่มีขนาดใหญ่จะถูกแบ่งออกเป็นหลายๆ หน้าเพจ ที่มีขนาดเล็กลงและมีการลิงก์ต่อกันไว้ และในการแปลงเอกสารจะถูกแคชเอาไว้เพื่อเพิ่มประสิทธิภาพ และยังสนับสนุนในการแปลงเฟรมด้วย นอกจากนี้ เนื่องจากมันสนับสนุนในการแปลงจาก HTML เป็น HTML ด้วยดังนั้นมันจึงมีฟังก์ชันคลีนอัพในการแก้ไขข้อผิดพลาดต่างที่พบในหน้าเว็บ HTML เช่น nested structure คือ การเปิดแท็กแล้วไม่ได้ปิดแท็กนั้น หรือมีการปิดแท็กผิดลำดับ ซึ่งสามารถมั่นใจได้ว่าเป็น HTML ที่ well-formed หรือ รูปแบบที่ถูกต้องสมบูรณ์ เหมือนกัน XHTML และ XML

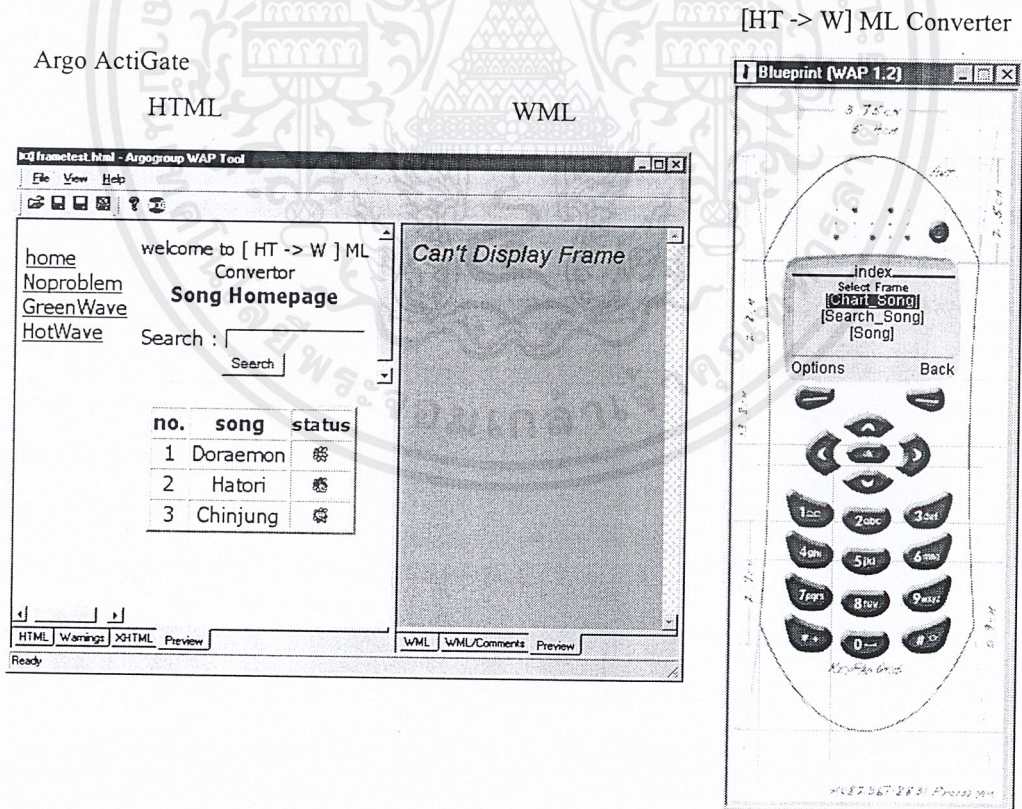
การเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Argo ActiGate	[HT -> W] ML Converter
<ul style="list-style-type: none"> - แปลงจากภาษา HTML เป็นภาษา WML, HDML และ XML - สามารถแปลงรูปแบบไฟล์ภาพได้ - สามารถแบ่งหน้าเพจที่มีขนาดใหญ่ให้มีขนาดเหมาะสมกับเด็กได้ - มีการแคชไฟล์ที่ถูกแปลง ดังนั้นเมื่อเพจที่นำมาแปลงมีการเปลี่ยนแปลง ก็จะทำได้ข้อมูลที่ผิดพลาด - สามารถแปลงไฟล์ HTML ที่ไม่อยู่ในรูป well-form HTML ได้ 	<ul style="list-style-type: none"> - แปลงได้จาก HTML เป็น WML เท่านั้น - ไม่สามารถแปลงไฟล์รูปภาพได้ - แสดงข้อมูลทั้งหมดใน 1 เด็ค - มีการดึงไฟล์ที่ต้องการแปลงมาทำการแปลงใหม่ทุกครั้ง ทำให้ได้ข้อมูลใหม่เสมอ <p>แปลงได้เฉพาะไฟล์ HTML ที่เป็น well-form HTML</p>

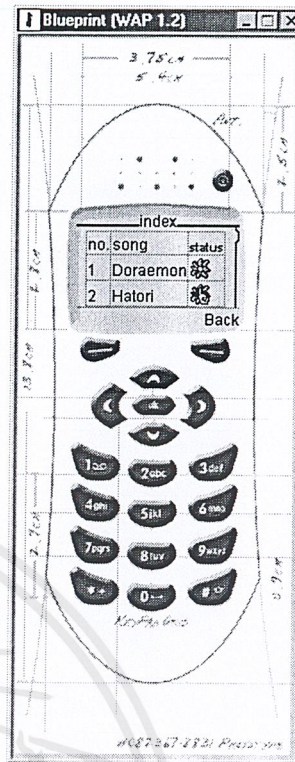
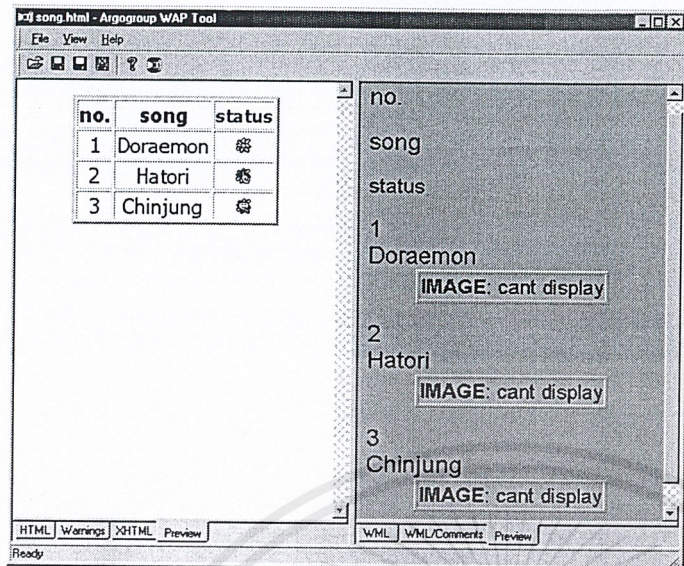
ตารางที่ 5.2 การเปรียบเทียบกับ Argo ActiGate

รูปภาพ ตัวอย่างแสดงการเปรียบเทียบการแสดงผลระหว่างตัวแปลงภาษา Argo ActiGate และ [HT -> W] ML Converter



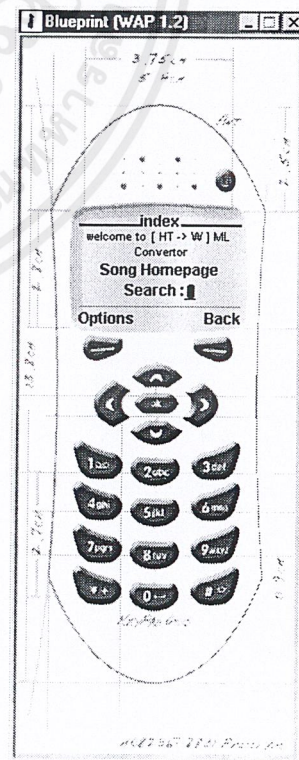
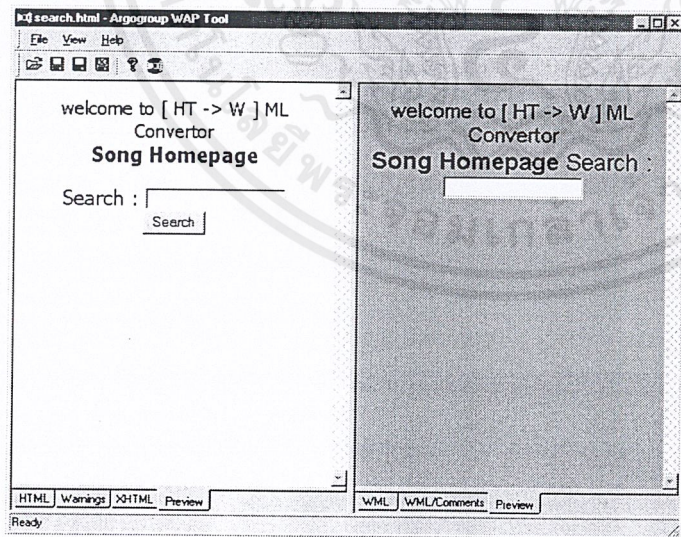
รูปที่ 5.4 ตัวอย่างเปรียบเทียบการแสดงผลการแปลงเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 ตัวอย่างเปรียบเทียบการแสดงผลการแปลงตาราง

* การแปลงแสดงรูปภาพของ [HT -> W] ML Converter เป็นฟังก์ชันของอิมูเลเตอร์โนเกีย



รูปที่ 5.6 ตัวอย่างเปรียบเทียบการแสดงผลการแปลงอินพุตฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิเคราะห์และสรุป

6.1 ข้อดี

ข้อดีหลักของการแปลงภาษา นอกเหนือจากการไม่ต้องสร้างไชต์ขึ้นมาใหม่ คือ

- ช่วยให้การดาวน์โหลดเว็บเพจอย่างรวดเร็ว
- ลดค่าใช้จ่าย
- ช่วยทำให้เป็นอิสระจากการออกแบบเดิมหรือไชต์ของจริง

การช่วยทำให้การเติบโตทางการตลาดเพิ่มเร็วขึ้นและลดค่าใช้จ่ายลง เนื่องจากการแปลงไชต์ที่มีอยู่แล้วช่วยลดเวลาในการเขียนโปรแกรมน้อยกว่าการเริ่มต้นใหม่ตั้งแต่ต้น ข้อดีอีกอันหนึ่งคือผู้ที่ จะแปลงไชต์ ไม่จำเป็นต้องเป็นผู้ออกแบบเว็บที่เข้าใจเกี่ยวกับภาษามาร์กอัปต่าง ๆ พอสมควรนักจึงจะ สามารถโปรแกรมแอปพลิเคชันการแปลงภาษาได้ จึงค่อนข้างเป็นประโยชน์ต่อความต้องการของตลาด ในปัจจุบัน สำหรับผู้พัฒนาเว็บ

ข้อมูลที่ตัดแบ่งออกมาจากเพจเดิม สามารถถูกเก็บไว้โดยวิธีที่ไม่ขึ้นกับรูปแบบนั้นๆ ดังนั้น จึง สามารถประยุกต์รูปแบบต่าง ๆ นอกเหนือจาก WML เพื่อตอบสนองกับชนิดของไคลเอนต์อื่น ๆ ด้วย ด้วย เหตุผลนี้ XML จึงได้รับความนิยมมาก เพราะเป็นการทำให้อุปกรณ์ที่ไม่ใช่ PC ให้สามารถใช้งานบนเว็บ ได้มากขึ้น ซึ่งตัวแปลงภาษาต้องถูกกำหนดว่าจะส่งคอนเทนต์ไปที่อุปกรณ์ตัวไหนบ้าง เช่น PC , โทรศัพท์ ทีวี, PDA, Internet screen-phone, TV set-top box และ อุปกรณ์อินเทอร์เน็ตอื่น ๆ ที่มี

ผลลัพธ์ของภาษามาร์กอัปที่แปลง ไม่จำเป็นต้องเป็น WML เท่านั้น อาจจะเป็น XML, Compact HTML, HDML, XHTML, รูปแบบใดๆ ในอนาคต หรือกระทั่งแปลงเป็นตัว HTML เอง จากแนวความคิดนี้ทำให้ไม่ใคร่ชอบพ่เกิดแนวความคิดที่จะทำบราวเซอร์ร่วมกันระหว่าง HTML และ WML (dual WML/HTML browser) โดยทำการแปลงจาก HTML ไปเป็นทั้ง WML และ HTML ที่ลดขนาดลง (reduced HTML)

ข้อดีอีกประการหนึ่งคือความสามารถในการสร้างเว็บไชต์ขึ้นมาอีกอันหนึ่ง (WAP mirror site) ที่เสมือนเว็บ เพื่อให้ผู้ใช้เว็บสามารถแอกเซส เนื่องจากไม่ต้องการให้ผู้ใช้เข้าถึงข้อมูลจริงๆ เพื่อเป็นการ เพิ่มความปลอดภัยให้กับเซิร์ฟเวอร์

ประเด็นสุดท้ายคือในการเรียนรู้สิ่งต่าง ๆ มากมายที่เกิดขึ้นอย่างรวดเร็วเพื่อสร้างเว็บไชต์ให้ดีที่สุด และต้องพิจารณาว่าอิเลเมนต์ของไชต์ที่มีอยู่แล้วว่าอันไหนเป็นข้อมูลที่สำคัญ เช่น ถ้าตัวแปลงใช้ XML ก็จะจัดการเพียงแต่ในส่วนที่แปลงเป็น XML โดยไม่ต้องทิ้งไชต์นั้นทั้งหมดเพื่อให้ให้ทันต่อเทคโนโลยี ใหม่

6.2 ข้อเสีย

ปัญหาที่สำคัญอีกอย่างหนึ่งคือด้านประสิทธิภาพการทำงาน ซึ่งทำให้สามารถเสียเปรียบต่อคู่แข่งได้ ประกอบด้วย 2 ด้านคือ

- ความต้องการกำลังงานในการประมวลผลเพิ่มมากขึ้นในการแปลงข้อมูลจากฐานข้อมูลก่อนจะส่งกลับ ยิ่งถ้าเมื่อผู้ใช้ 10,000 คนเข้ามาที่เว็บไซต์คุณในเวลาเดียวกัน อาจทำให้เซิร์ฟเวอร์เกิดโอเวอร์โหลด และก็จะทำให้ต้องซื้อฮาร์ดแวร์เพิ่มเติมอีกเพื่อรองรับ
- เอกสาร HTML ต้นฉบับอาจจะมีขนาดใหญ่กว่า WML ที่เสมือนกันอยู่มาก ดังนั้น ไซค์ที่ถูกสร้างสำหรับ WML จะหลีกเลี่ยงการประมวลผลข้อมูลที่ไม่สำคัญหรือไม่จำเป็นต้องใช้

6.3 ข้อจำกัด

- ถ้าต้องการสร้างเอาต์พุตในรูปแบบใหม่ที่แตกต่างออกไป ไม่เพียงแต่จะต้องกำหนดข้อมูลเอาต์พุตในไฟล์ properties แต่ต้องเพิ่มโค้ดในการสร้าง DOM ที่เหมาะสม
- ไฟล์ extractor properties และไฟล์ XSL ถูกลิงก์กันไว้โดยแบ่งแยกไม่ได้ ซึ่งควรจะแบ่งเป็นไฟล์ข้อมูลกับไฟล์ properties ต่างหาก
- ไฟล์ properties ต้องถูกอ่านอย่างไม่จำเป็นในทุกๆ ครั้งที่มีการแปลง ซึ่งสามารถแก้ไขได้โดยการโหลดไฟล์ properties เพียงครั้งเดียวตอนระหว่างที่โคจรมเริ่มต้นทำงาน
- แทบไม่มีการจัดการกับข้อผิดพลาดที่เกิดขึ้น โดยเฉพาะไม่มีการตรวจสอบว่าขนาดเค็คของ WML มีขนาดใหญ่เกินกว่าอุปกรณ์หรือไม่
- ไม่สามารถแปลงไฟล์ HTML ที่มีโครงสร้างไม่เป็น well-form หรือโครงสร้างไม่ถูกต้องสมบูรณ์ได้

6.4 สรุป

จากการทดลองสร้างตัวแปลงภาษาจาก HTML เป็น WML แบบอัตโนมัติ โดยทำการแปลงในทุกๆ แท็ก พบว่าสามารถแปลงเว็บเพจส่วนใหญ่ได้ แต่ในบางกรณีโค้ด HTML ต้นแบบมีความซับซ้อนมาก (เช่น ใช้ทูลในการสร้าง) หรือไม่อยู่ในรูปแบบที่ถูกต้องตามไวยากรณ์ จะทำให้ไม่สามารถแปลงได้

6.5 แนวทางการพัฒนา

ข้อเสนอแนะในการพัฒนาเครื่องมือในการแปลงภาษาต่อไป

การเพิ่มคุณสมบัติทำงาน (features)

เพื่อให้สามารถแปลง HTML ที่มีโครงสร้างที่ไม่อยู่ในรูปแบบที่ถูกต้องได้ โดยอาจใช้ Tidy จาก W3C หรือ Jtidy จากการอิมพลีเมนต์จาวา ซึ่งเป็นเครื่องมือที่ช่วยเพิ่มแท็กที่ขาดหายไป, จัดการเกี่ยวกับแท็กที่คร่อมกันอยู่ และอื่นๆ ที่จะทำให้เป็น well-form HTML (HTML ที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างที่ถูกต้อง) ที่สุด แม้ว่า Jtidy จะไม่สามารถแก้ไขได้ทุกข้อผิดพลาด แต่สามารถแจ้งปัญหาและข้อผิดพลาดระหว่างการแปลงได้ ดังนั้นเราจึงสามารถเขียนหาวิธีจัดการกับพวกข้อผิดพลาดเหล่านี้ได้ โดยอาจใช้ Jtidy ร่วมกันตัวตัวพาร์เซอร์ HTML อื่น ๆ จะช่วยทำให้การเขียนโปรแกรมง่ายขึ้น

ในการแปลงเรื่องฟอรัม บางครั้งอาจเจอหน้าเพจที่มีฟอรัมมากมาย เช่น มีปุ่มให้เลือก 25 ปุ่ม กับการกรอกข้อมูลอีก 10 ช่วงนั้น อาจทำให้ผู้ใช้เว็บไม่มีความพึงพอใจเท่าไรกับการแสดงผลของโทรศัพท์ที่สามารถแสดงได้หน้าละ 5 บรรทัดเท่านั้น ดังนั้น จึงควรวางวิธีที่จะลดจำนวนฟิลด์ที่จะไปยังโทรศัพท์ที่ให้น้อยลงที่สุด ตัวอย่างเช่น ในการค้นหาไฟล์เครื่องบิน ซึ่งเป็นหน้า HTML ที่มี 10 อินพุตฟิลด์ แต่เว็บเซิร์ฟเวอร์จะประมวลผลเพียง 3 ฟิลด์ คือ 'From', 'To', 'Date' ดังในกรณีนี้ จึงควรแสดงเพียงแค่ 3 ฟิลด์เท่านั้น แต่อาจจะทำการใส่ค่าอินพุตให้กับฟิลด์อื่น ๆ เป็นดีฟอลต์ไว้ ซึ่งอาจจะทำที่ฝั่งเซิร์ฟเวอร์หรือ ใส่เป็นค่าในฟิลด์อินพุตเป็นดีฟอลต์ที่ด้านโทรศัพท์ ดังนั้นเมื่อผู้ใช้กดปุ่ม 'Find' ค่าตัวแปรในฟอรัมทั้งหมดจะถูกส่งไปยังเว็บเซิร์ฟเวอร์พีเจอรอื่น ๆ ที่เป็นประโยชน์คือ สนับสนุน SSL โดยทำการเข้ารหัส, เลือกเอาบางข้อความที่เหมาะสม, ถอดรหัส เพื่อรักษาความปลอดภัยให้กับผู้ใช้ และยังมีสิ่งที่ต้องพึงระวัง คือ ช่วงโหวของการความปลอดภัยในเว็บเทคโนโลยีปัจจุบัน ดังนั้นจึงต้องทำไตร่ตรองเป็นอย่างดีไม่เพียงแต่เพื่อป้องกันเซิร์ฟเวอร์ของตนเองเท่านั้น ยังต้องทำให้ผู้ยอมรับในบริการอีกด้วย

ไซค์จำนวนมากนิยมใช้คุกกี้ (cookie) เพื่อสนับสนุนการเชื่อมโยง (navigation) และการจัดการเซสชัน (session) เนื่องจากการสนับสนุนคุกกี้ยังไม่แพร่หลายนัก ดังนั้นจึงไม่เป็นเรื่องง่ายสำหรับอุปกรณ์เว็บในปัจจุบัน แต่อาจทำได้โดยอิมพลิเมนต์การทำงานของคุกกี้เอง เพื่อให้ได้ไซค์ที่ทำงานอย่างถูกต้อง

ความยืดหยุ่น

ความยืดหยุ่นในที่นี้หมายถึงความง่ายต่อการแปลงเว็บไซต์ใหม่ต่าง ๆ โดยที่ไม่ต้องยุ่งเกี่ยวกับการเปลี่ยนแปลงโค้ดจาวาเลย โดยจะเปลี่ยนแค่ไฟล์คุณสมบัติและไฟล์ XML เท่านั้น และในการแยก extract properties กับ XML ออกจากกันจะยิ่งช่วยให้อุปกรณ์ (tool) มีความยืดหยุ่นมากขึ้น ฟังก์ชันการทำงานสำหรับการจัดการหน้าเพจที่มีขนาดใหญ่ ในการแบ่งออกเป็นหลาย ๆ เด็ค โดยการทำงานต้องมีการติดต่อกับยูสเซอร์เอเจนต์ด้วย เนื่องจากในอุปกรณ์เว็บที่แตกต่างกันข้อจำกัดของขนาดเด็คจะมีขนาดที่แตกต่างกัน

ภาคผนวก ก

ก.1 แท็ก HTML

<A>

ประเภท มีแท็กปิด
หน้าที่ ในกรณีที่ใช้กับแอตทริบิวต์ HREF จะใช้สร้างลิงก์ หรือสร้างจุดเชื่อมต่อที่จะอ้างไปยังเว็บเพจอื่น แต่หากใช้กับแอตทริบิวต์ NAME จะกำหนดให้จุดนั้นเป็น anchor ที่ลิงก์อื่นมายังจุดนี้ได้

แอตทริบิวต์

ACCESSKEY

ใช้กำหนดคีย์ลัดซึ่งจะให้ผลเหมือนกับการคลิกที่ลิงก์นั้น เช่น หากกำหนดแอตทริบิวต์ ACCESSKEY มีค่าเป็น c ผู้ใช้สามารถกด [Alt] + [c] ก็จะไปยังเว็บเพจลิงก์อยู่กับจุดนั้นได้เลย

HREF ใช้กำหนดชื่อเว็บเพจหน้าที่ต้องการอ้างอิง โดยต้องใส่ให้ตรงตามรูปแบบของ URL

NAME กำหนดให้จุดนี้มีชื่อตามที่ระบุ โดยชื่อนี้ไม่ควรซ้ำกับชื่อแท็กหรือชื่อลิงก์ตัวอื่น

REL กำหนดความสัมพันธ์ของเว็บเพจหน้าปัจจุบันกับเว็บเพจหน้าที่เกี่ยวข้อง

REV ตรงกันข้ามกับ REL คือกำหนดความสัมพันธ์ของเว็บเพจที่เชื่อมไว้กับเว็บเพจหน้าปัจจุบัน

TABINDEX กำหนดลำดับของลิงก์ที่เคอร์เซอร์จะกระโดดไปเมื่อผู้ใช้กด [Tab] แต่ละครั้ง

TARGET ในกรณีที่หน้าจอมีการแบ่งออกเป็นหลายเฟรม จะเป็นการกำหนดเฟรมที่ต้องการให้แสดงเมื่อลิงก์นี้ถูกเรียกใช้

<ACRONYM>

ประเภท มีแท็กปิด

หน้าที่ ใช้ครอบข้อความที่ต้องการแสดงคีย์ที่จะปรากฏอยู่ในเนื้อหาของเว็บเพจเช่น World Wide Web <ACRONYM> (WWW) </ACRONYM>

แอตทริบิวต์ ไม่มี

<ADDRESS>

ประเภท มีแท็กปิด

หน้าที่ ส่วนมากใช้ครอบข้อความที่เป็นอีเมลแอดเดรส ซึ่งอาจเป็นแอดเดรสของผู้เขียนเว็บเพจ โดยเว็บเบราว์เซอร์จะแสดงข้อความชนิดนี้เป็นตัวเอียง

แอตทริบิวต์ ไม่มี

<APPLET>

ประเภท	มีแท็กปิด
หน้าที่	ใช้เรียกโปรแกรมเสริมที่เป็น Java applet เข้ามาทำงานในเว็บเบราว์เซอร์
แอตทริบิวต์	
ALIGN	ใช้กำหนดการจัดวางของข้อความที่ต่อท้าย Java applet ถ้ากำหนดค่าเป็น TOP, MIDDLE หรือ BOTTOM แล้ว ข้อความที่ต่อท้ายโปรแกรมนี้จะถูกแสดงไว้เสมอกับขอบด้านบน หรือตรงกลาง ด้านล่างของวินโดว์ของโปรแกรม applet นั้นตามลำดับ แล้วขึ้นบรรทัดใหม่ต่อท้ายทางด้านล่างของวินโดว์นั้น ถ้ากำหนดค่าเป็น LEFT หรือ RIGHT แล้ว ข้อความที่ต่อท้ายเฟรมนี้จะแสดงล้อมเฟรมทางด้านซ้าย และขวาตามลำดับ
ALT	กำหนดข้อความที่จะแสดงไว้แทนตัวโปรแกรม ในกรณีที่เว็บเบราว์เซอร์ไม่สามารถเรียกใช้โปรแกรมได้
ARCHIVE	ชี้ไปที่ที่มีโปรแกรมเพิ่มหรือไฟล์ข้อมูลเพิ่มสำหรับออบเจ็กต์ที่ต้องถูกดึงมาก่อนตัวโปรแกรม
CODE	กำหนดชื่อโปรแกรมที่เรียกใช้
CODEBASE	กำหนดที่อยู่ของไฟล์โปรแกรมตามแบบ URL
HEIGHT	กำหนดความสูงของวินโดว์ที่ใช้แสดงโปรแกรม กำหนดหน่วยเป็นพิกเซล
HSPACE	กำหนดพื้นที่ว่างบริเวณด้านซ้ายและขวาของวินโดว์ที่โปรแกรมทำงานอยู่
NAME	กำหนดชื่อให้โปรแกรม applet นี้ เพื่อให้ applet อื่นสามารถอ้างอิงได้
OBJECT	เป็นวิธีเรียกใช้ Java applet แบบหนึ่งที่สามารถใช้แทนแอตทริบิวต์ CODE แต่วิธีอ้างอิง Java applet ด้วยแอตทริบิวต์นี้ต้องกำหนดเป็นหมายเลข Serial No.
VSPACE	กำหนดพื้นที่ว่างบริเวณด้านบนและล่างของวินโดว์ที่โปรแกรมทำงานอยู่
WIDTH	กำหนดความกว้างของวินโดว์ที่โปรแกรมจะทำงาน กำหนดหน่วยเป็นพิกเซล

<AREA>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้กำหนดพิกัดของพื้นที่ย่อยในรูปอิมเมจแมพ โดยในแต่ละพื้นที่ย่อยเหล่านี้จะทำตัวเป็นลิงก์ไปยังเว็บเพจหรือไฟล์ที่ถูกกำหนดไว้คู่กับลิงก์นี้
แอตทริบิวต์	
ALT	ใช้กำหนดข้อความสำหรับแต่ละพื้นที่ ซึ่งจะถูกแสดงเมื่อเว็บเบราว์เซอร์ไม่สามารถแสดงรูปภาพได้ หรือผู้เล่นสั่งไม่ให้ทำการโหลดรูปมาแสดงบนเว็บเพจ
COORDS	ใช้กำหนดพิกัดของพื้นที่ย่อยในรูป โดยตำแหน่งพิกัดเหล่านี้จะมาเป็นชุดและแยกแต่ละชุดด้วยเครื่องหมาย , (comma)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HREF	ใช้กำหนดชื่อไฟล์หรือเว็บเพจที่เชื่อมโยงไว้กับพื้นที่ย่อนั้น ซึ่งเมื่อผู้เล่นเว็บคลิกในบริเวณพื้นที่ย่อนั้น ก็จะเรียกไฟล์หรือเว็บเพจนั้นมาแสดง ให้กำหนดชื่อไฟล์ตามแบบ URL
NOHREF	ใช้กำหนดให้พื้นที่ย่อนั้นไม่มีการลิงก์ไฟล์หรือเว็บเพจใด
SHAPE	ใช้กำหนดลักษณะของแต่ละพื้นที่ย่อย ซึ่งมีค่ากำหนดได้คือ rect (สี่เหลี่ยมผืนผ้า), circle (วงกลม), poly (รูปหลายเหลี่ยม), default (พื้นที่ที่เหลือ)
TABINDEX	ใช้กำหนดลำดับให้แต่ละพื้นที่ย่อย เมื่อผู้ใช้กด Tab เคอร์เซอร์จะเลื่อนไปยังพื้นที่ย่อยต่างๆตามลำดับที่กำหนดไว้
TARGET	ใช้กำหนดเฟรมที่ต้องการให้แสดงข้อมูล

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความเพื่อทำให้ข้อความนั้นเป็นตัวหนา
แอตทริบิวต์	ไม่มี

<BASE>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้กำหนด URL กลางหรือชื่อเฟรมกลางซึ่งค่า URL กลางนี้จะถูกเรียกใช้เมื่อมีแท็กใดๆที่ใช้แอตทริบิวต์ HREF และกำหนดค่า URL ในแอตทริบิวต์ HREF เป็นรูปแบบย่อ ทำให้ไม่ต้องกำหนดเต็มรูปทุกครั้งในแอตทริบิวต์นี้ ส่วนชื่อเฟรมกลางจะถูกใช้เมื่อมีลิงก์ที่ไม่ได้กำหนดค่าใดๆให้กับแอตทริบิวต์ TARGET โดยเว็บเบราว์เซอร์จะถือว่าให้ใช้ชื่อเฟรมกลางนี้แทนโดยอัตโนมัติ
แอตทริบิวต์	
HREF	ใช้กำหนดค่า URL กลาง
TARGET	ใช้กำหนดชื่อเฟรมกลาง

<BASEFONT>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้กำหนดสี ขนาด และคุณสมบัติพื้นฐานอื่นๆของข้อความที่เป็นข้อมูลภายในตัวเอกสาร ซึ่งจะมีผลกับข้อความในแท็กต่างๆที่ไม่ใช่แท็กที่กำหนดคุณสมบัติของข้อความ เช่น แท็ก เป็นต้น
แอตทริบิวต์	
COLOR	ใช้กำหนดสีของข้อความ โดยอาจจะกำหนดเป็นชื่อสี หรือค่า RGB ใดๆอย่างหนึ่งก็ได้ ซึ่งสีที่เป็นค่ามาตรฐานก็คือสีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FACE	ใช้กำหนดฟอนต์ โดยสามารถกำหนดได้ครั้งละหลายๆฟอนต์ แต่เว็บเบราว์เซอร์จะดูจากฟอนต์แรกที่กำหนด หากไม่สามารถแสดงฟอนต์แรกได้ ก็จะเลื่อนลำดับไปใช้ฟอนต์ที่ 2,3 แทนตามลำดับ
SIZE	ใช้กำหนดขนาดของข้อความ

<BIG>

ประเภท	มีแท็กปิด
หน้าที่	ใช้นั้นข้อความที่ต้องการให้มีขนาดใหญ่กว่าข้อความขนาดมาตรฐาน
แอตทริบิวต์	ไม่มี

<BLOCKQUOTE>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่เป็นคำพูด โดยแยกข้อความนั้นออกจากข้อความส่วนอื่นแล้วนำมาขึ้นย่อหน้าใหม่ ซึ่งต่างจากการใช้แท็ก <Q>
แอตทริบิวต์	

CITE ถ้าต้องการใส่ข้อความเพื่ออ้างถึงที่มาของข้อความนั้น สามารถทำได้โดยใช้แอตทริบิวต์ CITE และกำหนด URL ที่เป็นที่อยู่ของข้อมูลนั้น

<BODY>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อมูลและแท็กต่างๆซึ่งจะใช้แสดงเป็นเว็บเพจบนจอตนเอง
แอตทริบิวต์	สำหรับแท็ก <BODY> นี้มักจะใช้แอตทริบิวต์ด้านล่างนี้ในการกำหนดสีหรือพื้นหลัง ซึ่งสีที่เกี่ยวข้องกับแอตทริบิวต์เหล่านี้จะมี 16 สีมาตรฐาน คือ AQUA, BLACK, BLUE, FUSHIA, GLAY, LIME, MAROON, NAVY, OLIVE, PURPLE, RED, SILVER, TEAL, WHITE, และ YELLOW
ALINK	ใช้กำหนดสีของข้อความที่เป็นจุดลิงก์ขณะถูกเรียกใช้งาน

BACKGROUND

ใช้กำหนดตำแหน่งและชื่อไฟล์รูปที่จะนำมาใช้เป็นภาพพื้นของเว็บเพจ โดยกำหนดชื่อตามแบบ URL ในกรณีที่มีรูปนั้นมีขนาดเล็กกว่าเว็บเพจ ก็จะนำรูปเหล่านั้นมาเรียงต่อกันในลักษณะเหมือนกับการปูกระเบื้อง

BGCOLOR กำหนดสีพื้น อาจจะกำหนดเป็นชื่อสีหรือค่า RGB อย่างใดอย่างหนึ่งก็ได้

LINK กำหนดสีของข้อความที่เป็นจุดลิงก์ที่ยังไม่เคยถูกเรียกใช้มาก่อน

TEXT ใช้กำหนดสีของข้อความทั่วไปที่อยู่ภายใต้กรอบของแท็ก <BODY>

VLINK ใช้กำหนดสีของข้อความที่เป็นจุดลิงก์ที่เคยถูกเรียกใช้งานมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**
**

ประเภท ไม่มีแท็กปิด

หน้าที่ กำหนดให้บรรทัดขึ้นบรรทัดใหม่ เพราะโดยปกติแล้วบรรทัดจะแสดงข้อความเรียงจากซ้ายไปขวาและเรียงจากบนลงล่าง โดยไม่สนใจว่าข้อความภายในไฟล์ HTML นั้นมีการจัดย่อหน้า หรือขึ้นบรรทัดใหม่ไว้อย่างไร (คือ ไม่ตีความ Carriage Return ที่อยู่ในข้อความเลย) ดังนั้นถ้าต้องการกำหนดให้มีการขึ้นบรรทัดใหม่ ผู้สร้างเว็บเพจจะต้องใส่เองโดยใช้แท็ก

แอตทริบิวต์

CLEAR เป็นแอตทริบิวต์ที่ใช้เพื่อระบุตำแหน่งของข้อความที่อยู่ถัดจากแท็ก
 เช่น ภายในเว็บเพจหน้าหนึ่งมีรูปที่อยู่ชิดด้านซ้าย ข้อความที่ 1 แท็ก <BR CLEAR = "left"> และตามด้วยข้อความที่ 2 ต่อท้าย เมื่ออยู่บนเว็บเพจจะปรากฏข้อความที่ 2 ขึ้นบรรทัดใหม่ที่ต่อท้ายรูปที่อยู่ด้านซ้าย โดยค่าที่สามารถกำหนดได้คือ left, right, all และ none

<BUTTON>

ประเภท มีแท็กปิด

หน้าที่ ใช้สร้างปุ่มรับคำสั่งบนฟอร์ม ปุ่มรับคำสั่งที่สร้างด้วยแท็กนี้จะได้ผลใกล้เคียงกับแท็ก <INPUT> ต่างกันตรงรูปปุ่มที่ได้ คือถ้าเพิ่มแท็ก เข้าไปภายใต้กรอบของแท็ก <BUTTON> ก็สามารถใช้รูปนั้นแทนรูปปุ่มได้

แอตทริบิวต์

DISABLED ห้ามไม่ให้ผู้ใช้คลิกที่ปุ่ม และบรรทัดจะแสดงปุ่มเป็นสีจางกว่าปุ่มปกติ

NAME กำหนดชื่อให้กับปุ่ม

TYPE กำหนดชนิดของปุ่ม มีค่าเป็นไปได้คือ RESET หรือ SUBMIT ซึ่งจะมีผลเหมือนปุ่ม RESET และ SUBMIT ที่สร้างจากแท็ก <INPUT>

VALUE กำหนดข้อความที่จะแสดงบนปุ่ม

<CAPTION>

ประเภท มีแท็กปิด

หน้าที่ ใช้ครอบข้อความสั้นๆที่เป็นชื่อตารางหรือข้อความกำกับตาราง โดยแท็กนี้ต้องอยู่ระหว่างแท็ก <TABLE> และ </TABLE> อีกทีหนึ่ง

แอตทริบิวต์

ALIGN ใช้กำหนดตำแหน่งของชื่อตาราง เช่น ALIGN=top ให้แสดงชื่อตารางไว้ที่ด้านบนเหนือตาราง ค่าที่เป็นไปได้คือ top, bottom, left และ right

<CENTER>

ประเภท	มีแท็กปิด
หน้าที่	ครอบข้อความหรือรูปภาพที่ต้องการให้อยู่ตรงกลางหน้าเว็บเพจ
แอตทริบิวต์	ไม่มี

<CITE>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการสื่อว่าเป็นคำที่ยืมมาใช้ หรือเอกสารที่ใช้อ้างอิง มักจะแสดงเป็นตัวเอียง
แอตทริบิวต์	ไม่มี

<CODE>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่เป็นโปรแกรม, Source code หรือสคริปต์ โดยเว็บเบราว์เซอร์จะแสดงข้อความด้วยฟอนต์ที่มีขนาดตัวอักษรเท่ากัน
แอตทริบิวต์	ไม่มี

<COL>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้กำหนดความกว้างและลักษณะอื่นให้กับคอลัมน์หรือบางคอลัมน์ที่อยู่ในกลุ่มตารางแท็ก <COL> เป็นตัวแทนของกลุ่มย่อยของเซลล์ที่อยู่ในกลุ่มใหญ่ที่กำหนดโดยแท็ก <COLGROUP> โดยต้องกำหนดแท็ก <COL> ไว้ได้แท็ก <COLGROUP> อีกที่หนึ่ง
แอตทริบิวต์	
ALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนที่เป็นข้อมูลในกลุ่มว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านข้างอย่างไร มีค่าเป็นไปได้คือ left, right, center และ justify
SPAN	กำหนดจำนวนคอลัมน์ในแต่ละแถวที่จะให้อยู่ในกลุ่ม
VALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนที่เป็นข้อมูลในกลุ่มว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านบนและล่างอย่างไร มีค่าเป็นไปได้คือ top, middle, bottom และ baseline
WIDTH	กำหนดความกว้างของเซลล์ที่อยู่ในกลุ่ม โดยกำหนดเป็นหน่วยพิกเซล

<COLGROUP>

ประเภท	มีแท็กปิด
--------	-----------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่	ใช้จัดกลุ่มของคอลัมน์ที่ต้องการให้มีขนาดและลักษณะเหมือนกันเข้าด้วยกันซึ่งจะทำให้สะดวกกว่าวิธีกำหนดขนาดและลักษณะของแต่ละคอลัมน์แยกกันและปิดระหว่างแท็ก <COLGROUP> และ </COLGROUP> จะไม่มีส่วนที่เป็นเนื้อหาหรือแท็กอื่นๆ นอกจากแท็ก <COL> เท่านั้น โดยแท็ก <COL> นี้จะใช้กำหนดลักษณะให้กับกลุ่มคอลัมน์ย่อยที่อยู่ใต้กลุ่มคอลัมน์ใหญ่ที่กำหนดโดยแท็ก <COLGROUP> อีกทีหนึ่ง
แอตทริบิวต์	
ALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนที่เป็นข้อมูลในกลุ่มว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านข้างอย่างไร มีค่าเป็นไปได้คือ left, right, center และ justify
SPAN	กำหนดจำนวนคอลัมน์ในแต่ละแถวที่จะให้อยู่ในกลุ่ม
VALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนที่เป็นข้อมูลในกลุ่มว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านบนและล่างอย่างไร มีค่าเป็นไปได้คือ top, middle, bottom และ baseline
WIDTH	กำหนดความกว้างของเซลล์ที่อยู่ในกลุ่ม โดยกำหนดเป็นหน่วยพิกเซล
	
ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการแสดงให้เห็นว่าถูกลบออก ซึ่งเว็บเบราว์เซอร์จะแสดงข้อความนั้นพร้อมทั้งมีเส้นขีดฆ่าพาดอยู่ตรงกลาง ทั้งนี้เพื่อให้เห็นว่าเว็บเพจนั้นมีการแก้ไขตรงตำแหน่งไหนบ้าง เพราะบางครั้งผู้ที่แก้ไขเว็บเพจกับผู้สร้างเป็นคนละคนกัน และต้องการแสดงให้เห็นตำแหน่งที่มีการแก้ไข
แอตทริบิวต์	ไม่มี
<DFN>	
ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการเน้นว่ามีความหมายพิเศษ และมีคำอธิบาย
แอตทริบิวต์	ไม่มี
<DIR>	
ประเภท	มีแท็กปิด
หน้าที่	ใช้แสดงรายการย่อยแบบมีจุดหรือสัญลักษณ์กำกับหน้าแต่ละรายการจะมีตัวอักษรได้ไม่เกิน 20 ตัว ซึ่งแท็กนี้แต่เดิมมีวัตถุประสงค์ให้ใช้กับข้อความที่ไม่กว้างเกินไปนัก เช่น ดัชนี หรือหมายเลขโทรศัพท์
แอตทริบิวต์	
COMPACT	ควบคุมรายการย่อยในบรรทัดต่างๆให้อยู่ชิดกันมากขึ้น โดยลดช่องว่างระหว่างบรรทัดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<DIV>

ประเภท	มีแท็กปิด
หน้าที่	ใช้กำหนดกลุ่มข้อความ หรือจัดแบ่งข้อความออกเป็นส่วน เพื่อช่วยให้สะดวกต่อการจัดวางข้อความในกลุ่มหรือส่วนนั้น
แอตทริบิวต์	
ALIGN	ใช้จัดวางตำแหน่งของข้อความว่าจะให้ชิดขอบด้านใด มีค่าที่เป็นไปได้คือ left, right, center และ justify

<DL>

ประเภท	มีแท็กปิด
หน้าที่	ใช้สร้างข้อมูลรายการแบบมีนิยาม โดยโครงสร้างรายการแบบนี้จะต่างจากข้อมูลรายการแบบอื่น คือ ในแต่ละรายการย่อยจะประกอบด้วย 2 ส่วนคือ รายการย่อยและรายละเอียดอธิบายในแต่ละรายการย่อย
แอตทริบิวต์	
COMPACT	ควบคุมรายการย่อยในบรรทัดต่างๆให้อยู่ชิดกันมากขึ้น โดยลดช่องว่างระหว่างบรรทัดลง

<DT>

ประเภท	มีแท็กปิด
หน้าที่	ใช้กำหนดแต่ละรายการย่อยที่อยู่ภายในรายการแบบมีนิยาม (แท็ก<DL>)
แอตทริบิวต์	ไม่มี

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการเน้นเป็นพิเศษ โดยเว็บเบราว์เซอร์จะแสดงเป็นตัวเอียง
แอตทริบิวต์	ไม่มี

<FIELDSET>

ประเภท	มีแท็กปิด
หน้าที่	ใช้จัดกลุ่มฟิลด์บางฟิลด์ไว้ด้วยกัน
แอตทริบิวต์	ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภท	มีแท็กปิด
หน้าที่	ใช้กำหนดสี ขนาด หรือฟอนต์ใหม่ให้กับข้อความที่ถูกแท็กนี้ครอบไว้ เพื่อให้ต่างจากคุณสมบัติพื้นฐานที่ถูกกำหนดโดยแท็ก <BASEFONT>
แอตทริบิวต์	จะเหมือนกับแอตทริบิวต์ที่ใช้กับแท็ก <BASEFONT> แต่สามารถกำหนดค่าให้แอตทริบิวต์ SIZE เป็นค่าสัมพัทธ์ได้ เช่น +1 คือใหญ่กว่าค่าที่กำหนดไว้ด้วย <BASEFONT> หนึ่งระดับ และ -2 คือเล็กกว่าค่าที่กำหนดไว้ด้วย <BASEFONT> สองระดับ เป็นต้น

<FORM>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อมูลและแท็กที่ใช้สร้างฟิลด์รับข้อมูลและปุ่มต่างๆที่ประกอบกันเป็นฟอร์ม นอกจากนี้แท็ก <FORM> ยังเป็นตัวกำหนดว่าจะส่งข้อมูลให้เว็บเซิร์ฟเวอร์อย่างไร และเรียกใช้โปรแกรมตัวใดทำงาน
แอตทริบิวต์	
ACCEPT-CHARSET	กำหนดชุดของตัวอักษรที่ฟอร์มสามารถรับได้
ACTION	กำหนดชื่อไฟล์หรือโปรแกรมที่ทำหน้าที่ประมวลผลข้อมูลที่ผู้เล่นเว็บป้อนผ่านฟอร์มเข้ามา โดยโปรแกรมหรือไฟล์นี้อยู่ที่ฝั่งเซิร์ฟเวอร์ และต้องกำหนดชื่อไฟล์ตามแบบ URL
ENCTYPE	กำหนดชนิดของไฟล์ที่จะถูกรับและส่งในแบบฟอร์มตามแบบ MIME
METHOD	กำหนดวิธีส่งข้อมูลให้เว็บเซิร์ฟเวอร์ มีค่าเป็นไปได้ 2 แบบคือ POST หรือ GET
TARGET	กำหนดชื่อเฟรมที่ต้องการให้แสดงผลพีชที่ได้จากการประมวลผลของเว็บเซิร์ฟเวอร์ ตามข้อมูลที่ฟอร์มส่งไปให้

<FRAME>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้กำหนดข้อมูลที่จะถูกแสดงไว้ในเฟรม
แอตทริบิวต์	
FRAMEBORDER	กำหนดว่าต้องการแสดงเส้นกรอบของเฟรมหรือไม่ มีค่าเป็นไปได้คือ 0 หมายถึงไม่ต้องแสดงเส้นกรอบ ส่วน 1 หมายถึงให้แสดงเส้นกรอบ
MARGINHEIGHT	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดระยะห่างระหว่างเส้นกรอบด้านบนของเฟรมกับตัวข้อมูล กำหนดหน่วยเป็นพิกเซล

MARGINWIDTH

กำหนดระยะห่างระหว่างเส้นกรอบด้านซ้ายของเฟรมกับตัวข้อมูล กำหนดหน่วยเป็นพิกเซล

NAME กำหนดชื่อให้กับเฟรม เพื่อให้แท็กอื่นๆสามารถอ้างอิงได้ เช่น แท็ก <A>, <FORM> และ <AREA> เป็นต้น

NORESIZE กำหนดว่าไม่ยอมให้ผู้ใช้ย่อหรือขยายขนาดเฟรม โดยปกติผู้ใช้สามารถใช้เมาส์ย่อหรือขยายขนาดเฟรมได้

SCROLLING

กำหนดว่าต้องการแสดงแถบเลื่อนด้านข้างของเฟรมหรือไม่ มีค่าเป็นไปได้คือ

yes หมายถึง แสดงแถบเลื่อนเสมอ

no หมายถึง ไม่ต้องแสดงแถบเลื่อนเลย

auto หมายถึง แสดงแถบเลื่อนในกรณีที่มีข้อมูลล้นเฟรมเท่านั้น

SRC กำหนดไฟล์ HTML หรือ ไฟล์ข้อมูลอื่นที่ใช้แสดงในเฟรม เป็นแอตทริบิวต์ที่จำเป็นต้องมีเสมอ

<FRAMESET>

ประเภท

มีแท็กปิด

หน้าที่

ใช้แบ่งพื้นที่ในเว็บเพจออกเป็นส่วนย่อยๆ

แอตทริบิวต์

ROWS

ใช้กำหนดว่าจะให้มีการแบ่งพื้นที่ทั้งหมดออกเป็นกี่แถว และในแต่ละแถวมีความสูงเท่าใด

COLS

ใช้กำหนดว่าจะให้มีการแบ่งพื้นที่ทั้งหมดกี่คอลัมน์ และในแต่ละคอลัมน์มีความกว้างเท่าใด

<H1> - <H6>

ประเภท

มีแท็กปิด

หน้าที่

ใช้ครอบข้อความที่เป็นหัวข้อในระดับต่างกัน ตัวอย่างเช่น <H1> ใช้ครอบหัวข้อที่มีความสำคัญมากที่สุด และ <H2> ใช้ครอบหัวข้อที่มีความสำคัญรองลงมา โดยส่วนมากแล้วบรรทัดแรกจะจัดข้อความที่เป็นหัวข้อเหล่านี้ให้มีขนาดตัวอักษรใหญ่และหนากว่าปกติ และก็จะเพิ่มที่ว่างเหนือและใต้บรรทัดนั้นให้มากขึ้นด้วย

แอตทริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALIGN จัดวางตำแหน่งข้อความว่าจะให้อยู่ชิดขอบด้านใด มีค่าเป็นไปได้คือ left (ให้ชิดขอบซ้าย), right (ให้ชิดขอบขวา), center (อยู่ตรงกลางระหว่างขอบซ้ายและขวา) และ justify (ให้ข้อความกระจายตามความกว้างของขอบทั้งสองด้าน) ถ้าไม่กำหนดไว้จะมีค่าเป็น left

<HEAD>

ประเภท มีแท็กปิด
หน้าที่ ใช้ครอบข้อความมแท็ก และแอตทริบิวต์ที่อยู่ในบริเวณส่วนหัวเอกสาร
แอตทริบิวต์ ไม่มี

<HR>

ประเภท ไม่มีแท็กปิด
หน้าที่ สร้างเส้นคั่นในแนวนอน เพื่อแบ่งข้อมูลออกเป็นส่วนๆอย่างชัดเจน
แอตทริบิวต์

ALIGN กำหนดตำแหน่งเส้นคั่นว่าจะให้ชิดขอบด้านใด ซึ่งจะมีผลก็ต่อเมื่อความยาวของเส้นคั่นนั้นสั้นกว่าความกว้างของหน้าจอ มีค่าเป็นไปได้คือ left , right และ center

NOSHADA กำหนดเส้นคั่นเป็นแบบทึบ

SIZE กำหนดความหนาของเส้นคั่น โดยกำหนดเป็นหน่วยพิกเซล

WIDTH กำหนดความยาวของเส้นคั่น โดยกำหนดหน่วยเป็นพิกเซลหรือเป็นเปอร์เซ็นต์ของขนาดหน้าจอ

<HTML>

ประเภท มีแท็กปิด
หน้าที่ ใช้ครอบข้อมูลที่นำเสนอด้วยภาษา HTML
แอตทริบิวต์ ไม่มี

<I>

ประเภท มีแท็กปิด
หน้าที่ ใช้ครอบข้อความที่เน้นเป็นตัวเอียง
แอตทริบิวต์ ไม่มี

<IFRAME>

ประเภท มีแท็กปิด
หน้าที่ ใช้สร้างเฟรมแบบที่ลอยได้ (floating frame)คือเฟรมนี้ไม่จำเป็นต้องมีการจัดแบ่งพื้นที่ไว้ก่อนล่วงหน้าเหมือนกับเฟรมที่สร้างด้วยแท็ก <FRAME> แต่จะมีลักษณะเหมือนกันกับรูปภาพบนเว็บเพจ คือสามารถจะสร้างและวางไว้ตรงไหนก็ได้ โดยจะแสดงเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรอบสี่เหลี่ยม และมีแถบเลื่อนด้านข้างถ้าข้อมูลในเฟรมมีมากกว่าที่จะแสดงในเฟรมได้หมดทีเดียว

แอตทริบิวต์

ALIGN ใช้กำหนดการจัดวางของข้อความที่ต่อท้ายเฟรม
ถ้ากำหนดค่าเป็น TOP, MIDDLE หรือ BOTTOM แล้ว ข้อความที่ต่อท้ายเฟรมนี้จะถูกแสดงไว้เสมอกับขอบด้านบน หรือตรงกลาง ด้านล่างของเฟรมก็ตามลำดับ และเมื่อข้อความยาวจนล้นจอ ข้อความที่เหลือก็จะถูกปิดไปขึ้นบรรทัดใหม่ที่ด้านล่างใต้เฟรม
ถ้ากำหนดค่าเป็น LEFT หรือ RIGHT แล้ว ข้อความที่ต่อท้ายเฟรมนี้จะแสดงล้อมเฟรมทางด้านซ้าย และขวาตามลำดับ

FRAMEBORDER

กำหนดว่าต้องการแสดงเส้นกรอบของเฟรมหรือไม่ มีค่าเป็นไปได้อีกคือ 0 หมายถึงไม่ตั้ง' แสดงเส้นกรอบ ส่วน 1 หมายถึงให้แสดงเส้นกรอบ

HEIGHT กำหนดความสูงของเฟรม กำหนดเป็นหน่วยพิกเซล

MARGINHEIGHT

กำหนดระยะห่างระหว่างเส้นกรอบด้านบนของเฟรมกับข้อมูล

MARGINWIDTH

กำหนดระยะห่างระหว่างเส้นกรอบด้านซ้ายของเฟรมกับตัวข้อมูล

NAME กำหนดชื่อให้เฟรม เพื่อให้แท็กอื่นๆสามารถอ้างอิงได้ เช่น แท็ก <A>, <FORM> และ <AREA> เป็นต้น

SCROLLING

กำหนดว่าต้องการแสดงแถบเลื่อนหรือไม่ มีค่าเป็นไปได้อีกคือ

yes หมายถึง แสดงแถบเลื่อนเสมอ

no หมายถึง ไม่แสดงแถบเลื่อน

auto หมายถึง แสดงแถบเลื่อนในกรณีที่มีข้อมูลล้นเฟรม

SRC กำหนดชื่อไฟล์ข้อมูลอื่นที่ใช้แสดงในเฟรม โดยกำหนดตามรูปแบบ URL

WIDTH กำหนดความกว้างของเฟรม กำหนดเป็นหน่วยพิกเซล

ประเภท ไม่มีแท็กปิด

หน้าที่ ใช้วางรูปไว้ในเว็บเพจ โดยไฟล์ภาพอาจเป็น .GIF หรือ .JPEG ก็ได้ ในกรณีที่ต้องการนำรูปมาใช้เป็นอิมเมจแมพ ต้องกำหนดแอตทริบิวต์ ISMAP หรือ USEMAP อย่างใดอย่างหนึ่งด้วยเสมอ

แอตทริบิวต์

SRC กำหนดตำแหน่งและชื่อไฟล์รูป โดยกำหนดชื่อตาม URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WIDTH และ HEIGHT

กำหนดความกว้างและความสูงของรูปโดยมีหน่วยเป็นพิกเซล ซึ่งการกำหนดแบบนี้ทำให้เบราว์เซอร์สามารถกำหนดพื้นที่ให้ภาพนั้น พร้อมกับจัดวางข้อความอื่นบนเว็บเพจไปได้เลย โดยไฟล์รูปนั้นก็ยังคงถูกทยอยดาวน์โหลดเข้ามา

ALT กำหนดข้อความที่จะแสดงบริเวณรูปหรือโปรแกรม plug-in ต่างๆ ซึ่งข้อความเหล่านี้จะปรากฏเมื่อเบราว์เซอร์ยังโหลดรูปหรือโปรแกรมไม่เสร็จ หรือหากเว็บเบราว์เซอร์ที่ใช้ไม่สามารถแสดงภาพกราฟิกได้ เช่น Lynx ก็จะมาแสดงข้อความนี้แทน

BORDER กำหนดความหนาของกรอบรูป แต่โดยปกติรูปจะไม่ใส่กรอบไว้ให้ ยกเว้นแต่รูปที่เป็นไฮเปอร์ลิงก์จะถูกใส่กรอบที่มีสีไว้ด้วย ซึ่งสามารถจะยกเลิกได้โดยการกำหนดให้ BORDER=0 แทน

ALIGN กำหนดตำแหน่งที่ใช้แสดงข้อความที่ตามหลังรูป โดยหากกำหนดเป็น top, middle หรือ bottom ข้อความจะถูกวางไว้เสมอกับด้านบนของรูป, อยู่กลาง ระหว่างความสูงของรูป และอยู่เสมอกับด้านล่างของรูปตามลำดับ โดยข้อความในบรรทัดถัดไปจะแสดงอยู่ได้รูป แต่หากกำหนดเป็นแบบ left หรือ right ข้อความทั้งหมดจะอยู่ล้อมรูปทางด้านซ้ายและทางขวาตามลำดับ

HSPACE และ VSPACE

กำหนดระยะห่างระหว่างข้อมูลกับรูป หรือเป็นการกำหนดพื้นที่ว่างรอบรูปนั่นเองโดยมีหน่วยเป็นพิกเซล สำหรับ HSPACE จะใช้กำหนดพื้นที่ว่างซึ่งอยู่ทางด้านซ้ายและขวาของรูป ส่วน VSPACE ใช้กำหนดพื้นที่ว่างเหนือและใต้รูป

ISMAP กำหนดให้รูปที่กำหนดด้วยแท็ก เป็นรูปที่จะใช้เป็นอิมเมจแมพแบบแปลความหมายโดยเว็บเบราว์เซอร์

USEMAP กำหนดให้รูปที่ถูกนำมาใช้เป็นอิมเมจแมพแบบแปลความหมายโดยเว็บเบราว์เซอร์

<INPUT>

ประเภท ไม่มีแท็กปิด

หน้าที่ ใช้สร้างฟิลด์รับข้อมูลแบบต่างๆตามค่าที่กำหนดให้กับแอตทริบิวต์ TYPE

แอตทริบิวต์

NAME ใช้กำหนดชื่อของฟิลด์รับข้อมูล ฟิลด์รับข้อมูลทุกตัวควรมีชื่อ ทั้งนี้เพราะโปรแกรมประมวลผลทางฝั่งเซิร์ฟเวอร์จะอ้างถึงข้อมูลภายในฟิลด์ด้วยชื่อนี้ได้

VALUE เป็นแอตทริบิวต์ที่ใช้กำหนดค่าเริ่มต้นให้ฟิลด์รับข้อมูล หรือกำหนดข้อความที่จะแสดงไว้บนปุ่มรับคำสั่ง

TYPE ใช้กำหนดว่าชนิดของฟิลด์ มีค่าเป็นไปได้อย่างไร

text	ฟิลด์รับข้อความแบบบรรทัดเดียว โดยมีแอตทริบิวต์อื่นที่ต้องกำหนดร่วมด้วย คือ NAME, VALUE, LENGTH, MAXLENGTH, DISABLE และ READONLY
password	ฟิลด์รับข้อความแบบบรรทัดเดียวคล้ายกับแบบ TEXT แต่ใช้กับการป้อนรหัสผ่านโดยเฉพาะ โดยรหัสผ่านหรือความที่ถูกป้อนเข้าไปจะแสดงเป็นเครื่องหมาย * แทนตัวอักษรแต่ละตัวเสมอ
hidden	ฟิลด์นี้จะไม่ถูกแสดงบนเว็บเพจ แต่ข้อมูลที่เกี่ยวข้องอยู่ในฟิลด์จะถูกส่งไปให้เว็บเซิร์ฟเวอร์ สำหรับฟิลด์แบบนี้ต้องมีการกำหนดแอตทริบิวต์ NAME และ VALUE ร่วมด้วย
checkbox	ฟิลด์แบบนี้ผู้ใช้ต้องคลิกเพื่อเลือกหรือไม่เลือกรายการนั้น โดยต้องกำหนดแอตทริบิวต์ NAME เพื่อกำหนดชื่อให้กับฟิลด์ และแอตทริบิวต์ VALUE เพื่อกำหนดค่าที่ส่งให้กับเว็บเซิร์ฟเวอร์หากฟิลด์นั้นถูกคลิกเลือก กรณีที่ต้องการให้มีค่าเริ่มต้นของฟิลด์ใดเป็นแบบถูกเลือกไว้ ก็ให้ใส่แอตทริบิวต์ CHECKED กำกับไปด้วย
radio	ฟิลด์แบบนี้จะมีตัวเลือกหลายตัว แต่ผู้เล่นเว็บจะเลือกได้เพียงตัวเดียวเท่านั้น โดยจะต้องใส่แท็ก <INPUT> เท่ากับจำนวนตัวเลือกที่มี และกำหนดแอตทริบิวต์ NAME เพื่อกำหนดชื่อตัวแปรในแท็กเหล่านั้นให้เหมือนกัน แต่ใส่แอตทริบิวต์ VALUE พร้อมทั้งค่าในแต่ละแท็กไม่เหมือนกัน ซึ่งค่านี้จะเป็นค่าที่ส่งให้กับเว็บเซิร์ฟเวอร์เมื่อรายการนั้นถูกเลือก หากต้องการให้ตัวเลือกใดถูกเลือกไว้เป็นค่าเริ่มต้น ก็สามารถทำได้โดยใส่แอตทริบิวต์ CHECKED กำกับไว้ที่ค่านั้น
file	ฟิลด์ที่ใช้ใส่ชื่อไฟล์เพื่อส่ง (upload) ไปยังเว็บเซิร์ฟเวอร์พร้อมกับข้อมูลอื่นที่ผู้เล่นเว็บกรอกผ่านฟอร์ม โดยสามารถใส่แอตทริบิวต์ NAME เพื่อกำหนดชื่อให้กับฟิลด์ VALUE เพื่อกำหนดค่าเริ่มต้นให้กับฟิลด์ ACCEPT เพื่อกำหนดประเภทของไฟล์ และ DISABLED เพื่อกำหนดให้ฟิลด์นี้ไม่รับข้อมูลใดๆ
image	ใช้กำหนดรูปแบบแทนปุ่มรับคำสั่ง โดยใส่แอตทริบิวต์ SRC เพื่อกำหนดตำแหน่งและชื่อไฟล์ที่เกี่ยวข้อง ALT เพื่อระบุชื่อหรือข้อความที่ต้องการให้แสดงแทนในกรณีที่ไม่สามารถหารูปนั้นพบ ALIGN เพื่อกำหนดตำแหน่งที่จะให้รูปนั้นปรากฏและ DISABLED เพื่อกำหนดให้รูปนั้นไม่รับข้อมูลใดๆ
button	ใช้สร้างปุ่มรับคำสั่ง โดยใส่แอตทริบิวต์ VALUE เพื่อกำหนดข้อความที่จะให้ปรากฏอยู่บนปุ่ม แอตทริบิวต์ onclick เพื่อกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	สคริปต์หรือ โปรแกรมบนฝั่งเซิร์ฟเวอร์ที่จะถูกเรียกให้ทำงานเมื่อปุ่มนี้โดนคลิก และ DISABLED เพื่อกำหนดให้ปุ่มนี้ไม่ทำงาน
submit	ใช้สร้างปุ่มรับคำสั่ง SUBMIT ซึ่งเมื่อผู้ใช้คลิกที่ปุ่มนี้ บราวเซอร์จะส่งข้อมูลทั้งหมดบนฟอร์มไปให้เว็บเซิร์ฟเวอร์
reset	ใช้สร้างปุ่มรับคำสั่ง RESET ซึ่งเมื่อผู้ใช้คลิกที่ปุ่มนี้ บราวเซอร์จะล้างข้อมูลทั้งหมดที่อยู่ในฟอร์ม ให้เหมือนตอนเริ่มต้น
DISABLED	ใช้ระงับการทำงานของฟิลด์ทุกชนิด โดยเมื่อเพิ่มแอตทริบิวต์ DISABLED บราวเซอร์จะยังคงแสดงฟิลด์เหล่านั้น แต่จะไม่ยอมให้ผู้ใช้ป้อนข้อมูลหรือเปลี่ยนแปลงตัวเลือกในฟิลด์เหล่านั้นได้
LENGTH	ใช้กำหนดความยาวเป็นจำนวนตัวอักษรให้กับฟิลด์รับข้อมูลที่จะแสดงบนหน้าจอ
MAXLENGTH	ใช้กำหนดความยาวสูงสุดเป็นจำนวนตัวอักษรที่ฟิลด์นั้นจะรับได้
READONLY	กำหนดให้ฟิลด์นี้แสดงข้อมูลเพียงอย่างเดียว ห้ามผู้ใช้ป้อนหรือแก้ไขข้อมูล
<INS>	
ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการแสดงให้เห็นว่าถูกเพิ่มเติมเข้าไป มีความหมายตรงกันข้ามกับแท็ก โดยเว็บเบราว์เซอร์แสดงข้อความพร้อมทั้งมีเส้นขีดอยู่ใต้ข้อความนั้น
แอตทริบิวต์	ไม่มี
<ISINDEX>	
ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้สร้างฟิลด์รับข้อความ และเรียกใช้โปรแกรมค้นหาข้อมูลตามข้อความที่ผู้ใช้กรอกเข้ามา ปัจจุบันไม่นิยมใช้กันแล้ว เพราะสามารถใช้ฟอร์มทำงานแทนได้
แอตทริบิวต์	
PROMPT	ใช้กำหนดข้อความที่จะปรากฏบนเว็บเพจ พร้อมทั้งฟิลด์รับข้อความที่จะถูกป้อนเข้ามาโดยนักเล่นเว็บ
<KBD>	
ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการแสดงให้เห็นว่าเป็นสิ่งที่ป้อนเข้ามาผ่านทางคีย์บอร์ด โดยเบราว์เซอร์จะแสดงข้อความในแท็กนี้ด้วยฟอนต์ที่มีขนาดตัวอักษรขนาดเท่ากันทุกตัว
แอตทริบิวต์	ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<LABEL>

ประเภท มีแท็กปิด
 หน้าที่ กำหนดข้อความที่แสดงกำกับฟิลด์ โดยส่วนใหญ่จะใช้คู่กับฟิลด์ที่เป็นแบบ text เพื่อให้ผู้ใช้ทราบว่าข้อมูลที่ใส่กรอกนั้นเป็นข้อมูลที่เกี่ยวข้องกับอะไร เช่น ชื่อหรือที่อยู่ เป็นต้น ถ้าผู้ใช้คลิกที่ข้อความนี้เบราว์เซอร์จะเลื่อนเคอร์เซอร์ไปยังฟิลด์ที่คู่กันทันที

แอตทริบิวต์

ACCESSKEY

กำหนดปุ่มในแป้นพิมพ์เพื่อให้เคอร์เซอร์เลื่อนไปยังฟิลด์หรือปุ่มภายในฟอร์มที่ต้องการ เช่น หากกำหนดว่า ACCESSKEY = "A" ผู้ใช้สามารถกด Alt + A หรือ Ctrl + A เพื่อให้เคอร์เซอร์เลื่อนไปยังฟิลด์รับข้อมูลหรือปุ่มนั้นทันที

DISABLED ห้ามไม่ให้ผู้ใช้คลิกข้อความ LABEL นี้

FOR กำหนดค่าให้เป็น ID ของฟิลด์ที่ต้องการให้คู่กับ LABEL นี้

<LEGEND>

ประเภท มีแท็กปิด
 หน้าที่ สร้างข้อความกำกับกลุ่มของฟิลด์ที่สร้างโดยแท็ก <FIELDSET>

แอตทริบิวต์

ACCESSKEY

กำหนดปุ่มในแป้นพิมพ์เพื่อให้เคอร์เซอร์เลื่อนไปยังกลุ่มของฟิลด์ที่ต้องการ เช่น หากกำหนดว่า ACCESSKEY = "A" ผู้ใช้สามารถกด Alt + A หรือ Ctrl + A เพื่อให้เคอร์เซอร์เลื่อนไปยังกลุ่มของฟิลด์นั้น

ALIGN ใช้กำหนดตำแหน่งของข้อความ โดยมีค่าที่กำหนดได้คือ left, right, top และ bottom

ประเภท มีแท็กปิด
 หน้าที่ ใช้แสดงรายการย่อแบบมีลำดับ โดยแสดงหนึ่งบรรทัดต่อหนึ่งรายการ

แอตทริบิวต์

START กำหนดค่าเริ่มต้นของเลขที่อยู่หน้ารายการนั้น โดยใช้ได้กับรายการย่อยของแท็ก เท่านั้น

TYPE กำหนดลักษณะตัวเลขหรือเครื่องหมายที่กำกับอยู่หน้ารายการย่อย สามารถใช้ได้กับรายการย่อยของแท็ก และ หากเป็นแท็ก มีค่าที่กำหนดได้คือ 1 (ใส่ตัวเลขอารบิกนำหน้า), A (ใส่ตัวพิมพ์ใหญ่ภาษาอังกฤษนำหน้า), a (ใส่ตัวพิมพ์เล็กภาษาอังกฤษนำหน้า), i (ใส่เลขโรมันแบบตัวเล็กนำหน้า), I (ใส่เลขโรมันแบบตัวใหญ่นำหน้า) ส่วนแท็ก มีค่าที่กำหนดได้คือ square, circle และ disc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<LINK>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้เพื่ออ้างอิง หรืออ้างถึงข้อมูลในไฟล์อื่น
แอตทริบิวต์	
HREF	ใช้อ้างถึงชื่อไฟล์ โดยกำหนดชื่อไฟล์แบบ URL
TITLE	ใช้กำหนดข้อความสั้นๆที่เป็นคำอธิบายไฟล์ที่อ้างถึง
REL	ใช้อธิบายความสัมพันธ์ของไฟล์ที่อ้างถึงกับไฟล์ปัจจุบัน มีค่าเป็นไปได้หลายค่า
REV	ใช้อธิบายความสัมพันธ์ของไฟล์ปัจจุบันกับไฟล์ที่อ้างถึง มักมีค่าตรงกันข้ามกับค่าของแอตทริบิวต์ REL มีค่าเป็นไปได้เหมือนแอตทริบิวต์ REL

<MAP>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบแท็ก <AREA> ซึ่งทำหน้าที่กำหนดพิกัดของพื้นที่ย่อยแต่ละส่วนภายในรูปอิมเมจแมพ
แอตทริบิวต์	
NAME	ใช้กำหนดชื่อให้แท็ก <MAP> ซึ่งจะถูกรับไปใช้อ้างอิงจากแท็กอื่นๆ เช่น แอตทริบิวต์ USEMAP ของแท็ก

<MENU>

ประเภท	มีแท็กปิด
หน้าที่	ใช้แสดงข้อความที่เป็นแบบรายการย่อย โดยมีจุดหรือสัญลักษณ์อื่นกำกับหน้ารายการย่อยแต่ละข้อ โดยข้อความในรายการย่อยมักจะยาวไม่เกิน 20 ตัวอักษร ส่วนมากแล้วผลที่ได้จะเหมือนกับการใช้แท็ก
แอตทริบิวต์	
COMPACT	ควบคุมรายการย่อยในบรรทัดต่างๆ ให้อยู่ชิดกันมากขึ้น โดยลดช่องว่างระหว่างบรรทัดลง

<META>

ประเภท	ไม่มีแท็กปิด
หน้าที่	แท็กนี้ใช้งานได้หลายอย่างเช่น ใช้ใส่ข้อมูลเพิ่มเติมให้เว็บเพจ ซึ่งข้อมูลเหล่านี้จะไม่ปรากฏบนเว็บเพจ โดยส่วนมากมักจะใช้เพื่อเตือนความจำเท่านั้น เช่น ใส่คำสำคัญ (keyword) ของเว็บเพจนี้ ใส่ชื่อผู้เขียนหรือซอฟต์แวร์ที่ใช้สร้างเว็บเพจ ซึ่งโปรแกรมค้นหา (search engine) บางตัวก็ใช้คำสำคัญนี้ในการค้นหาเว็บเพจตามคำที่ผู้เล่นเว็บระบุอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วย นอกจากนั้นแท็กนี้ยังสามารถส่งข้อมูลเพื่อติดต่อกับเว็บเซิร์ฟเวอร์ได้ เช่น สั่งให้เว็บเซิร์ฟเวอร์ส่งไฟล์ HTML ตัวใหม่มาเองโดยอัตโนมัติภายในเวลาที่กำหนด เป็นต้น

แอตทริบิวต์

HTTP-EQUIV

ใช้ระบุประเภทของ header ที่จะส่งไปพร้อมกับไฟล์เอกสาร ซึ่งค่าของ header นี้จะเก็บไว้ในแอตทริบิวต์ CONTENT สำหรับค่าของ HTTP-EQUIV ที่ใช้กันบ่อยๆก็คือ REFRESH ที่ใช้สั่งให้มีการปรับปรุงเว็บเพจหน้าจอใหม่ภายในเวลาที่กำหนดไว้ในแอตทริบิวต์ CONTENT นอกจากนี้ EXPIRES ก็เป็นอีกค่าหนึ่งที่ถูกใช้บ่อย ซึ่งจะบอกให้ทราบว่าข้อมูลต่างๆที่อยู่ในเว็บเพจเชื่อถือได้ถึงวันที่ที่ระบุไว้เท่านั้น

NAME กำหนดชื่อหรือข้อมูลให้กับตัวแปร meta (meta-variable) ดังนี้ AUTHOR, KEYWORD, GENERATOR, และ DESCRIPTION โดยเฉพาะค่าที่กำหนดให้กับ KEYWORDS จะถูกโปรแกรมค้นหา(search engine) ใช้ในการค้นหาว่าตรงตามที่ผู้ใช้ระบุมาหรือไม่

SCHEME กำหนดรูปแบบของข้อมูลเพื่อใช้ตีความว่าข้อมูลนั้นต้องอ่านอย่างไร เช่น <META SCHEME = "9-digit-Zipcode" NAME="zip" CONTENT = "02134-1078">

CONTENT สามารถใช้งานได้ทั้งสองแบบคือ ใช้กำหนดค่าให้ข้อมูลเพิ่มเติม และกำหนดค่าที่ส่งให้เว็บเซิร์ฟเวอร์

<NOFRAMES>

ประเภท

มีแท็กปิด

หน้าที่

ใช้แสดงข้อความในกรณีเว็บเบราว์เซอร์ที่ใช้ไม่สามารถแสดงผลได้ซึ่งเบราว์เซอร์จะแสดงข้อความตามที่กำหนดไว้ในแท็กนี้แทน

แอตทริบิวต์

ไม่มี

<OBJECT>

ประเภท

มีแท็กปิด

หน้าที่

ใช้เรียกโปรแกรมเสริม หรือออบเจ็กต์เข้ามาทำงานหรือไว้ในเว็บเพจ เช่น ภาพ, กราฟิก, ไฟล์เอกสาร, applet หรือ ActiveX

แอตทริบิวต์

ALIGN ใช้กำหนดการจัดวางข้อความ

WIDTH และ HEIGHT

กำหนดความกว้างและความสูงของพื้นที่ที่ใช้แสดงออบเจ็กต์ กำหนดเป็นหน่วยพิกเซลหรือเปอร์เซ็นต์

BORDER กำหนดความหนาของกรอบ กำหนดหน่วยเป็นพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HSPACE และ VSPACE

กำหนดระยะห่างระหว่างเส้นกรอบกับออบเจกต์ทั้งในแนวนอนและแนวตั้งสามารถกำหนดหน่วยเป็นพิกเซล หรือเปอร์เซ็นต์ของขนาดพื้นที่ของโปรแกรมเสริมอย่างใดอย่างหนึ่งก็ได้

CLASSID กำหนดชื่อโปรแกรมหรือไฟล์ที่ต้องการใช้

CODEBASE กำหนดที่อยู่ของไฟล์ที่เก็บออบเจกต์นั้นตามแบบ URL

CODETYPE กำหนดชนิดของไฟล์หรือโปรแกรมตามแบบ MIME

STANDBY กำหนดข้อความที่จะแสดงแทนตัวโปรแกรมเสริมหรือรูป ในกรณีที่เบราว์เซอร์ยังไม่สามารถแสดงโปรแกรมได้ หรือในขณะที่เบราว์เซอร์กำลังดาวน์โหลดโปรแกรมอยู่

TYPE กำหนดชนิดข้อมูลที่จะถูกส่งให้ออบเจกต์ตามแบบ MIME

USEMAP ใช้กำหนดให้ภาพนั้นทำหน้าที่เป็นอิมเมจแมพแบบทางฝั่งเบราว์เซอร์

ประเภท มีแท็กปิด

หน้าที่ ใช้แสดงรายการย่อยแบบที่มีเลขหรือตัวอักษรกำกับหน้ารายการย่อย โดยเรียงลำดับจากน้อยไปมาก และมีค่าเริ่มต้นเป็น 1

แอตทริบิวต์

COMPACT ควบคุมรายการย่อยในบรรทัดต่างๆ ให้อยู่ชิดกันมากขึ้น โดยลดช่องว่างระหว่างบรรทัดลง

START กำหนดค่าเริ่มต้นให้กับเลขหรือตัวอักษรที่กำกับหน้ารายการย่อย โดยปกติจะมีค่าเป็น 1 ซึ่งสามารถกำหนดค่าเริ่มต้นใหม่เป็นหมายเลขอะไรก็ได้

TYPE กำหนดชนิดหรือประเภทของค่าที่จะใช้กำกับหน้ารายการย่อย มีค่าเป็นไปได้คือ ตัวเลข (1), ตัวพิมพ์ใหญ่ (A), ตัวอักษรพิมพ์เล็ก(i), อักษรเลขโรมันตัวเล็ก(I) และอักษรเลขโรมันตัวใหญ่(I)

<OPTION>

ประเภท มีแท็กปิด

หน้าที่ สร้างฟิลด์แบบมีหลายตัวเลือก โดยผู้ใช้สามารถเลือกตัวเลือกเหล่านั้นได้ครั้งละ 1 หรือหลายตัวเลือกก็ได้

แอตทริบิวต์

DISABLED ห้ามไม่ให้ผู้ใช้เลือกตัวเลือกรายการนี้และเบราว์เซอร์จะแสดงฟิลด์เป็นสีจางกว่าฟิลด์ปกติ

SELECTED กำหนดให้รายการนี้เป็นค่าเริ่มต้น คือถูกเลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VALUE กำหนดค่าให้รายการนี้ ซึ่งจะเป็นค่าเดียวกับค่าที่ถูกเก็บไว้ในหน่วยความจำและถูกส่งไปยังเว็บเซิร์ฟเวอร์

<P>

ประเภท มีแท็กปิด
หน้าที่ ใช้ครอบข้อความเพื่อสร้างย่อหน้าใหม่ ในกรณีที่มีการใช้แท็กนี้หลายๆครั้ง เพื่อเพิ่มบรรทัดว่างๆบนหน้าจอ เว็บเบราว์เซอร์จะไม่สนใจ ซึ่งต่างจากแท็ก
 เพราะเมื่อใช้แท็ก
 หลายๆตัวจะขึ้นบรรทัดใหม่ให้เท่ากับจำนวนแท็กที่ใส่ไว้

แอตทริบิวต์

ALIGN กำหนดตำแหน่งที่ใช้วางข้อความว่าจะให้ชิดขอบด้านใด มีค่าเป็นไปได้คือ left, right, center และ justify ถ้าไม่กำหนดไว้จะมีค่าเป็น left

<PARAM>

ประเภท ไม่มีแท็กปิด
หน้าที่ ใช้กำหนดค่าพารามิเตอร์ที่จะต้องส่งให้กับโปรแกรมเสริม

แอตทริบิวต์

NAME กำหนดชื่อพารามิเตอร์
TYPE กำหนดชนิดของข้อมูลที่เบราว์เซอร์ต้องรับ กำหนดค่าตาม MIME
VALUE กำหนดค่าพารามิเตอร์
VALUETYPE กำหนดชนิดของข้อมูล เช่น DATA, REF และ OBJECT

<PRE>

ประเภท มีแท็กปิด
หน้าที่ ใช้ครอบข้อความที่ไม่ให้มีการจัดรูปแบบ โดยเว็บเบราว์เซอร์จะแสดงข้อความในแท็กนี้ด้วยฟอนต์ที่มีขนาดตัวอักษรเท่ากัน และจัดให้เหมือนกับที่ใส่ไว้ในไฟล์ HTML

แอตทริบิวต์ ไม่มี

<Q>

ประเภท มีแท็กปิด
หน้าที่ ใช้ครอบข้อความที่อ้างมาจากแหล่งอื่น โดยเว็บเบราว์เซอร์จะแสดงไว้ในเครื่องหมายคำพูด

แอตทริบิวต์

CITE อ้างถึงแหล่งต้นตอหรือเว็บเพจของคำพูดนั้น โดยกำหนดเป็นค่า URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<S>, <STRIKE>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการให้เส้นลายพาดกลางตลอดแนวข้อความนั้น
แอตทริบิวต์	ไม่มี

<SAMP>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่เป็นตัวอย่างโปรแกรมหรือสคริปต์ โดยเว็บเบราว์เซอร์มักจะแสดงข้อความนี้ด้วยฟอนต์ที่มีขนาดตัวอักษรเท่ากันทุกตัว
แอตทริบิวต์	ไม่มี

<SCRIPT>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่เป็นสคริปต์ในไฟล์ HTML
แอตทริบิวต์	
DEFER	ใช้เพื่อบอกให้เบราว์เซอร์รู้ว่าสคริปต์ชุดนี้ไม่มีการแสดงข้อมูลลงบนเว็บเพจเลย ดังนั้นระหว่างที่เว็บเบราว์เซอร์กำลังแปลงไฟล์ HTML เป็นเว็บเพจอยู่นั้นเบราว์เซอร์ไม่มีความจำเป็นต้องอ่านสคริปต์นี้เลย วิธีนี้ช่วยให้เว็บเบราว์เซอร์แสดงเว็บเพจได้เร็วขึ้น
LANGUAGE	ใช้กำหนดภาษาสคริปต์ที่ใช้ เช่น VB Script หรือ JavaScript
SRC	ใช้อ้างถึงตำแหน่งของไฟล์ที่เก็บสคริปต์ไว้ โดยต้องกำหนดในรูปของ URL
TYPE	ใช้กำหนดชนิดของสคริปต์ตามมาตรฐาน MIME เช่น text/javascript หรือ text/vbscript

<SELECT>

ประเภท	มีแท็กปิด
หน้าที่	สร้างฟิลด์แบบมีหลายตัวเลือก โดยผู้ใช้สามารถเลือกตัวเลือกเหล่านั้นได้ครั้งละ 1 หรือหลายตัวเลือกได้
แอตทริบิวต์	
DISABLED	ห้ามไม่ให้ผู้ใช้ป้อนหรือแก้ไขข้อมูลภายในฟิลด์ได้
MULTIPLE	กำหนดให้ผู้ใช้สามารถเลือกตัวเลือกได้มากกว่า 1 รายการ หากจะเลือกหลายตัวเลือก ให้กด Ctrl ค้างไว้พร้อมกับคลิกรายการที่ต้องการเลือก
NAME	กำหนดชื่อให้ฟิลด์
SIZE	กำหนดจำนวนตัวเลือกที่จะแสดงบนเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<SMALL>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการให้มีขนาดเล็กกว่าปกติหนึ่งระดับ
แอตทริบิวต์	ไม่มี

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่กำหนดให้เป็นรูปแบบพิเศษ โดยปกติมักจะใช้กับคำหรือข้อความสั้นๆที่อยู่ย่อหน้าหนึ่ง
แอตทริบิวต์	
STYLE	ใช้กำหนดรูปแบบที่ต้องการให้มีผลกับข้อความที่อยู่ภายในแท็ก และ เท่านั้น เช่น กำหนดพอนด์ ขนาด และสีของตัวอักษรที่จะใช้

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการเป็นพิเศษ โดยบราวเซอร์จะแสดงเป็นตัวหนา
แอตทริบิวต์	ไม่มี

<STYLE>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบส่วนที่เป็นข้อกำหนดสไตลชีต ซึ่งเป็นวิธีกำหนดลักษณะแท็กแบบใหม่และมีข้อดีกว่าวิธีกำหนดด้วยแอตทริบิวต์และแท็กแบบเดิมหลายประการ ดังนั้น W3C จึงพยายามผลักดันให้นักเขียนเว็บเพจหันมาใช้วิธีนี้แทนแอตทริบิวต์และแท็กเดิม
แอตทริบิวต์	
MEDIA	ใช้กำหนดชนิดของอุปกรณ์หรือสื่อที่ใช้แสดงผลซึ่งเหมาะสำหรับใช้กับข้อกำหนดสไตลชีตชุดนี้
TITLE	ใช้กำหนดข้อความสั้นๆเพื่ออธิบายสไตลชีตชุดนี้
TYPE	ใช้กำหนดชนิดของข้อมูลภายใต้แท็กนี้ สำหรับสไตลชีตแล้วมักจะเป็น text/css1

<SUB>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการให้มีขนาดเล็กและอยู่ต่ำกว่าระดับปกติ
แอตทริบิวต์	ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<SUP>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการให้ข้อความมีขนาดเล็กและสูงกว่าระดับปกติ
แอตทริบิวต์	ไม่มี

<TABLE>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อมูลและแท็กทั้งหมดที่ใช้ตาราง
แอตทริบิวต์	
ALIGN	ใช้กำหนดตำแหน่งของตารางบนเว็บเพจ มีค่าเป็นไปได้ 3 ค่าคือ left, right และ center โดยหากกำหนดเป็นค่า left หรือ right ข้อความที่อยู่ถัดจากแท็ก <TABLE> จะแสดงล้อมตารางนั้น
BORDER	กำหนดความหนาของเส้นกรอบตาราง กำหนดเป็นหน่วยพิกเซล
BGCOLOR	กำหนดสีพื้นในตาราง ซึ่งทำได้ 2 วิธีคือ กำหนดเป็นชื่อสี หรือใช้ค่า RGB
CELLPADDING	กำหนดระยะระหว่างเส้นกรอบของตารางกับข้อมูลที่อยู่ในแต่ละเซลล์กำหนดเป็นหน่วยพิกเซล
CELLSPACING	กำหนดระยะห่างระหว่างเซลล์ข้อมูล กำหนดเป็นหน่วยพิกเซล
COLS	กำหนดจำนวนคอลัมน์ในแต่ละแถว ทั้งนี้จะช่วยให้เว็บเบราว์เซอร์แสดงตารางได้รวดเร็วยิ่งขึ้น
FRAME	กำหนดว่าต้องการแสดงเส้นกรอบที่ขอบด้านใดบ้าง มี 4 ด้านคือ ด้านบน, ด้านล่าง, ด้านซ้าย และ ด้านขวา ค่าที่เป็นไปได้มีดังนี้ above ใส่กรอบด้านบน below ใส่กรอบด้านล่าง border ใส่กรอบทั้ง 4 ด้าน box เช่นเดียวกับ border hsides ใส่เส้นกรอบเฉพาะด้านบนและด้านล่าง lhs ใส่เส้นกรอบเฉพาะด้านซ้าย rhs ใส่เส้นกรอบเฉพาะด้านขวา vsides ใส่เส้นกรอบเฉพาะด้านซ้ายและด้านขวา void ไม่มีเส้นกรอบที่ด้านใดเลย
RULES	กำหนดว่าต้องการแสดงเส้นกรอบกับส่วนใดของตาราง มีค่าเป็นไปได้อย่างนี้คือ all ใส่เส้นกรอบทุกช่องในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cols	ใส่เส้นคั่นระหว่างคอลัมน์
groups	ใส่เส้นกรอบรอบกลุ่มของช่องในตารางที่กำหนดด้วยแท็กต่างๆ เช่น <THEAD>, <TBODY>, <TFOOT> หรือ <COLGROUPS>
none	ไม่แสดงกรอบที่ใดเลย
rows	ใส่เส้นคั่นระหว่างแถว
WIDTH	ใช้กำหนดความกว้างของตาราง โดยกำหนดเป็นหน่วยพิกเซล หรือเป็นเปอร์เซ็นต์ที่เทียบกับความกว้างของจอ

<TBODY>

ประเภท	มีแท็กปิด
หน้าที่	ใช้กรอบข้อมูลส่วนตัวข้อมูลของตาราง
แอตทริบิวต์	
ALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนที่เป็นข้อมูลในตาราง ว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านข้างอย่างไร มีค่าเป็นไปได้คือ left, right, center และ justify
VALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนที่เป็นข้อมูลในตาราง ว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านบนและล่างอย่างไร มีค่าเป็นไปได้คือ top, middle, bottom และ baseline

<TD>, <TH>

ประเภท	มีแท็กปิด
หน้าที่	ใช้กำหนดเซลล์ตารางโดย <TH> ใช้กับเซลล์ที่เป็นหัวตาราง ซึ่งมีผลให้ข้อความในเซลล์นั้นถูกแสดงเป็นแบบตัวหนา และอยู่ตรงกลางระหว่างความกว้างของเซลล์ ส่วนแท็ก <TD> ใช้กับส่วนที่เซลล์ข้อมูลของตาราง โดยจะแสดงเป็นแบบตัวธรรมดา ชิดกับเส้นกรอบด้านซ้ายของตาราง และอยู่กลางหากความสูงของเซลล์นั้นยาวเกินกว่าจำนวนบรรทัดของข้อความที่อยู่ในเซลล์นั้น

แอตทริบิวต์

ALIGN	ใช้กำหนดตำแหน่งข้อมูลของเซลล์ว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านข้างอย่างไร มีค่าเป็นไปได้คือ left, right, center และ justify
BGCOLOR	ใช้กำหนดสีพื้นภายในแต่ละเซลล์
COLSPAN	ใช้กำหนดว่าจะรวมอีกกี่คอลัมน์เป็นหนึ่งเซลล์
NOWRAP	สั่งให้บรรทัดข้อความไม่ต้องนำข้อความที่ยาวเกินเซลล์ไปขึ้นบรรทัดใหม่
ROWSPAN	ใช้กำหนดว่าจะรวมอีกกี่แถวเป็นหนึ่งเซลล์ที่ใช้แสดงข้อมูล
VALIGN	ใช้กำหนดตำแหน่งข้อมูลของเซลล์ว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านบนและล่างอย่างไร มีค่าเป็นไปได้คือ top, middle, bottom และ baseline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<TEXTAREA>

ประเภท	มีแท็กปิด
หน้าที่	ใช้สร้างฟิลด์รับข้อความแบบหลายบรรทัด
แอตทริบิวต์	
COLS	กำหนดจำนวนตัวอักษรที่ผู้ใช้จะป้อนได้ในแต่ละบรรทัด
DISABLED	ห้ามไม่ให้ผู้ใช้ป้อนข้อมูลในฟิลด์นี้ได้ และเบราว์เซอร์จะแสดงฟิลด์เป็นสีจางกว่าฟิลด์ปกติ
NAME	กำหนดชื่อให้กับฟิลด์ เพื่อที่จะใช้อ้างอิงในโปรแกรมได้
READONLY	กำหนดให้ฟิลด์นี้แสดงข้อมูลเพียงอย่างเดียว ห้ามผู้ใช้ป้อนหรือแก้ไขข้อมูล
ROWS	กำหนดจำนวนบรรทัดที่ให้ป้อนข้อความ

<TFOOT>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อมูลส่วนท้ายตาราง แท็กนี้ต้องอยู่ระหว่างแท็ก <TABLE> และ </TABLE>
แอตทริบิวต์	
ALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนท้ายตารางว่าจะให้อยู่ชิดกับเส้นขอบตารางด้านข้างอย่างไร มีค่าเป็นไปได้คือ left, right, center และ justify
VALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนท้ายตารางว่าจะให้อยู่ชิดกับเส้นขอบตารางด้านบนและล่างอย่างไร มีค่าเป็นไปได้คือ top, middle, bottom และ baseline

<THEAD>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อมูลส่วนหัวตาราง แท็กนี้ต้องอยู่ระหว่างแท็ก <TABLE> และ </TABLE>
แอตทริบิวต์	
ALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนหัวตารางว่าจะให้อยู่ชิดกับเส้นขอบตารางด้านข้างอย่างไร มีค่าเป็นไปได้คือ left, right, center และ justify
VALIGN	ใช้กำหนดตำแหน่งข้อมูลในเซลล์ส่วนหัวตารางว่าจะให้อยู่ชิดกับเส้นขอบตารางด้านบนและล่างอย่างไร มีค่าเป็นไปได้คือ top, middle, bottom และ baseline

<TITLE>

ประเภท	มีแท็กปิด
--------	-----------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ ใช้กำหนดชื่อหรือหัวเรื่องให้กับไฟล์ HTML ซึ่งมักจะแสดงไว้ที่ไตเติลบาร์หรือที่ขอบด้านบนของเว็บเพจนั้น รวมทั้งรายการของ history หรือ favorites ด้วยและเว็บเบราว์เซอร์จะใช้ชื่อนี้เฉพาะเพื่ออ้างถึงเว็บเพจ นอกจากนี้โปรแกรมค้นหาข้อมูล (Search Engine) บางอย่าง เช่น Yahoo หรือ Altavista ก็ใช้ข้อมูลส่วนนี้ในการจัดการเว็บเพจนั้นๆว่าควรจะอยู่ในหมวดหมู่ใด (indexing) สำหรับแท็ก <TITLE> นี้ถือได้ว่ามีความสำคัญต่อเว็บเพจมาก ดังนั้น HTML 4.0 จึงได้กำหนดไว้ว่าไฟล์ HTML ทุกไฟล์ต้องมีแท็ก <TITLE>

แอตทริบิวต์ ไม่มี

<TR>

ประเภท มีแท็กปิด

หน้าที่ ใช้กำหนดแถวของตาราง โดยอาจเป็นส่วนหัวของตาราง, ส่วนท้าย และส่วนที่เป็นข้อมูลในตาราง

แอตทริบิวต์

ALIGN ใช้กำหนดตำแหน่งข้อมูลของเซลล์ในแต่ละแถวว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านข้างอย่างไร มีค่าเป็นไปได้คือ left, right, center และ justify

BGCOLOR ใช้กำหนดสีพื้นภายในแต่ละแถว

VALIGN ใช้กำหนดตำแหน่งข้อมูลของเซลล์ในแต่ละแถวว่าจะให้อยู่ชิดกับเส้นกรอบตารางด้านบนและล่างอย่างไร มีค่าเป็นไปได้คือ top, middle, bottom และ baseline

<TT>

ประเภท มีแท็กปิด

หน้าที่ ใช้ครอบข้อความที่ต้องการให้ใช้ฟอนต์แบบที่มีขนาดตัวอักษรเท่ากันทุกตัว เช่น Courier

แอตทริบิวต์ ไม่มี

<U>

ประเภท มีแท็กปิด

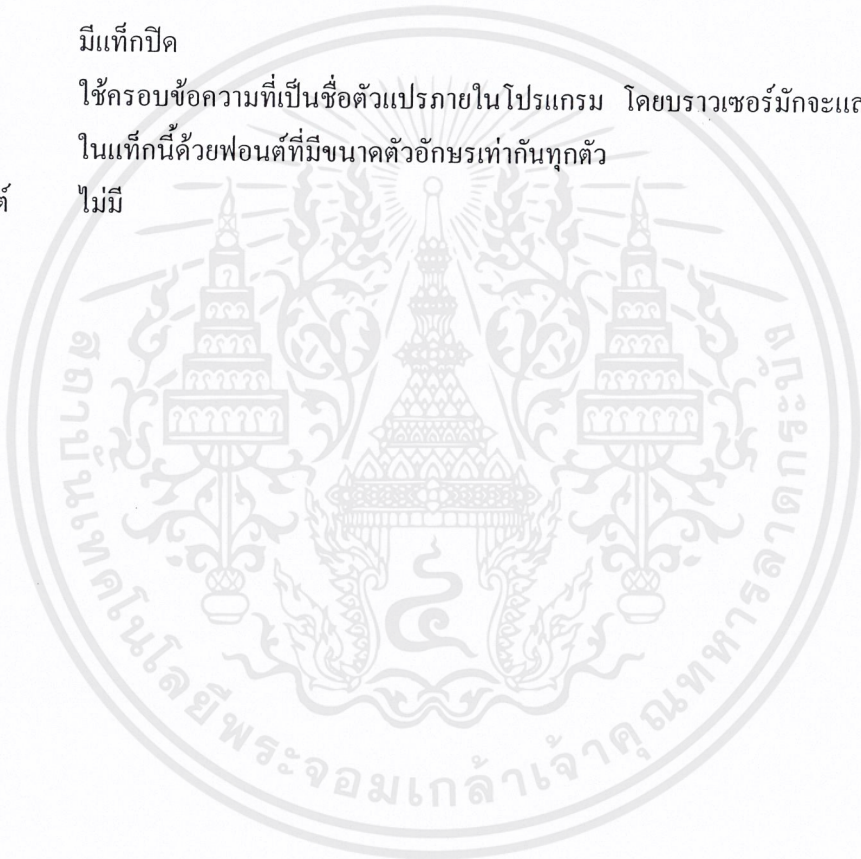
หน้าที่ ใช้ครอบข้อความที่ต้องการให้มีขีดเส้นใต้ข้อความนั้น

แอตทริบิวต์ ไม่มี

ประเภท	มีแท็กปิด
หน้าที่	ใช้แสดงรายการย่อยแบบที่มีจุดหรือสัญลักษณ์กำกับหน้ารายการย่อย
แอตทริบิวต์	
COMPACT	ควบคุมรายการย่อยในบรรทัดต่างๆ ให้อยู่ชิดกันมากขึ้น โดยช่องว่างระหว่างบรรทัด ลง
TYPE	ใช้กำหนดลักษณะของสัญลักษณ์ที่จะใช้กำกับหน้าแต่ละรายการย่อย มีค่าเป็นไปได้ดังนี้ disc, square และ circle

<VAR>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่เป็นชื่อตัวแปรภายในโปรแกรม โดยบราวเซอร์มักจะแสดงข้อความ ในแท็กนี้ด้วยฟอนต์ที่มีขนาดตัวอักษรเท่ากันทุกตัว
แอตทริบิวต์	ไม่มี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 แท็ก WML

<a>

ประเภท	มีแท็กปิด
หน้าที่	สร้างลิงก์ โดยเปรียบเสมือนเป็นตัวย่อ หรือชอร์ตคัตของแท็ก <anchor> กับแท็กคำสั่งของงาน <go> นั่นเอง
แอตทริบิวต์	
href	กำหนดชื่อ URI หรือการ์ดที่ต้องการเชื่อมโยงไปถึง
title	ใส่ข้อความที่เป็นชื่อ ที่จะอธิบายลิงก์อย่างย่อ

<access>

ประเภท	มีแท็กปิด
หน้าที่	ควบคุมการเข้าถึงข้อมูลของผู้เข้าชม (access) หรือต้องการอนุญาตให้ผู้เข้าชมเฉพาะกลุ่มเข้ามาอ่านข้อมูลได้เฉพาะ WAP เพจบางหน้าเท่านั้น
แอตทริบิวต์	
domain	กำหนดชื่อของโดเมนเนมหรือที่อยู่ (IP Address) ของเอกสาร WML
path	ชื่อของตำแหน่งหรือโฟลเดอร์ ที่อยู่ภายในโดเมนเนมที่ได้ระบุไว้ใน domain

<anchor>

ประเภท	มีแท็กปิด
หน้าที่	สร้างลิงก์ เพื่อเชื่อมโยงระหว่างการ์ดหรือหน้า WAP เพจ ซึ่งสามารถทำงานได้เช่นเดียวกับแท็ก <do> แต่จะต่างกันตรงที่ข้อความในแท็ก <anchor> จะไม่ปรากฏในเมนู Options ของซอฟต์แวร์ แต่จะปรากฏบนหน้าจอ โดยมีลักษณะเป็นลิงก์ที่เราสามารถเลือกได้เลย และเนื่องจากแท็กนี้ จะต้องปรากฏบนหน้าจอ จึงต้องใส่แท็กนี้ในระหว่างแท็ก <p> ด้วย เหมือนข้อความทั่ว ๆ ไป และในแท็กนี้จะต้องคำสั่งของงาน (Task) ที่ทำหน้าที่เปลี่ยนการ์ดหรือหน้า WAP เพจ เมื่อลิงก์ของแท็กนี้ถูกเลือก ซึ่งแท็กของงานที่สามารถอยู่ในแท็กนี้คือ go, prev และ refresh
แอตทริบิวต์	
title	ใส่ข้อความที่เป็นชื่อ ที่จะอธิบายลิงก์อย่างย่อ

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการทำให้หนา
แอตทริบิวต์	ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<big>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการทำให้ตัวอักษรมีขนาดใหญ่ขึ้น
แอตทริบิวต์	ไม่มี

ประเภท	ไม่มีแท็กปิด
หน้าที่	กำหนดให้ขึ้นบรรทัดใหม่สำหรับข้อความที่อยู่ต่อจากแท็กนี้
แอตทริบิวต์	ไม่มี

<card>

ประเภท	มีแท็กปิด
หน้าที่	กำหนดจุดเริ่มต้นและจุดสิ้นสุดของการ์ด ในแต่ละการ์ดหนึ่งสามารถมีการ์ดได้หลาย ๆ การ์ด ซึ่งแต่ละการ์ดก็จะบรรจุข้อมูลและวิธีการตอบโต้กับผู้ใช้ที่แตกต่างกันไป
แอตทริบิวต์	
title	ใช้สำหรับใส่ข้อความที่เป็นชื่อ ที่จะอธิบายภาพรวมของข้อมูลในการ์ด
id	ใช้สำหรับกำหนดชื่อให้กับการ์ด เพื่อสามารถอ้างอิงจากการ์ดอื่นได้

<do>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ในการช่วยให้ผู้ใช้สามารถเข้าถึง และเชื่อมโยงจากการ์ดที่แสดงอยู่ในขณะนั้นไปสู่การ์ดหรือ URI อื่นๆ ได้ ซึ่งแท็ก <do> จะทำหน้าที่เชื่อมต่อระหว่างสิ่งที่แสดงออกมาบนหน้าจอภาพกับงานที่เราต้องการให้หน้า WAP ทำ โดยแท็ก <do> นี้ สามารถใช้ได้กับทั้งระดับเด็กและการ์ด
แอตทริบิวต์	
type	เป็นตัวแสดงแก่เครื่องโทรศัพท์มือถือ (WAP บราวเซอร์) ให้ทราบว่าแท็กนั้นถูกใช้เพื่อทำงานอะไร และยังช่วยให้บราวเซอร์สร้างอินเทอร์เฟซที่เหมาะสมกับงานนั้น ๆ ด้วย ซึ่งแอตทริบิวต์นี้ต้องมีเสมอ มิฉะนั้นจะเกิด Syntax error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าแอตทริบิวต์	ความหมาย
Accept	เป็นค่าที่ใช้กันมาก มีไว้สำหรับแสดงการยอมรับให้ทำงานตามที่กำหนด เป็นตัวแสดงถึงการย้อนกลับไปยังหน้าก่อน
Prev	
Help	เป็นตัวแสดงความต้องการความช่วยเหลือ
Reset	เป็นตัวแสดงถึงตัวเลือก หรือการทำงานอื่น ๆ ที่สามารถเพิ่มเติมได้
Options	เป็นตัวแสดงถึงการลบ หรือเริ่มต้นการทำงานใหม่ เป็นตัวแสดงถึงการลบวัตถุ หรือตัวเลือกต่าง ๆ
Delete	เป็นชนิดที่ใช้ทั่วไป
Unknown	เป็นชนิดที่เราไม่ได้กำหนดลงไป หรือเกิดจากการพิมพ์ค่าของแอตทริบิวต์ ที่ไม่มีในนิยามหรือพิมพ์ผิด มีความหมายเหมือนกับ unknown
"" (empty string)	
x-*, X-*	เป็นชนิดที่ใช้ทดลอง
vnd.*, VND.*	ผู้ขายหรือยูสเซอร์เอเจนต์กำหนดชนิดของ vnd.co-type ซึ่ง co คือ ชื่อบริษัท และ type เป็น do-type

label	ใช้กำหนดชื่อของลิงก์ ที่จะแสดงออกมาบนหน้าจอในเมนูได้
optional	เป็นแอตทริบิวต์ที่อนุญาตให้ WAP บราวเซอร์ คิดว่าแท็กนี้เป็นตัวเลือก ซึ่งจะนำมาใช้ หรือไม่ก็ได้ โดยสามารถกำหนดค่าเป็น "true" หรือ "false" ถ้ากำหนดเป็น "true" เครื่องอาจจะไม่นำแท็กนี้มาใช้ก็ได้
name	ใช้กำหนดชื่อของแท็ก <do> เพื่อใช้ในการอ้างอิง

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการเน้นหรือให้ความสำคัญ
แอตทริบิวต์	ไม่มี

<fieldset>

ประเภท	มีแท็กปิด
หน้าที่	จัดกลุ่มให้กับส่วนรับข้อมูล ซึ่งจะทำให้บราวเซอร์แสดงส่วนรับข้อมูลต่าง ๆ ได้ดีขึ้น นอกจากจะมีแท็กที่ทำหน้าที่รับข้อมูลแล้ว ยังสามารถใส่แท็กต่างๆ ได้อีกดังต่อไปนี้
	<ul style="list-style-type: none"> - แท็ก fieldset (fieldset ซ้อน fieldset) - แท็ก a - แท็ก img - แท็ก tab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แท็ก br
- แท็กการตกแต่งตัวอักษร เช่น em, b, u

แอตทริบิวต์

title กำหนดชื่อให้กับแท็ก

<go>

ประเภท มีแท็กปิด

หน้าที่ ใช้กำหนดให้ WAP บราวเซอร์เปลี่ยนหน้าของ WAP เพจ ไปตามตำแหน่งที่กำหนดขึ้น โดยแอตทริบิวต์ href ซึ่งตำแหน่งที่กำหนดใน href อาจจะเป็น URI หรือชื่อของ card หรือ deck ในไซต์นั้นก็ได้

แอตทริบิวต์

href ใช้กำหนด URI หรือชื่อของการ์ดที่ต้องการเชื่อมต่อไปถึง

sendreferer สามารถกำหนดค่าเป็น true หรือ false ก็ได้ ซึ่งเป็นแอตทริบิวต์ที่อนุญาตให้เครื่อง PDA ส่ง URI ของหน้า WAP เพจนั้นไปยังเซิร์ฟเวอร์ เมื่อทำการติดต่อกับเครื่องเซิร์ฟเวอร์ โดยปกติเราจะกำหนดค่าแอตทริบิวต์นี้เป็น true เมื่อเราต้องการควบคุมบุคคลที่จะเข้าถึงข้อมูลในเครื่องเซิร์ฟเวอร์

method กำหนดวิธีการส่งค่าผ่านโปรโตคอล HTTP ซึ่งค่าได้ 2 อย่าง คือ POST หรือ GET

- GET เป็นวิธีการส่งค่าของข้อมูลไปพร้อมกับค่าของ URL โดยมีเครื่องหมาย ? กั้นระหว่างชื่อของตัวแปรที่ทำการส่งกับ URL และทางเซิร์ฟเวอร์จะสามารถรับค่าเหล่านี้ได้โดยการใช้คำสั่ง QUERRY_STRING
- POST เป็นวิธีการส่งข้อมูลมาทาง stdin ซึ่งมีหน่วยเป็นไบต์ และจำนวนไบต์ของข้อมูลที่ส่งจะถูกเก็บค่าไว้ในตัวแปร CONTENT_LENGTH การส่งข้อมูลด้วยวิธีนี้จะเป็นการป้องกันมิให้ข้อมูลอ่านได้ง่ายกว่าวิธีการส่งแบบ GET เพียงเล็กน้อย

accept-charset

กำหนดลิสต์ของการเข้ารหัสตัวอักษรที่เซิร์ฟเวอร์สามารถรองรับได้

<head>

ประเภท มีแท็กปิด

หน้าที่ เป็นส่วนหัวของเอกสาร เป็นส่วนที่เก็บข้อมูลและรายละเอียดที่มีความสัมพันธ์กับเอกสาร WML ทั้งเอกสาร (deck) แท็กที่สามารถมีได้ภายในแท็ก <head> คือ แท็ก <access> และ <meta>

แอตทริบิวต์ ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<i>

ประเภท	มีแท็กปิด
หน้าที่	ใช้ครอบข้อความที่ต้องการทำให้เป็นตัวอักษรเอียง
แอตทริบิวต์	ไม่มี

ประเภท	ไม่มีแท็กปิด
หน้าที่	แสดงรูปภาพลงในหน้า WAP
แอตทริบิวต์	
src	แสดงตำแหน่งของที่อยู่และชื่อของไฟล์รูปภาพที่ต้องการนำมาใช้ หรือที่เรียกว่า URI ของรูปภาพ
alt	เป็นทางเลือก (alternative) ของการแสดงผลของรูปภาพ ในกรณีที่เบราว์เซอร์ไม่สามารถแสดงภาพที่ต้องการออกมาได้ เบราว์เซอร์จะแสดงข้อความที่กำหนดในแอตทริบิวต์นี้ออกมาแทน
localsrc	ระบุไคคอนที่ได้ถูกกำหนดไว้แล้ว ถ้าแอตทริบิวต์นี้ระบุไคคอนถูกต้อง จะถูกใช้แทนแอตทริบิวต์ src และ alt
align	ทำหน้าที่จัดวางตำแหน่งของรูปภาพกับข้อความ โดยมีการกำหนดได้ 3 แบบ คือ <ul style="list-style-type: none"> - การวางข้อความเสมอขอบบนของรูปภาพ (top) - การวางข้อความในตำแหน่งตรงกลางของรูปภาพ (middle) - การวางข้อความเสมอขอบล่างของรูปภาพ (bottom)
vspace	กำหนดขนาดของช่องว่างที่ต้องการใส่ระหว่างขอบของรูปภาพในตำแหน่งบนและล่าง
hspace	กำหนดขนาดของช่องว่างที่ต้องการใส่ระหว่างขอบของรูปภาพในตำแหน่งซ้ายและขวา
height	ทำหน้าที่จองขนาดความสูงพื้นที่ของเบราว์เซอร์ล่วงหน้าเพื่อทำการโหลดรูปนั้น การกำหนดพื้นที่สำหรับดาวน์โหลดรูปจะมีประโยชน์ คือเบราว์เซอร์สามารถแสดงข้อความต่อจากพื้นที่ ที่เรากำหนดได้เลย โดยไม่ต้องรอให้เบราว์เซอร์โหลดรูปเสร็จ
width	ทำหน้าที่จองขนาดความกว้างพื้นที่ของเบราว์เซอร์ล่วงหน้าเพื่อทำการโหลดรูปนั้น

<input/>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้รับข้อมูลที่เป็นตัวอักษร ซึ่งพิมพ์โดยผู้ใช้
แอตทริบิวต์	
name	กำหนดชื่อตัวแปรที่ใช้สำหรับเก็บข้อมูลที่ผู้ใช้ใส่เข้ามา ดังนั้นจึงต้องกำหนดค่าให้กับแอตทริบิวต์นี้ทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

value	กำหนดค่าเริ่มต้นหรือค่าอัตโนมัติให้กับตัวแปรที่ระบุในแอตทริบิวต์ name และเป็นค่าที่จะแสดงหน้าจอครั้งแรก เมื่อผู้ใช้ได้เข้ามาในหน้านี้
type	ใช้กำหนดการแสดงตัวอักษรในช่องรับข้อมูล สามารถกำหนดได้ 2 ค่า คือ <ul style="list-style-type: none"> - text หมายถึง กำหนดให้การแสดงตัวอักษรให้เป็นไปตามปกติ - password หมายถึง กำหนดให้การแสดงตัวอักษรไม่ตรงตามที่ผู้ใช้พิมพ์เข้ามา โดยทั่วไปแล้วตัวอักษรที่บราวเซอร์แสดงคือเครื่องหมายดอกจัน
title	ใช้สำหรับตั้งชื่อให้กับแท็ก
maxlength	ใช้กำหนดจำนวนตัวอักษรที่มากที่สุดที่ผู้ใช้สามารถใส่ได้ ถ้าหากไม่กำหนดค่าให้กับแอตทริบิวต์นี้ บราวเซอร์จะถือว่าจำนวนตัวอักษรที่ใส่ได้นั้นมีจำนวนไม่จำกัด
format	ใช้กำหนดรูปแบบในการรับข้อมูล

ค่าแอตทริบิวต์	ความหมาย
A	อนุญาตให้ใส่ได้เฉพาะตัวอักษรตัวใหญ่ และสัญลักษณ์ต่าง ๆ ที่ไม่ใช่ตัวเลข
a	อนุญาตให้ใส่ได้เฉพาะตัวอักษรตัวเล็ก และสัญลักษณ์ต่าง ๆ ที่ไม่ใช่ตัวเลข
N	อนุญาตให้ใส่ได้เฉพาะตัวเลข
X	อนุญาตให้ใส่ได้เฉพาะตัวอักษรตัวใหญ่
x	อนุญาตให้ใส่ได้เฉพาะตัวอักษรตัวเล็ก
M	อนุญาตให้ใส่ตัวอักษรอะไรก็ได้ แต่บราวเซอร์จะถือว่าเป็นอักษรตัวใหญ่
m	อนุญาตให้ใส่ตัวอักษรอะไรก็ได้ แต่บราวเซอร์จะถือว่าเป็นอักษรตัวเล็ก
*f	อนุญาตให้ใส่ตัวอักษรเป็นจำนวนเท่าไรก็ได้ โดยที่ f คือ ค่าต่าง ๆ ที่ผ่านมา เช่น *A, *a, *N เป็นต้น
nf	อนุญาตให้ใส่ตัวอักษรเป็นจำนวนเท่ากับค่า n โดย n มีค่าระหว่าง 1-9 และ f คือค่าต่าง ๆ ที่ผ่านมา เช่น 3a, 5M เป็นต้น
\c	กำหนดให้แสดงตัวอักษรหลังเครื่องหมาย \ เช่น \- ในช่องใส่ข้อมูลจะมีเครื่องหมายขีด (-) แสดง

emptyok	สำหรับอินพุตที่อนุญาตให้ผู้ใช้ใส่หรือไม่ใส่ค่าก็ได้ ค่าที่สามารถกำหนดให้แอตทริบิวต์นี้คือ <ul style="list-style-type: none"> - true หมายความว่า ผู้ใช้ไม่ต้องใส่ข้อมูลก็ได้ - false หมายความว่า ผู้ใช้จะต้องใส่ข้อมูลก็ได้
tabindex	ใช้กำหนดลำดับที่ของแท็บ ในกรณีที่ผู้ใช้กดปุ่มแท็บ
size	ใช้กำหนดความกว้างของช่องใส่ข้อมูล โดยจะวัดเป็นจำนวนตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<meta/>

ประเภท	ไม่มีแท็กปิด
หน้าที่	เป็นส่วนที่ให้ข้อมูลทั่ว ๆ ไปที่เกี่ยวข้องกับเอกสาร WML ทั้งเอกสาร (deck) ซึ่งข้อมูลส่วนนี้ไม่สามารถถูกเก็บไว้ในส่วนของเนื้อหาภายในการ์ดได้ งานหลักๆ ของแท็ก <meta> มีดังต่อไปนี้
	<ol style="list-style-type: none"> 1. กำหนดคำอธิบาย และคำที่แสดงความหมายของ WAP เพจ 2. ใช้ป้องกันมิให้บราวเซอร์ดาวน์โหลดเอกสาร WML จากแคช (cache) 3. เปลี่ยนค่า URI หรือต้องการดึงข้อมูลจากเว็บเซิร์ฟเวอร์ เมื่อถึงเวลาที่กำหนดโดยอัตโนมัติ

แอตทริบิวต์

content	ใช้ในการกำหนดค่าคุณสมบัติของเอกสารที่ต้องการส่งให้เซิร์ฟเวอร์ หรือใช้ในการกำหนดค่าให้กับคุณสมบัติของเอกสารที่กำหนดในแอตทริบิวต์ name
name	ใช้ในการระบุชื่อของคุณสมบัติของเอกสาร เช่น keywords, description หรือ author
http-equiv	จะใช้แทนแอตทริบิวต์ name เพื่อจะนำค่าของข้อมูลไปสร้าง HTTP header ซึ่งเป็นข้อมูลที่สำคัญที่รับส่งข้อมูลโดยโปรโตคอล HTTP
forua	มีค่าได้ 2 ค่า คือ “true” หรือ “false” ซึ่งจะกำหนดค่าเป็นเท็จ เมื่อต้องการให้เกิดเว็ลบข้อมูลของแท็ก <meta> ออก เมื่อทำการส่งข้อมูลไปยังเครื่องโทรศัพท์ของผู้รับ และมีค่าเป็นจริง เมื่อเราบังคับให้เกิดเว็ลบต้องส่งค่าของแท็ก <meta> ไปยังเครื่องของผู้รับ
scheme	ใช้ในการกำหนดรูปแบบ หรือมาตรฐานที่ใช้ในการตีความข้อมูลที่อยู่ในแอตทริบิวต์ โดยค่าของ scheme ที่จะใช้ อาจจะแตกต่างกันไปในแต่ละความหมายของแท็ก <meta>

<noop>

ประเภท	มีแท็กปิด
หน้าที่	ใช้เพื่อทำการยกเลิกการทำงานของเทมเพลต (template) ในระดับเด็ก โดยมากจากคำว่า “no operation” หรือแปลว่า “ไม่ทำอะไรเลย”
แอตทริบิวต์	ไม่มี

<onevent>

ประเภท	มีแท็กปิด
หน้าที่	เป็นแท็กของเหตุการณ์ที่สามารถกำหนดชนิดของเหตุการณ์เป็น onenterforward, onenterbackward หรือ ontimer ได้ พร้อมทั้งสามารถกำหนดคำสั่งของงาน (Task) ที่จะใช้ได้อีกด้วย และการใช้แท็ก <onevent> นี้ มีประโยชน์มากในการควบคุมการทำงานของตัวเครื่องโทรศัพท์มือถือได้มากขึ้นเพราะว่า แท็ก <onevent> นี้จะถูกใช้งานเป็นแท็กที่กำหนดหน้าที่กำกับแท็กคำสั่งของงาน (เช่น go, prev, refresh หรือ noop) อีกที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงมีประโยชน์มากกว่าการใช้เหตุการณ์ (event) เพื่อเป็นแอตทริบิวต์ในแท็ก <card> หรือ <template> อย่างเดียว เพราะสามารถกำหนดคำสั่งที่ต้องการทำได้มากขึ้น

แอตทริบิวต์

type ใช้กำหนดเหตุการณ์ที่ใช้สำหรับเปลี่ยน URI

- ontimer

เหตุการณ์ ontimer จะใช้เป็นแอตทริบิวต์ในแท็ก <card> และ <template> เพื่อกำหนดหน้า WAP เพจจะเปลี่ยนไปสู่การ์ดใบใหม่ หรือที่อยู่ใหม่ชื่ออะไร ในระยะเวลาที่กำหนด ซึ่งจะใช้คู่กับแท็ก <timer>

- onenterforward

เหตุการณ์ onenterforward จะเกิดขึ้นเมื่อผู้ใช้เปิดการ์ดที่มีแอตทริบิวต์นี้ขึ้นมาใช้งาน โดยอาจจะเข้ามาโดยคำสั่ง <go> หรือวิธีอื่น ๆ ก็ได้ และเหตุการณ์ onenterforward นี้ก็เป็นแอตทริบิวต์หนึ่งของแท็ก <card> และ <template> ซึ่งจะเป็นตัวกำหนดค่าของ URI ที่ต้องการจะไป เมื่อการ์ดนี้ถูกเปิดขึ้นมา

- onenterbackward

เป็นเหตุการณ์ที่เกิดขึ้นเช่นเดียวกับเหตุการณ์ของ onenterforward แต่จะแตกต่างกันตรงที่เหตุการณ์นี้จะเกิดขึ้น เมื่อการ์ดนี้ถูกเปิดโดยคำสั่ง <prev/> หรือโดยวิธีอื่น ๆ ก็ได้ ซึ่งเกิดจากการที่ผู้เยี่ยมชมอ่านค่า URI ที่ได้รับจากที่เก็บไว้ใน history นั้นเอง

<option>

ประเภท มีแท็กปิด

หน้าที่ ใช้ร่วมกับแท็ก <select>

แอตทริบิวต์

value ใช้กำหนดค่าของตัวเลือก ซึ่งค่านี้จะเป็นค่าที่ตัวแปรในแอตทริบิวต์ name ของแท็ก select รับไปเมื่อผู้ใช้เลือกตัวเลือกนี้

index ใช้กำหนดลำดับของตัวเลือก

title ใช้ตั้งชื่อให้กับแท็ก

onpick เป็นเหตุการณ์ที่จะทำงานเมื่อผู้ใช้เลือกหรือยกเลิกการเลือกเมนูรับค่า (input) เมื่อตัวเลือกที่มีแอตทริบิวต์นี้ถูกเลือก เหตุการณ์ onpick นี้จะทำหน้าที่กำหนด URI ของหน้า WAP เพจใหม่ที่จะนำมาแสดงผลนั่นเอง

ใช้สำหรับลิงก์ไปยังหน้าอื่น ๆ เมื่อมีการเลือกตัวเลือก หรือการยกเลิกการเลือกเกิดขึ้น ในกรณีที่เลือกได้เพียงตัวเลือกเดียว แอตทริบิวต์นี้จะทำงาน (ลิงก์ไปยังหน้าที่กำหนด

ไว้) เฉพาะมีการเลือกเกิดขึ้นเท่านั้น ส่วนในกรณีที่สามารถเลือกได้หลายตัวเลือก แอตทริบิวต์นี้จะทำงานทั้งเมื่อมีการเลือก และยกเลิกการเลือกเกิดขึ้น

<optgroup>

ประเภท	มีแท็กปิด
หน้าที่	จัดกลุ่มให้กับตัวเลือก จะช่วยให้บราวเซอร์แสดงตัวเลือกต่าง ๆ ได้ง่ายขึ้น มีการใช้ร่วมกับแท็ก select และ option
แอตทริบิวต์	
title	กำหนดชื่อให้กับกลุ่มตัวเลือก

<p>

ประเภท	มีแท็กปิด
หน้าที่	ระบุพารากราฟของข้อมูลสู่จอภาพ โดยเป็นตัวกำหนดการจัดวางตัวอักษร และการแสดงตัวอักษร
แอตทริบิวต์	
align	ใช้กำหนดการจัดวางตัวอักษร ซึ่งสามารถกำหนดได้ 3 แบบ คือ ซิดซ้าย ซิดขวา และตรงกลาง (left, right, center) แต่ถ้าหากว่าไม่ได้กำหนดค่าให้กับแอตทริบิวต์นี้ บราวเซอร์จะใช้ค่าอัตโนมัติคือ จัดซิดซ้าย
mode	ใช้สำหรับจัดการกับข้อความในแต่ละบรรทัดว่าจะให้ขึ้นบรรทัดใหม่หรือไม่ ถ้าข้อความนั้นยาวเกินกว่าหน้าจอจะแสดงได้ในหนึ่งบรรทัด ค่าของแอตทริบิวต์นี้คือ wrap และ nowrap
	wrap หมายถึง กำหนดให้การแสดงตัวอักษร จะต้องขึ้นบรรทัดใหม่ ถ้าข้อความมีความยาวมากกว่าที่จะแสดงได้ในหนึ่งบรรทัด
	nowrap หมายถึง กำหนดให้การแสดงตัวอักษร ไม่ต้องขึ้นบรรทัดใหม่ ถ้าข้อความมีความยาวมากกว่าที่จะแสดงได้ในหนึ่งบรรทัด
	ถ้าหากเราไม่ได้กำหนดค่าให้กับแอตทริบิวต์นี้ ค่าที่บราวเซอร์จะใช้คือ ค่าที่เรากำหนดไว้ครั้งสุดท้ายในแท็ก <p> ที่อยู่ในกรณีนั้น หรือถ้าหากเราไม่เคยกำหนดค่านี้มาก่อนเลย ค่าที่บราวเซอร์ใช้คือ wrap

<postfield>

ประเภท	มีแท็กปิด
หน้าที่	ส่งค่าตัวแปรไปยังเพจอื่นและสคริปต์ ทั้งแบบ GET และ POST โดยการส่งแบบ GET นั้นจะส่งโดยใช้แท็ก postfield หรือไม่ใช้ก็ได้ แต่การส่งแบบ GET นั้นจะต้องใช้แท็ก postfield

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอตทริบิวต์

name	ชื่อตัวแปรที่จะส่งค่าที่ผู้ใช้เลือก
value	ค่าตัวแปรที่จะถูกส่งกลับไปยังไฟล์ที่ทำหน้าที่เป็น โปรแกรมจัดการข้อมูล

<prev/>

ประเภท	ไม่มีแท็กปิด
หน้าที่	สร้างลิงก์กลับไปหน้าเก่า เพื่อให้สามารถย้อนกลับไปยังหน้าที่ผ่านมาแล้ว เมื่อเรากlickที่ตัวเลือกเพื่อใช้แท็กนี้ ค่าของ URI ที่อยู่ใน history จะถูกนำมาออกมาอ่านและ WAP เภจจะเปลี่ยนเป็นหน้าที่นำออกมาจาก history

แอตทริบิวต์ ไม่มี

<refresh/>

ประเภท	ไม่มีแท็กปิด
หน้าที่	ใช้ในการโหลดหน้า WAP เภจขึ้นมาแสดงใหม่ หรือเพื่อโหลดข้อมูลรวมทั้งค่าของตัวแปรล่าสุดออกมาแสดงอีกครั้งหนึ่ง

แอตทริบิวต์ ไม่มี

<select>

ประเภท	มีแท็กปิด
หน้าที่	แสดงตัวเลือกในลักษณะเมนู ใช้ร่วมกับแท็ก <option>
แอตทริบิวต์	
name	ใช้สำหรับตั้งชื่อตัวแปรที่จะรับค่าตัวเลือกที่ผู้ใช้เลือก
iname	ใช้สำหรับตั้งชื่อตัวแปรที่ใช้เก็บค่าดัชนีของตัวเลือกที่ถูกเลือก
value	ใช้สำหรับกำหนดค่าเริ่มต้นให้กับตัวแปรที่ตั้งไว้ในแอตทริบิวต์ name ในกรณีที่กำหนดการเลือกให้สามารถเลือกได้หลายตัวเลือก ค่าแต่ละค่าต้องขึ้นด้วยเครื่องหมายเซมิโคลอน (;)
ivalue	ใช้สำหรับกำหนดค่าดัชนีของตัวเลือกเริ่มต้นให้กับตัวแปรที่ตั้งไว้ด้วยแอตทริบิวต์ iname ถ้าหากเป็นการเลือกได้หลายตัวเลือกค่าของดัชนีจะต้องขึ้นด้วยเครื่องหมายเซมิโคลอน (;)
title	ใช้ตั้งชื่อให้กับแท็ก
multiple	ใช้สำหรับกำหนดจำนวนตัวเลือกที่สามารถเลือกได้ โดยค่าที่สามารถกำหนดได้คือ true หรือ false <ul style="list-style-type: none"> - true หมายความว่า ผู้ใช้สามารถเลือกได้หลายตัวเลือก - false หมายความว่า ผู้ใช้สามารถเลือกได้เพียงตัวเลือกเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

tabindex ใช้กำหนดลำดับที่ของแท็บ ในกรณีที่ผู้ใช้กดปุ่มแท็บ

<setvar>

ประเภท มีแท็กปิด
 หน้าที่ กำหนดตัวแปร ซึ่งจะใช้เพียงแคภายในแท็ก <refresh>, <go > เท่านั้น
 แอตทริบิวต์
 name ชื่อของตัวแปรที่กำหนด
 value ค่าของตัวแปรที่ระบุ

<small>

ประเภท มีแท็กปิด
 หน้าที่ ใช้ครอบข้อความที่ต้องการทำให้ตัวอักษรมีขนาดเล็กลง
 แอตทริบิวต์ ไม่มี

ประเภท มีแท็กปิด
 หน้าที่ ใช้ครอบข้อความที่ต้องการเน้นหรือให้ความสำคัญเป็นพิเศษ
 แอตทริบิวต์ ไม่มี

<table>

ประเภท มีแท็กปิด
 หน้าที่ สร้างตาราง เพื่อนำเสนอข้อมูลในลักษณะตาราง โดยเป็นตัวกำหนดจุดเริ่มต้นและจะสิ้นสุดของตาราง
 แอตทริบิวต์
 title กำหนดชื่อตาราง
 align ใช้สำหรับจัดวางตัวอักษรหรือรูปภาพที่แสดงในแต่ละคอลัมน์ ค่าที่กำหนดให้แอตทริบิวต์ได้แก่ C, L และ R
 C หมายถึง จัดให้อยู่กึ่งกลางคอลัมน์
 L หมายถึง จัดให้อยู่ชิดซ้ายของคอลัมน์
 R หมายถึง จัดให้อยู่ชิดขวาของคอลัมน์
 การกำหนดการจัดวางนี้สามารถระบุให้แต่ละคอลัมน์ได้ด้วยการระบุค่าเรียงกันไป
 columns ใช้สำหรับระบุจำนวนคอลัมน์ที่มีตาราง ซึ่งจะต้องกำหนดค่าให้แอตทริบิวต์นี้ทุกครั้ง ถ้าหากจำนวนแท็ก td มีน้อยกว่าค่าที่กำหนดให้กับแอตทริบิวต์นี้ บราวเซอร์จะทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มคอลัมน์ว่างให้โดยอัตโนมัติ แต่ถ้าหากจำนวนแท็ก td มีมากกว่าค่าที่เรากำหนด บรรทัดจะใส่คอลัมน์ที่เกินมาลงทางซ้ายของตาราง

<td>

ประเภท

มีแท็กปิด

หน้าที่

ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของคอลัมน์หนึ่งคอลัมน์ในตาราง

ข้อมูลต่าง ๆ ที่ต้องการนำเสนอจะต้องอยู่ในแท็ก td ดังนั้นภายในแท็ก td จึงมีแท็กย่อยอื่น ๆ เพื่อใช้ควบคุมการแสดงผล หรือการโต้ตอบกับผู้ใช้แท็กเหล่านั้น ได้แก่

แท็ก a (<a>), แท็ก anchor (<anchor>) ใช้สำหรับเชื่อมโยงไปยังข้อมูลส่วนอื่น หรือการ์ดอื่น ๆ

แท็ก br (
) ใช้ขึ้นบรรทัดใหม่

แท็ก img () ใช้สำหรับแสดงรูปภาพ

แอตทริบิวต์

ไม่มี

<template>

ประเภท

มีแท็กปิด

หน้าที่

กำหนดคำสั่งงาน (task) หรือเหตุการณ์ (event) ต่าง ๆ ซึ่งคำสั่งของเหตุการณ์หรืองานที่เขียนในเทมเพลตนี้จะสามารถนำมาใช้ได้กับทุกการ์ดในเด็กนั้น ยกเว้นคำสั่งของงานนั้นจะถูกยกเลิกโดยคำสั่งอื่นที่เขียนขึ้นมาภายในการ์ดอีกที ภายในเทมเพลตนี้สามารถมีแท็ก <do> เพื่อทำหน้าที่ใส่คำสั่งงาน หรือแท็ก <onevent> เพื่อกำหนดคำสั่งงานเมื่อมีเหตุการณ์ต่าง ๆ เกิดขึ้น

แอตทริบิวต์

ไม่มี

<timer/>

ประเภท

ไม่มีแท็กปิด

หน้าที่

เป็นแท็กที่ช่วยในการกำหนดระยะเวลาที่เราต้องการนั่นเอง การนับเวลาของ timer จะเป็นการนับถอยหลังตั้งแต่เราเริ่มเปิดการ์ดที่มีแท็กนี้ขึ้นมา และจะหยุดเวลาเมื่อออกจากการ์ดใบนั้น timer จะนับเวลาถอยหลังไปเรื่อย ๆ จนกระทั่งหมดเวลา หรือได้เวลาเท่ากับ 0 จากนั้นจึงจะทำการบังคับให้เหตุการณ์ที่กำหนดใน ontimer ทำงาน โดยที่หนึ่งการ์ดสามารถมีแท็ก timer ได้เพียงแค่ 1 แท็กเท่านั้น

แอตทริบิวต์

name

กำหนดชื่อของตัวแปรที่ใช้เก็บค่าของเวลา

value

กำหนดค่าของเวลาที่เราต้องการ ซึ่งการกำหนดเวลาของ timer จะมีค่าเป็นหน่วยละ 1/10 วินาที และแท็ก timer นี้จะต้องมีแอตทริบิวต์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<tr>

ประเภท มีแท็กปิด
 หน้าที่ ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของแถวหนึ่งแถวในตาราง
 แอตทริบิวต์ ไม่มี

<u>

ประเภท มีแท็กปิด
 หน้าที่ ใช้ครอบข้อความที่ต้องการขีดเส้นใต้
 แอตทริบิวต์ ไม่มี

<wml>

ประเภท มีแท็กปิด
 หน้าที่ ระบุนจุดเริ่มต้นและจุดสิ้นสุดของเด็ก
 แอตทริบิวต์ ไม่มี



ก.3 XSLT

ก.3.1 XSL อิลิเมนต์

xsl:apply-imports

หน้าที่ เรียกใช้เทมเพลตในสไตลชีตที่ถูกอิมพอร์ตมา
 แอตทริบิวต์ ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิลิเมนต์ child ไม่มี

xsl:apply-templates

หน้าที่ เปรียบเทียบแต่ละอิลิเมนต์ child ของเอกสาร XML กับเทมเพลตในสไตลชีต ถ้าหากแมตช์กัน จะทำการเอาต์พุตเทมเพลตที่แมตช์กับโหนดนั้น โดยภายในเทมเพลตที่แมตช์กับโหนดก็จะมีอิลิเมนต์ xsl:apply-templates เพื่อหาโหนดที่จะแมตช์กับเทมเพลตภายในโหนด child ของตัวมันเองต่อไป โดยในการเปรียบเทียบจะทำในทุก ๆ โหนดของต้นไม้เอกสาร XML

แอตทริบิวต์

select ใช้เลือกเซตของอิลิเมนต์ child ที่ต้องการแบบเจาะจง แทนที่จะเลือกทั้งหมด

mode

อิลิเมนต์ child sort with-param

xsl:attribute

หน้าที่ กำหนดค่าแอตทริบิวต์ให้กับเอกสารเอาต์พุต

แอตทริบิวต์

name ชื่อแอตทริบิวต์ของอิลิเมนต์

namespace

อิลิเมนต์ child ไม่มี

xsl:attribute-set

หน้าที่ ประยุกต์กลุ่มที่เหมือนกันของแอตทริบิวต์ เข้ากับอิลิเมนต์ที่แตกต่างกัน ไม่ว่าจะอยู่ในคลาสเดียวกันหรือต่างกันก็ตาม โดยทำการกำหนดแอตทริบิวต์ต่างๆ รวมเข้าเป็นสมาชิกของเซตแอตทริบิวต์ ที่ระดับบนของสไตลชีต

แอตทริบิวต์

name กำหนดชื่อให้กับเซตของแอตทริบิวต์

use-attribute-sets

รวมเซตของแอตทริบิวต์เข้ากับอิลิเมนต์

อิลิเมนต์ child attribute

xsl:call-template

หน้าที่ ปรากฏในเนื้อหาของกฎเทมเพลต เพื่อเรียกใช้ซึ่งเมื่อมีการประมวลผล xsl:call-template จะถูกแทนที่ด้วยเนื้อหาของอิลิเมนต์ xsl:template ตามชื่อที่ระบุ

แอตทริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

name	ชื่อของเทมเพลตที่จะทำการเรียกใช้
อิลิเมนต์ child	with-param
xsl:choose	
หน้าที่	เลือกจากหนึ่งในเอาต์พุตที่เป็นไปได้หลาย ๆ เอาต์พุต โดยขึ้นกับเงื่อนไขที่เป็นไปได้หลายๆ อย่าง แต่ละเงื่อนไขและเทมเพลตเอาต์พุตจะขึ้นกับอิลิเมนต์ child คือ xsl:when และ xsl:otherwise
แอตทริบิวต์	ไม่มี
อิลิเมนต์ child	when, otherwise
xsl:comment	
หน้าที่	แทรกคอมเมนต์ในเอกสารเอาต์พุต โดยคอนเทนต์ของมันเป็นข้อความที่อยู่ในคอมเมนต์
แอตทริบิวต์	ไม่มี
อิลิเมนต์ child	ไม่มี
xsl:copy	
หน้าที่	ก๊อปปี้โหนด source ไปยังเอาต์พุตแต่ไม่ได้ก๊อปปี้อิลิเมนต์ child, แอตทริบิวต์ หรือเนื้อหาข้อมูลให้โดยอัตโนมัติ จะเป็นประโยชน์เมื่อมีการแปลงจากเอกสารหนึ่งไปยังภาษามาร์กอัปอื่นที่มีความคล้ายคลึงกัน แต่ xsl:copy จะสามารถก๊อปปี้ได้เพียงแค่โหนด source
แอตทริบิวต์	use-attribute-sets
อิลิเมนต์ child	ไม่มี
xsl:copy-of	
หน้าที่	ก๊อปปี้โหนดที่นอกเหนือจากโหนด source ได้
แอตทริบิวต์	
select	เลือกโหนดที่จะทำการก๊อปปี้
อิลิเมนต์ child	ไม่มี
xsl:for-each	
หน้าที่	ประมวลผลหลาย ๆ อิลิเมนต์ โดยจะประมวลผลในทุก ๆ อิลิเมนต์ที่ถูกเลือกในแอตทริบิวต์ select
แอตทริบิวต์	select

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิลิเมนต์ child ไม่มี

xsl:if

หน้าที่ สร้างทางเลือกของเอาต์พุต โดยขึ้นกับรูปแบบที่กำหนดในอินพุต

แอตทริบิวต์

test เป็นค่าบูลีน ถ้าหากเป็นจริง ข้อมูลของอิลิเมนต์ xsl:if นั้นจะเป็นเอาต์พุต

อิลิเมนต์ child ไม่มี

xsl:import

หน้าที่ รวมสไตลชีตเข้าด้วยกัน โดยต้องเป็นอิลิเมนต์ระดับบน ก่อนอิลิเมนต์ระดับบนใด ๆ ทั้งหมดในอิลิเมนต์รูตของ xsl:stylesheet ถ้าหากกฎของสไตลชีตที่ถูกอิมพอร์ตเข้ามาขัดแย้งกับกฎของสไตลชีตที่อิมพอร์ต จะใช้กฎของสไตลชีตที่อิมพอร์ต แต่ถ้ามีกฎของสไตลชีตที่ถูกอิมพอร์ตเข้ามาขัดแย้งกันจะถือเอากฎของสไตลชีตที่อิมพอร์ตอันสุดท้ายเป็นหลัก

แอตทริบิวต์

href ระบุ URI ของสไตลชีตที่อิมพอร์ต

อิลิเมนต์ child ไม่มี

xsl:include

หน้าที่ เป็นอิลิเมนต์ที่อยู่ระดับบน ที่ก๊อปปี้สไตลชีตอื่น ๆ เข้าไปในสไตลชีตปัจจุบัน โดยทุก ๆ สไตลชีตจะมีความสำคัญเท่ากัน

แอตทริบิวต์

href ระบุ URI ของสไตลชีตที่จะรวมเข้าไป (include)

อิลิเมนต์ child ไม่มี

xsl:number

หน้าที่ แทรกตัวเลขในเอกสารเอาต์พุต

แอตทริบิวต์

level กำหนดระดับในการนับ

- any นับทุก ๆ อิลิเมนต์ที่เหมือนกับโหนด current

- single นับโหนด sibling ของโหนด source

- multiple นับทั้งโหนด sibling และ โหนด ancestor ของโหนด current

count ระบุสิ่งที่จะทำการนับ

from ระบุว่าการนับอิลิเมนต์จะเริ่มที่ในทรี่ของอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

value	ค่าตัวเลขที่จะทำการแทรก
format	กำหนดรูปแบบของค่าตัวเลข
lang	
letter-value	เป็นตัวบอกว่าตัวอักษรจะถูกแปลเป็นตัวอักษรหรือเป็นตัวเลข โดยมีค่าแอตทริบิวต์คือ alphabetic (ลำดับของตัวอักษร) กับ traditional (ลำดับของตัวเลข)
grouping-separator	ระบุตัวแบ่งกลุ่มใช้ในการแบ่งกลุ่มของดิจิทัล (digit)
grouping-size	ระบุจำนวนของดิจิทัลที่ถูกใช้ในกลุ่ม
อิลิเมนต์ child	ไม่มี
xsl:otherwise	
หน้าที่	หากไม่มีเงื่อนไขใดของอิลิเมนต์ xsl:when เป็นจริงจะเอาต์พุตข้อมูลของอิลิเมนต์ลูกของ xsl:otherwise
แอตทริบิวต์	ไม่มี
อิลิเมนต์ child	ไม่มี
xsl:output	
หน้าที่	กำหนดรูปแบบของการเอาต์พุต
แอตทริบิวต์	
method	ระบุวิธีการเอาต์พุต โดยทั่วไปจะมีค่า 1 ใน 3 ค่า ต่อไปนี้ - text - xml - html
version	ระบุเวอร์ชันของ XML ที่ถูกใช้ใน XML Declaration
indent	ถ้า มีค่าเป็น yes (ดีฟอลต์มีค่าเป็น no) หมายถึงตัวประมวลผลถูกอนุญาตให้สามารถเพิ่มอ (แต่ลบออกไม่ได้) ช่องว่างพิเศษในเอาต์พุตได้เพื่อพยายามให้อเอาต์พุตมีรูปแบบที่สวยงาม
encoding	ระบุวิธีการเข้ารหัส
media-type	กำหนดชนิด MIME (MIME type) ของเอาต์พุต
doctype-system	ตัวบ่งชี้ SYSTEM สำหรับ DTD
doctype-public	ตัวบ่งชี้ PUBLIC สำหรับ DTD
omit-xml-declaration	มีค่าเป็น yes หรือ no ถ้าค่าเป็น yes การประกาศค่าของ XML (XML declaration) จะไม่ถูกรวมในเอาต์พุตด้วย
standalone	เซตแอตทริบิวต์ yes หรือ no ใน XML Declaration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cdata-section-elements

ใส่ชื่อของอิลิเมนต์พิเศษซึ่งข้อมูลที่เป็นแท็กจะถูกซ่อน (wrap) ในส่วนของ CDATA

อิลิเมนต์ child ไม่มี

param

หน้าที่ กำหนดพารามิเตอร์เพื่อเป็นค่าตัวแปรส่งให้กับเทมเพลตในการสร้างเอาต์พุต โดยเป็นอิลิเมนต์ child ของอิลิเมนต์ `xsl:template`

แอตทริบิวต์ name ชื่อพารามิเตอร์
select กำหนดค่าสตริงของพารามิเตอร์

อิลิเมนต์ child ไม่มี

preserve-space

หน้าที่ คงช่องว่างที่อยู่ข้างหน้าและข้างกลางอิลิเมนต์ไว้

แอตทริบิวต์ elements

อิลิเมนต์ child ไม่มี

xsl:processing-instruction

หน้าที่ สร้างคำสั่งประมวลผล ในเอกสารเอาต์พุต คอนเทนต์ของอิลิเมนต์จะเป็นคอนเทนต์ของคำสั่งประมวลผล

แอตทริบิวต์ name ระบุเป้าหมายของคำสั่งประมวลผล

อิลิเมนต์ child ไม่มี

xsl:sort

หน้าที่ จัดลำดับของอิลิเมนต์เอาต์พุตให้แตกต่างจากที่ปรากฏในอินพุต โดยจะกำหนดคีย์ที่ใช้ในการเรียงอิลิเมนต์ของเอาต์พุต โดยคีย์ฟอลด์จะเรียงลำดับตามตัวอักษร ดูจากตัวแรกจากนั้นก็พิจารณาตัวที่สอง, สาม ไปเรื่อย ๆ

แอตทริบิวต์

select เลือกอิลิเมนต์ที่จะทำการเรียงลำดับ

lang เซตภาษาของคีย์

data-type เซตชนิดของข้อมูลที่เรียงลำดับ เช่น number

order กำหนดได้ว่าจะเรียงจากอิลิเมนต์ไหนก่อน ระหว่างเรียงจาก ascending หรือ decending

case-order กำหนดได้สองค่า คือ upper-first หรือ lower-first เพื่อกำหนดว่าจะเรียงตัวอักษรตัวใหญ่หรือตัวเล็กก่อน

อิลิเมนต์ child ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

xsl:strip-space

แอตทริบิวต์	elements
อิลิเมนต์ child	ไม่มี
หน้าที่	ลบช่องว่างที่อยู่ข้างหน้าและข้างหลังอิลิเมนต์ออกไป

xsl:stylesheet

หน้าที่	เป็นอิลิเมนต์รากของเอกสาร XSLที่ใช้ XML ในการอธิบายกฎเกณฑ์, แพตเทิร์น และ เทมเพลต โดยจะประกอบไปด้วยกฎของเทมเพลตต่าง ๆ
แอตทริบิวต์	exclude-result-prefixes, extension-element-prefixes, id โดยมีแอตทริบิวต์ที่ต้องระบุค่าที่แน่นอน คือ
version	1.0
xmlns:xsl	http://www.w3.org/1999/XSL/Transform
อิลิเมนต์ child	import, include, strip-space, preserve-space, output, key, decimal-format, namespace-alias, attribute-set, variable, param, template

xsl:template

หน้าที่	กำหนดกฎของเทมเพลต เอาต์พุตเทมเพลต คือคอนเทนต์ของอิลิเมนต์นี้
แอตทริบิวต์	
match	แพตเทิร์นของกฎ
name, priority, mode	
อิลิเมนต์ child	ไม่มี

xsl:text

หน้าที่	แทรกข้อมูลในเอกสารเอาต์พุตเหมือนกับเท็กซ์ที่ต้องการจริง ๆ โดยมีข้อดี 2 ข้อ คือ อย่างแรก สามารถเอาต์พุตช่องว่างในสไลด์ซีตตามที่กำหนดจริง เพราะช่องว่างในสไลด์ซีตจะไม่ถูกมองว่าสำคัญ และจะถูกกลดลงเหลือแค่ช่องว่างเดียวโดยอัตโนมัติ ข้อสองคือ สามารถแทรกตัวอักษรพิเศษ < และ & ได้เลย โดยไม่ต้องแปลเป็น < และ & ก่อน
แอตทริบิวต์	disable-output-escaping
อิลิเมนต์ child	ไม่มี

xsl:value-of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ จัดการเกี่ยวกับค่าของโหนด โดยการก๊อปปี้ค่าของบางสิ่งจากเอกสารอินพุตไปยังเอกสารเอาต์พุต ค่าของโหนด ส่วนใหญ่เป็นสตริง บางทีอาจเป็นค่าว่างก็ได้ โดยสตริงนี้ขึ้นกับชนิดของโหนด ชนิดของโหนดส่วนใหญ่ คืออิลิเมนต์และ ค่าของโหนดอิลิเมนต์ เป็นการต่อกันของทุกข้อมูลตัวอักษรที่ถูกพาร์ (แต่ไม่ใช่มาร์กอัป) ระหว่างแท็กเริ่มต้นกับแท็กสิ้นสุดของอิลิเมนต์

xsl:value-of เหมาะสำหรับการใช้เพียงแค่นั้นในคอนเท็กซ์ที่มีโหนดไม่ซับซ้อนนัก แต่ถ้ามีไอเท็มจำนวนมากมายที่ถูกเลือก จะมีเพียงโหนดแรกเท่านั้นที่จะถูกเลือก

แอตทริบิวต์ select ระบุสิ่งที่ต้องการค่า ที่แน่นอน

disable-output-escaping

อิลิเมนต์ child ไม่มี

xsl:variable

หน้าที่ กำหนดค่าคงที่สำหรับใช้ได้ตลอดทั้งสไตลชีต โดยเป็นค่าแอตทริบิวต์ในเทมเพลต ค่าของ xsl:variable สามารถประกอบด้วยมาร์กอัปรวมทั้งคำสั่ง XSL อื่น ๆ ได้ด้วย ดังนั้นจึงหมายความว่าสามารถคำนวณค่าของตัวแปร โดยใช้ข้อมูลอื่น ๆ ได้รวมทั้งค่าของตัวแปรอื่นด้วย อย่างไรก็ตามตัวแปรไม่สามารถอ้างถึงตัวเองได้ไม่ว่าทางตรงหรือทางอ้อมก็ตาม โดยสามารถเป็น children ของระดับบนสุดของอิลิเมนต์รูดของ xsl:stylesheet หรือ อยู่ในเทมเพลตต่าง ๆ ก็ได้ ตัวแปรที่อยู่ระดับบนของสไตลชีตสามารถเข้าถึงจากที่ใดก็ได้ได้ในเทมเพลตต่าง ๆ เรียกว่า ตัวแปรโกลบอล แต่ตัวแปรที่ประกาศในเทมเพลตใดๆ สามารถเข้าถึงได้เพียงแค่อิลิเมนต์ following-sibling และ descendant ของมันเท่านั้น เรียกว่า ตัวแปรโลคอล หากตัวแปรโลคอลมีชื่อเหมือนตัวแปรโกลบอลให้ใช้ตัวแปรโลคอล แต่ในกรณีที่มีตัวแปร โลคอลมีชื่อเหมือนกันจะใช้ตัวแปร โลคอลที่ใกล้ที่สุดเท่านั้น

แอตทริบิวต์

name ชื่อตัวแปรที่จะใช้อ้างถึง

select

อิลิเมนต์ child ไม่มี

xsl:when

หน้าที่ เป็นเงื่อนไขของของ xsl:choose และกำหนดเทมเพลตของเอาต์พุต ถ้าหากมีเงื่อนไขที่เป็นจริงมากกว่า 1 ทางเลือก จะทำงานที่เงื่อนไขแรกเท่านั้น

แอตทริบิวต์

test เป็นเอ็กเพรสชันของ Xpath โดยเป็นค่าบูลีน

อิลิเมนต์ child ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

xsl:with-param

หน้าที่ กำหนดพารามิเตอร์เพื่อเป็นค่าตัวแปรส่งให้กับเทมเพลตในการสร้างเอาต์พุต โดยเป็นอิลิเมนต์ child ของอิลิเมนต์ xsl:call-template หรือ xsl:apply-template

แอตทริบิวต์ name, select

อิลิเมนต์ child ไม่มี

ก.3.2 ค่าของโหนด

Node Type	Value
Root	ค่าของอิลิเมนต์รูต
Element	การต่อกันของทุกข้อมูลตัวอักษรที่ถูกพาร์ภายในอิลิเมนต์ รวมทั้งตัวอักษรข้อมูลในทุก ๆ descendant ของอิลิเมนต์
Text	เท็กซ์ของโหนด โดยเฉพาะของโหนดนั่นเอง
Attribute	ค่าแอตทริบิวต์นี้ถูกกำหนดโดยส่วนที่ 3.3 ของข้อกำหนด XML 1.0 โดยทั่วไปแล้วค่าแอตทริบิวต์หลังเอนทิตีจะถูกตัดช่องว่างที่อยู่ข้างหน้าและหลังออก และไม่รวมชื่อของแอตทริบิวต์ เครื่องหมายเท่ากับ และเครื่องหมายคำพูดด้วย
Namespace	URI ของ namespace
Processing instruction	ค่าของคำสั่งประมวลผล แต่ไม่รวมชื่อของคำสั่งประมวลผล, <? หรือ ?>
Comment	เท็กซ์ของคอมเมนต์ ไม่รวม <!-- และ →

ก.3.3 รูปแบบสำหรับการแมตซ์โหนด

แอตทริบิวต์ match

สำหรับอิลิเมนต์ xsl:template สนับสนุนไวยากรณ์ที่มีความซับซ้อน ในการแสดงโหนดที่ต้องการและไม่ต้องการแมตซ์

แอตทริบิวต์ select

สำหรับ xsl:apply-templates, xsl:value-of, xsl:for-each, xsl:copy-of และ xsl:sort สนับสนุนซูเปอร์เซตของโหนดที่ต้องการและไม่ต้องการเลือก

แมตซ์กับโหนดรูต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารเอาต์พุตที่มีรูปแบบถูกต้อง เอาต์พุตแรกจากการการแปลงรูปแบบของ XSL ควรเป็นเอาต์พุตของอิลิเมนต์รูตของเอกสาร ดังนั้นโดยทั่วไปสไตล์ชีตของ XSL จึงเริ่มต้นทำงานกับโหนดรูต ในการระบุโหนดรูต ค่าของแอตทริบิวต์ match เท่ากับ “/”

แมตช์กับชื่ออิลิเมนต์

รูปแบบที่เป็นพื้นฐานที่สุดประกอบไปด้วยชื่ออิลิเมนต์เดี่ยวซึ่งจะแมตช์กับทุกอิลิเมนต์ที่มีชื่อเดียวกัน

แมตช์ children ด้วย /

ในแอตทริบิวต์ match ไม่สามารถจำกัด children ของโหนด current ดังนั้นจึงใช้สัญลักษณ์ / เพื่อให้แมตช์กับไฮราคีของอิลิเมนต์ที่ระบุเท่านั้น ถ้าหากใช้เดี่ยว ๆ สัญลักษณ์ / จะอ้างอิงโหนดรูต อย่างไรก็ตามสามารถใช้ระหว่างชื่อ ในการระบุว่าชื่อที่สองเป็น child ของชื่อแรก

ขณะที่ / อ้างถึง โหนดรูต แต่ /* อ้างถึงอิลิเมนต์รูต

แมตช์ descendant ด้วย //

อ้างอิงอิลิเมนต์ descendant ที่ระดับใด ๆ โดยจะทำการเลือกแพตเทิร์นที่ descendant ใด ๆ ของโหนดรูต

แมตช์ด้วย ID

แมตช์แอตทริบิวต์ด้วย เครื่องหมาย @

แมตช์คำสั่งประมวลผลด้วย processing-instruction()

แมตช์โหนดเท็กซ์ด้วย text()

การใช้โอเปอเรเตอร์ OR ด้วยเครื่องหมาย |

อนุญาตให้เทมเพลตหนึ่งสามารถแมตช์ได้หลายแพตเทิร์น ถ้ามีโหนดหนึ่งแมตช์กับหนึ่งแพตเทิร์นหรืออื่น ๆ มันจะเรียกให้เทมเพลตนั้นทำงาน

การตรวจสอบ ด้วยเครื่องหมาย []

ตรวจสอบเกี่ยวกับมีโหนดที่แมตช์กับแพตเทิร์น

ก.3.4 Node Axes

ตารางสรุป Node Axes และความหมายของมัน

Axis	เลือกจาก
Ancestor	Parent ของโหนด current, parent ของ parent ของโหนด current , parent ของ parent ของ parent ของโหนด current ออกมาจนกระทั่งถึงโหนด root
Ancestor-or-self	Ancestor ของโหนด current และ ตัวโหนด current เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3.6 Expression Type (ชนิดของเอ็กเพรสชัน)

Node Set

โหนดเซต	คือ กลุ่มของโหนดจากเอกสารอินพุต
คอนเท็กซ์โหนด	คือ สมาชิกของคอนเท็กซ์โหนดลิสต์
คอนเท็กซ์โหนดลิสต์	คือ กลุ่มของอิลิเมนต์ที่แมตช์กับกฎเดียวกันในเวลาเดียวกัน เช่น ผลลัพธ์จาก <code>xsl:for-each</code>

ฟังก์ชันที่จัดการกับโหนดเซต

ฟังก์ชัน	ชนิดที่ตอบกลับ	สิ่งที่ตอบกลับ
<code>position()</code>	ตัวเลข	ตำแหน่งของคอนเท็กซ์โหนดในคอนเท็กซ์โหนดลิสต์ โหนดแรกจะมีตำแหน่งเท่ากับ 1
<code>last()</code>	ตัวเลข	จำนวนของโหนดในเซตคอนเท็กซ์โหนด
<code>count(node-set)</code>	ตัวเลข	จำนวนของโหนดใน node-set
<code>id(string)</code>	โหนดเซต	โหนดเซตที่มีอิลิเมนต์เดียวที่ไต่ตามในเอกสาร โดยจะมี ID เป็นสตริง หรือ เซตว่าง ถ้าไม่มีอิลิเมนต์ที่มี ID ระบุ
<code>key(string name, Object value)</code>	โหนดเซต	โหนดเซตที่ประกอบไปด้วยทุกโหนดในเอกสาร ที่มีคีย์กับค่าที่กำหนด โดยคีย์จะถูกเซตไว้ที่อิลิเมนต์ <code>xsl:key</code> ที่อยู่ระดับบน
<code>document(string URI, string base)</code>	โหนดเซต	โหนดเซตในเอกสารหรือส่วนที่อ้างถึงโดย URI โหนดจะถูกเลือกจากชื่อของ anchor หรือ Xpointer ที่ถูกใช้โดย URI ถ้าหากไม่มีชื่อของ anchor หรือ XPointer แล้วอิลิเมนต์รูตของเอกสารคือโหนดเซต โดย relative URI จะอ้างอิงกับ base URI หรือโหนด current ในเอกสารอินพุตถ้าไม่กำหนด base URI
<code>local-name(node set)</code>	โหนดเซต	ส่วนโลคอล (ทุกสิ่งหลัง prefix ของ namespace) ของโหนดแรกในอาร์กิวเมนต์ node set โดยสามารถใช้โดยไม่ต้องใส่อาร์กิวเมนต์เพื่อรับส่วนโลคอลของโหนดคอนเท็กซ์
<code>Namespace-uri(node set)</code>	สตริง	URI ของ namespace ของโหนดแรกในโหนดเซต โดยสามารถใช้โดยไม่ต้องใส่อาร์กิวเมนต์เพื่อรับ URI ของ namespace ของคอนเท็กซ์โหนด โดยจะส่งค่า ว่างเปล่า (empty string) ถ้าโหนดอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		namespace ที่เป็นดีฟอลต์
name(node set)	สตริง	ชื่อของโหนดแรกในอาร์กิวเมนต์โหนดเซต สามารถถูกใช้โดยไม่ต้องมีอาร์กิวเมนต์ในการรับชื่อของคอนเท็กซ์โหนด
Generate-id(node set)	สตริง	ตัวระบุพิเศษสำหรับโหนดแรกในอาร์กิวเมนต์ node set และสามารถถูกใช้โดยปราศจากอาร์กิวเมนต์ใดๆ เพื่อสร้าง ID สำหรับคอนเท็กซ์โหนด

Boolean

บูลีนประกอบด้วยสองค่าคือ true (จริง) และ false (เท็จ) และถ้าหากต้องการใช้สตริงหรือตัวเลขเป็นค่าบูลีน สามารถทำการแปลงได้โดยใช้ฟังก์ชัน boolean() และยังมีคีย์เวิร์ดที่เกี่ยวข้องคือ and และ or เพื่อจัดการกับค่าบูลีนสองค่าตามกฎตรรกะ ส่วน not ไม่เป็นคีย์เวิร์ดใน XSL แต่เป็นฟังก์ชัน not() ถ้าต้องการกลับค่าบูลีน

number

หากต้องการแปลงค่าที่ไม่เป็นตัวเลขเช่น สตริงหรือบูลีน ให้เป็นตัวเลขสามารถทำได้โดยใช้ฟังก์ชัน number()

โดยมีฟังก์ชันในการจัดการกับตัวเลข 4 ฟังก์ชันคือ

floor ()	return อินทิเจอร์ที่ใหญ่ที่สุดที่เล็กกว่าตัวเลขนั้น
ceiling()	return อินทิเจอร์ที่เล็กที่สุดที่ใหญ่กว่าตัวเลขนั้น
round()	return อินทิเจอร์ที่ใกล้กับตัวเลขที่สุด
sum()	return ผลบวกของแต่ละอาร์กิวเมนต์

String

สตริง คือ ซีควเนต์ของคาแร็กเตอร์ยูนิโค้ด ซึ่งข้อมูลชนิดอื่น สามารถแปลงเป็นสตริงได้ โดยใช้ string()

ตารางแสดงฟังก์ชันที่จัดการเกี่ยวกับสตริง

ฟังก์ชัน	ชนิดที่ตอบกลับ	สิ่งที่ตอบกลับ
start-with(main_string, prefix_string)	บูลีน	ค่า true ถ้า main_string เริ่มต้นด้วย prefix_string ถ้าไม่ใช่ค่าจะเป็น false
Contains(containing_string, contained_string)	บูลีน	ค่า true ถ้า มี contained_string อยู่ใน containing_string ถ้าไม่ใช่ค่าจะเป็น false
Substring(string, offset, length)	สตริง	สตริงที่มีจำนวนคาแรกเตอร์เท่ากับ length เริ่มจาก offset ที่ระบุใน string หรือทุกคาแรกเตอร์จาก offset ไปยังสุด string ถ้าไม่มีการกำหนด length
Substring-before(string, marker-string)	สตริง	ส่วนหนึ่งของ string จากเริ่มต้นสตริง ไปจนถึง (แต่ไม่ได้รวม) marker-string ตำแหน่งแรก
Substring-after(string, marker-string)	สตริง	ส่วนหนึ่งของ string จากท้าย ของ marker-string ตำแหน่งแรกไปจนถึงสิ้นสุด string
string-length(string)	ตัวเลข	จำนวนของคาแรกเตอร์ใน string
Normalize-space(string)	สตริง	String ที่ถูกตัดช่องว่างทั้งข้างหน้าและข้างหลัง และตัดช่องว่างที่ต่อกันให้เหลือเพียงแค่ช่องว่างเดียวเท่านั้น
translate(string, replaced_text, replacement_text)	สตริง	สิ่ง string ที่ตำแหน่งของคาแรกเตอร์ใน replace_text ถูกแทนที่ด้วยคาแรกเตอร์จาก replacement_text
concat(string1, string2,...)	สตริง	ส่งการต่อกันของสตริงทั้งหมดที่ถูกส่งเป็นอาร์กิวเมนต์ในลำดับที่มันถูกส่งค่า
Format-number(number, format-string, locale-string)	สตริง	ส่งกลับรูปแบบสตริงของตัวเลขที่ถูกกำหนดรูปแบบตามที่ format-string ที่กำหนด เหมือนกับคลาสของ Java 1.1 โดยมี locale-string เป็นอาร์กิวเมนต์ออปชันที่กำหนดชื่อของอิลิเมนต์ xsl:decimal-format ที่ถูกใช้เพื่อแปลสตริงให้มีรูปแบบ

Result Tree Fragments

เป็นส่วนของเอกสาร XML ที่มีโหนดหรือเซตของโหนดไม่สมบูรณ์ การใช้ฟังก์ชัน document() กับ URI ที่ชี้ไปยังส่วนกลางของอิลิเมนต์จะช่วยสร้างส่วนที่หายไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การติดตั้งโปรแกรม

ข.1 ความต้องการของระบบ

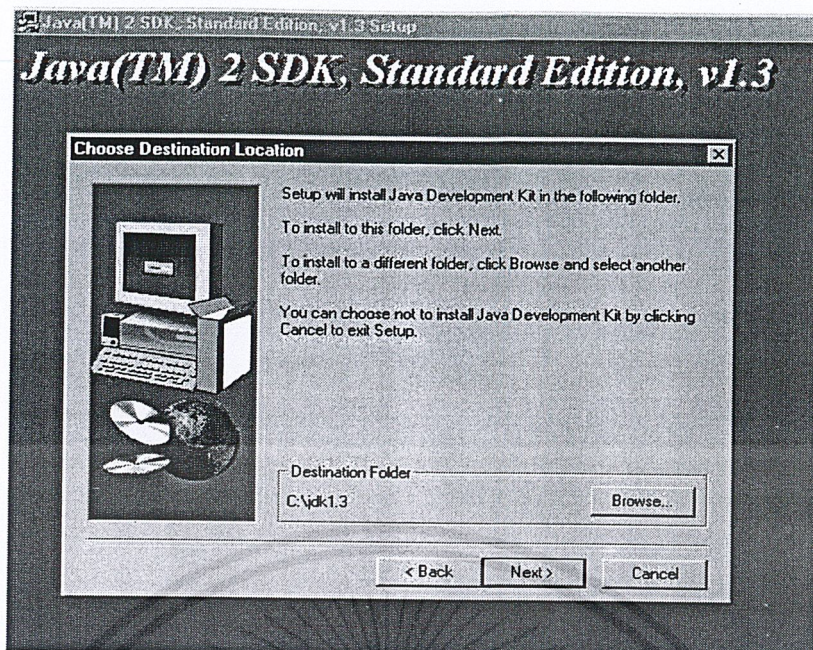
ระบบปฏิบัติการ (OS)	: วินโดว์ 98 / 2000 / NT
หน่วยประมวลผลกลาง (CPU)	: ทำงานได้ดีที่ความเร็วตั้งแต่ 300 เมกะเฮิร์ต ขึ้นไป
หน่วยความจำ (RAM)	: 64 MB หรือ มากกว่า
หน่วยความจุ (Hard Disk)	: 4 GB ขึ้นไป
ความละเอียดจอภาพ (Monitor)	: 65,000 สี หรือมากกว่า
การเชื่อมต่อโครงข่าย	: เครือข่ายอินเทอร์เน็ต
การติดตั้งโปรแกรมฝั่งเซิร์ฟเวอร์	: JDK 1.3, เว็บเซิร์ฟเวอร์ที่รองรับการทำงานเซิร์ฟเล็ต,
การติดตั้งโปรแกรมฝั่งไคลเอนท์	: แอ็พบราวเซอร์

ข.2 การติดตั้งระบบเพื่อใช้งานตัวแปลภาษา

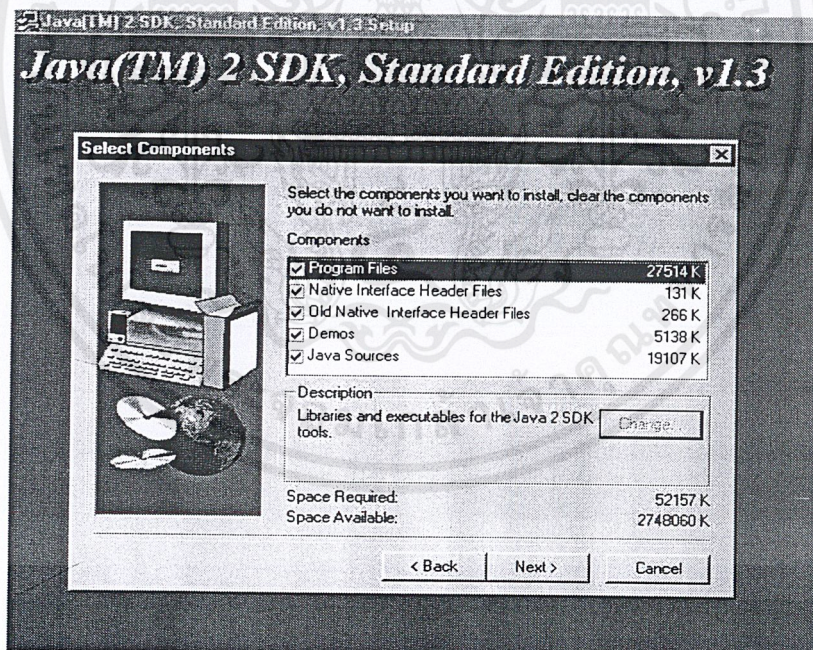
ข.2.1 การดาวน์โหลดและติดตั้ง Java Development Kit (JDK) 1.3

ดาวน์โหลดไฟล์ติดตั้ง โดยเลือกใช้ JDK เวอร์ชัน 1.3 สำหรับวินโดว์ ชื่อ "j2sdk1_3_0win.exe" ในกรณีที่เครื่องคอมพิวเตอร์เป็นแพลตฟอร์มอื่นก็ให้เลือกไฟล์ของ JDK ตามแพลตฟอร์มที่ตรงกับเครื่องคอมพิวเตอร์มาใช้แทน การดาวน์โหลดสามารถทำได้โดย ดาวน์โหลดผ่านทางบราวเซอร์จากเว็บไซต์ของบริษัท Sun Microsystems (<http://java.sun.com/j2se/1.3/>) หรืออาจใช้บริการFTP จากบราวเซอร์ก็ได้ โดยกำหนด URL เป็น <ftp://ftp.javasoft.com>

ไฟล์ j2sdk1_3_0win.exe ที่ดาวน์โหลดได้มาจะมีไฟล์ใช้งานถูกแพ็คเก็บรวมกันอยู่ ดังนั้นจึงต้องแตกไฟล์เหล่านั้นออกมา โดยรันไฟล์ที่ได้มา จากนั้นการขยายไฟล์จะถูกทำให้โดยอัตโนมัติ แต่ถ้าหากมีการติดตั้ง JDK เวอร์ชันเก่ากว่าไว้แล้วในเครื่อง ให้ลบออกก่อนทำการขยายไฟล์

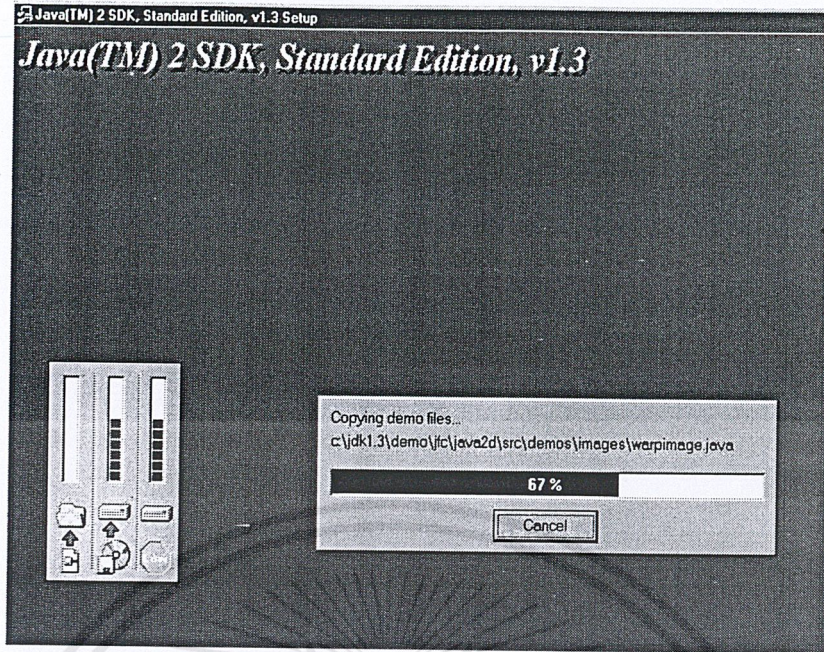


รูปที่ ข.1 แสดงขั้นตอนการระบุพาทในการติดตั้ง JDK 1.3

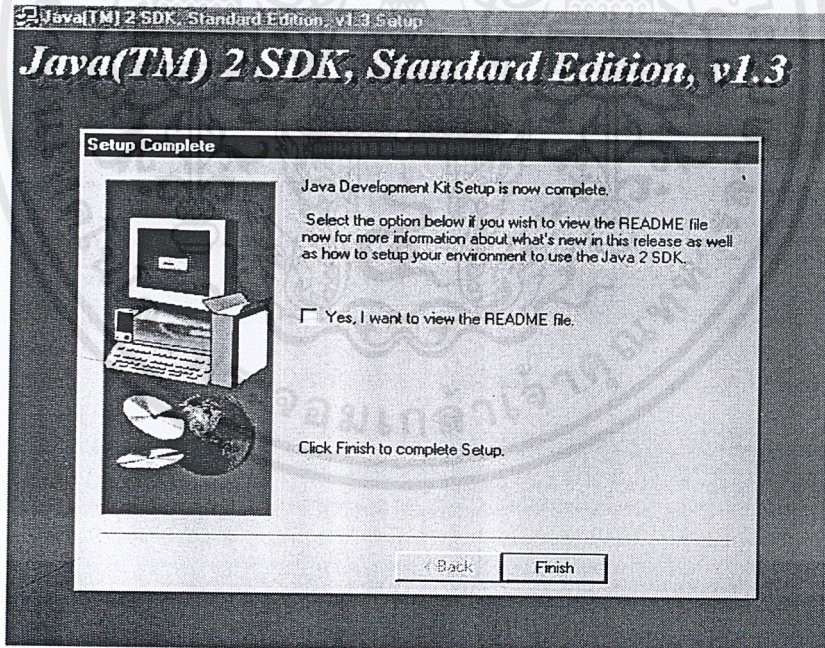


รูปที่ ข.2 แสดงขั้นตอนการเลือกคอมโพเนนต์ต่างๆ ในการติดตั้ง JDK 1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.3 แสดงกระบวนการคัดลอกไฟล์ในระหว่างการทำการติดตั้ง JDK 1.3



รูปที่ ข.4 แสดงการเสร็จสิ้นการติดตั้ง JDK 1.3

การปรับแต่งสภาพแวดล้อมให้เหมาะสม เช่น การเพิ่มเติมพาธไว้ในไฟล์ AUTOEXEC.BAT.BAT ของระบบที่ใช้งานอยู่ เพื่อให้เรียกใช้ทูลต่างๆ ของชุดพัฒนา JDK ได้จากใดแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

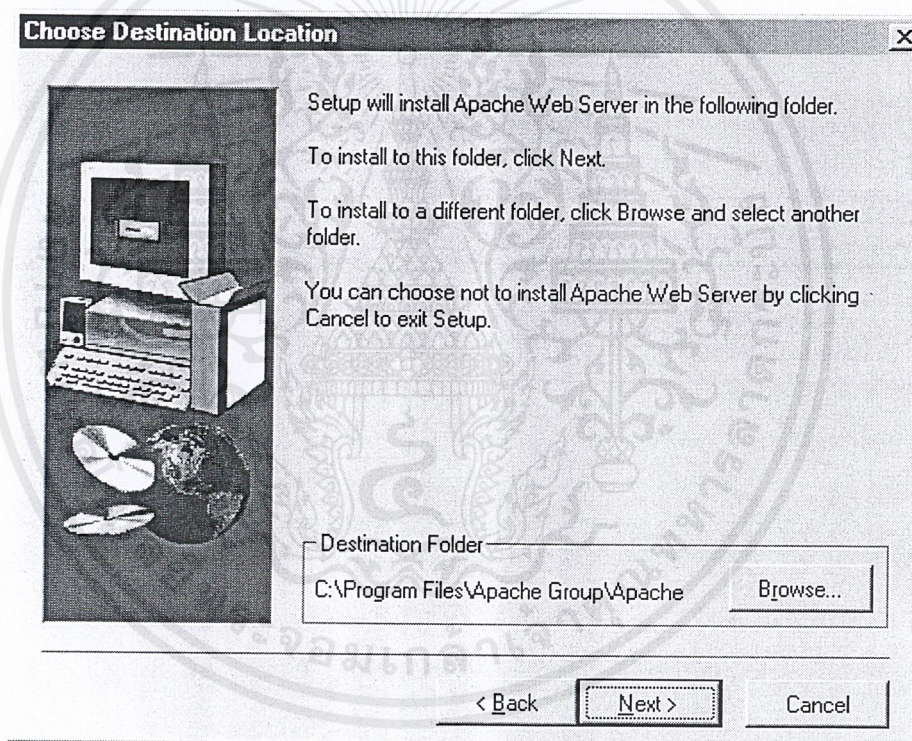
เทอร์ใดก็ได้ที่มีอยู่บนระบบ โดยทำการแก้ไข ไฟล์ AUTOEXEC.BAT.BAT โดยใช้เอดิเตอร์ใดก็ได้เปิดไฟล์นี้ขึ้นมา แล้วเข้าไปเพิ่มเติมพารชื่อ \jdk1.3\bin โดยอาจทำการเพิ่ม

- set CLASSPATH = [drive]:\[path]\jdk1.3\bin; หรือ
- ;[drive]:\[path]\jdk1.3\bin; ต่อท้ายตัวอักษรสุดท้ายในคลาสพาทที่มีอยู่เดิม

ข.2.2 การติดตั้งเว็บเซิร์ฟเวอร์

สามารถดาวน์โหลด Apache HTTP Server เวอร์ชัน 1.3.14 สำหรับวินโดวส์ได้ที่ URL ต่อไปนี้ http://httpd.apache.org/dist/httpd/binaries/win32/old/apache_1_3_14_win32_r2.exe หรือ เวอร์ชันที่ใหม่กว่าได้ที่ <http://httpd.apache.org/dist/httpd/> หลังจากนั้นให้รันไฟล์ .exe ที่ดาวน์โหลดมา ระหว่างการติดตั้งโปรแกรมจะให้เลือก

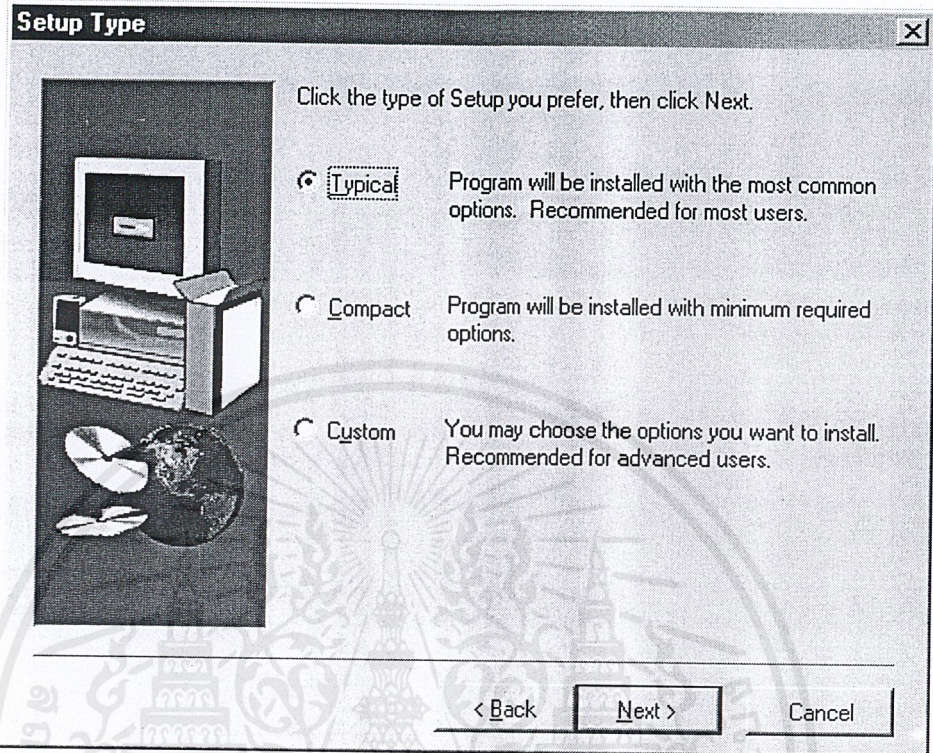
- ไดรเรททอรี (directory) ที่จะทำการติดตั้ง Apache Web Server (ค่าดีฟอลต์ได้แก่ C:\Program Files\Apache Group\Apache



รูปที่ ข.5 การติดตั้ง Apache Web Server ขั้นที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

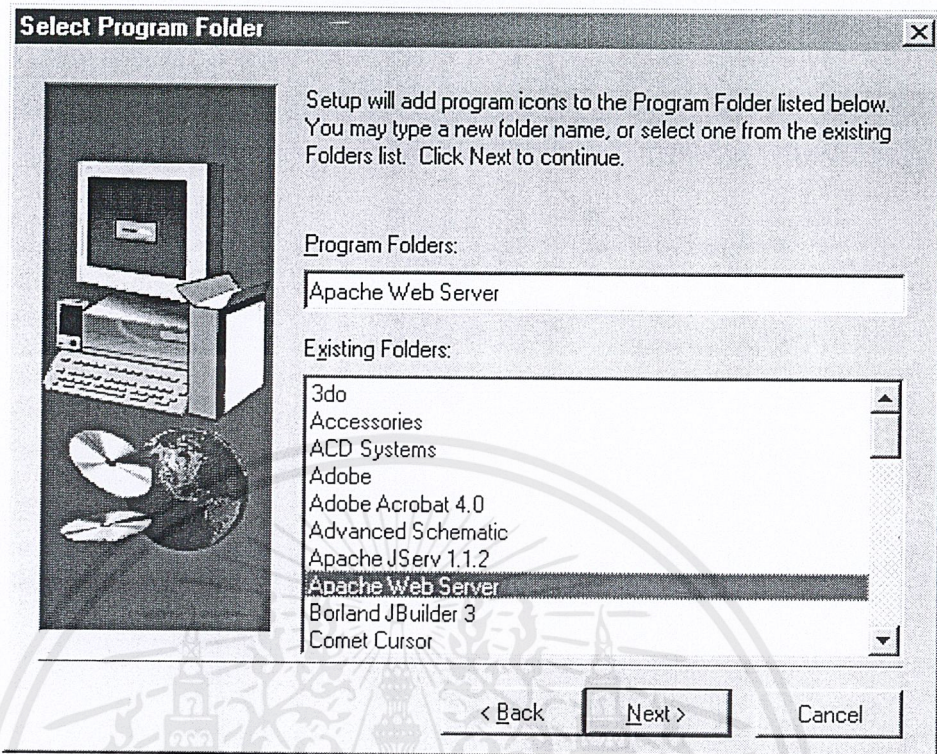
- ชนิดของการติดตั้งมี 3 แบบได้แก่ “Typical” จะติดตั้งทุกอย่างยกเว้นซอสโค้ด (source code), “Compact” ไม่ติดตั้งคู่มือและซอสโค้ด และ “Custom” จะติดตั้งซอสโค้ดให้ด้วย



รูปที่ ข.6 การติดตั้ง Apache Web Server ขั้นที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชื่อของเมนูที่จะปรากฏอยู่บน Start Programs เมนู (ค่าดีฟอลต์ได้แก่ Apache Web Server)

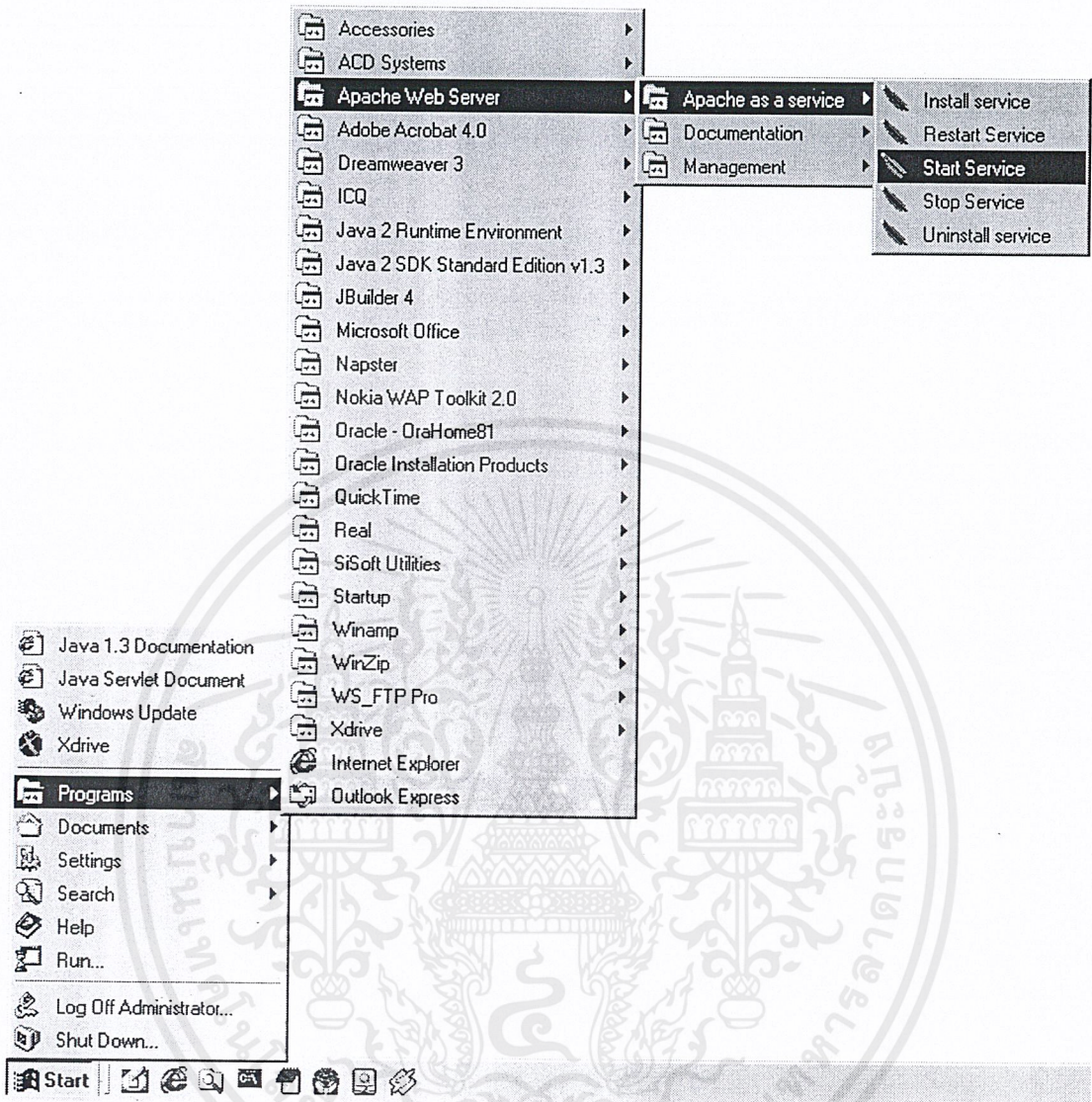


รูปที่ ข.7 การติดตั้ง Apache Web Server ขั้นที่ 3

ในกรณีที่มีการติดตั้ง Apache Web Server ข้างลงในไดเรกทอรีที่มี Apache Web Server อยู่แล้ว ไฟล์ของเดิมต่างๆ จะไม่ถูกเขียนทับ แต่ไฟล์ที่ถูกติดตั้งลงไปใหม่นั้นจะถูกเติม .default ลงไป ยกตัวอย่างเช่น มีไฟล์ http.conf อยู่แล้ว http.conf ที่ถูกติดตั้งลงไปใหม่จะถูกเปลี่ยนชื่อเป็น http.conf.default

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

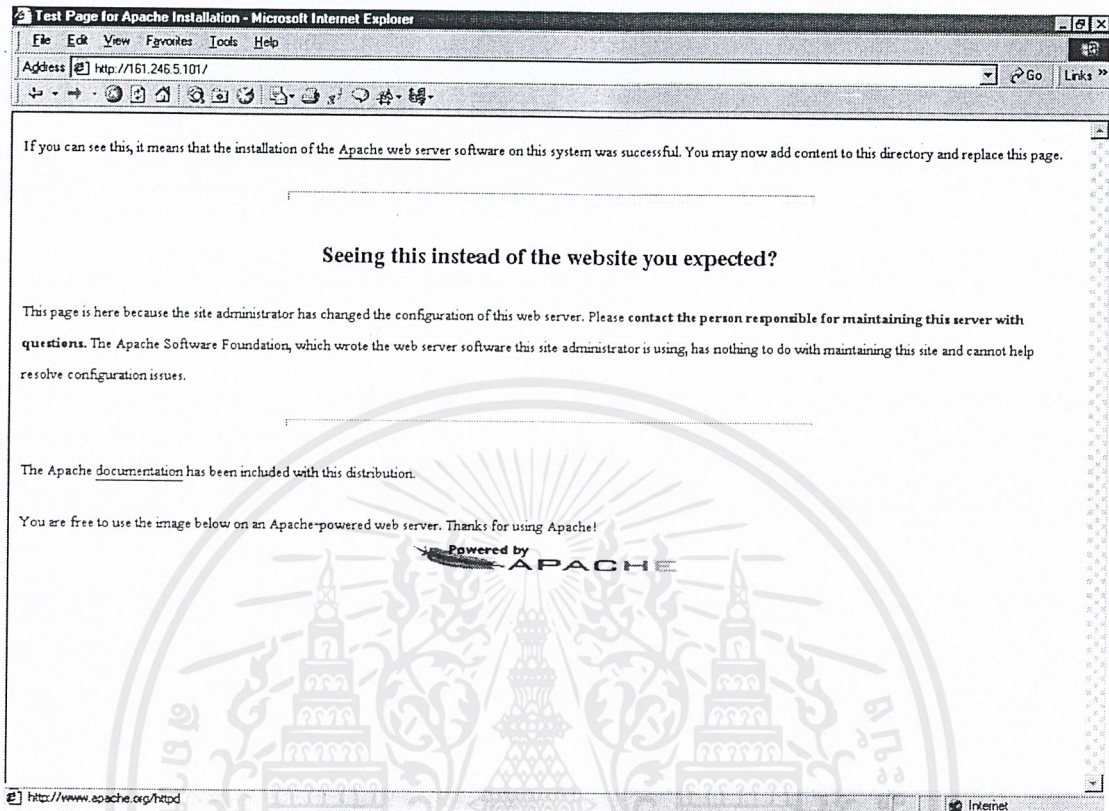
เมื่อทำการติดตั้งเรียบร้อยแล้ว ให้ start Apache ที่ Start → Programs → Apache Web Server
 → Apache as a service → Start Service ดังภาพ



รูปที่ ข.8 การรัน Apache Web Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

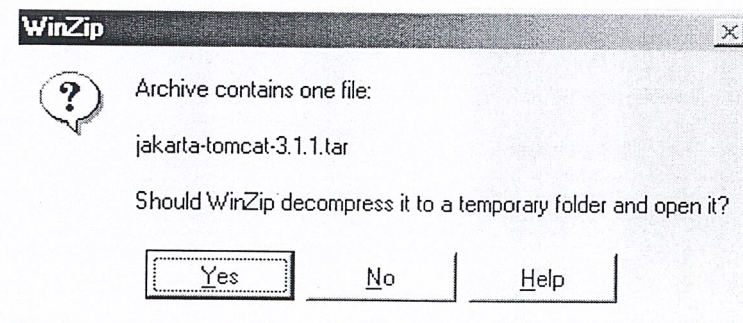
เมื่อ start service เรียบร้อยแล้วให้ทดลองเปิด IE แล้วไปที่ <http://localhost> หรือ <http://127.0.0.1> หรือ [http://\[IP address\]](http://[IP address]) ในที่นี้จะใช้ <http://161.246.5.101> เพื่อทดสอบว่า web server ที่ทำการติดตั้งใช้งานได้หรือไม่ ถ้าติดตั้งสำเร็จ IE ควรแสดงเว็บเพจดังรูป



รูปที่ ข.9 เว็บเพจแสดงให้เห็นว่าการลง Apache Web Server เสร็จสมบูรณ์

ข.2.3 การติดตั้งเซิร์ฟเวทเอนจิน

เนื่องจาก Apache Web Server ไม่รองรับจาวาเซิร์ฟเวท ดังนั้นหลังจากติดตั้งเว็บเซิร์ฟเวอร์จึงต้องทำการติดตั้ง Jakarta-Tomcat ซึ่งเป็นเซิร์ฟเวทเอนจินเพิ่มเติม Jakarta-Tomcat v3.1.1 สามารถดาวน์โหลดได้จาก <http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.1.1/bin/jakarta-tomcat-3.1.1.tar.gz> และสามารถดาวน์โหลดเวอร์ชันอื่นๆ ได้จาก <http://jakarta.apache.org/builds/> ทำการ unzip ไฟล์ที่ดาวน์โหลดมาด้วยโปรแกรม WinZip 8.0 โปรแกรมจะขึ้นไดอะล็อก (dialog) มาดังภาพ



รูปที่ ข.10 ไดอะล็อกของโปรแกรม WinZip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าตอบ Yes WinZip จะทำการ unzip ไฟล์ย่อยออกมาทั้งหมด ถ้าตอบ No WinZip จะ unzip ได้ไฟล์ jakarta-tomcat3.1.1.tar ออกมาเพียงไฟล์เดียวแล้วเราจะต้องตั้ง unzip ไฟล์ jakarta-tomcat3.1.1.tar ด้วยตัวเองอีกรอบ ในที่นี้เราจะตอบ Yes และเลือกให้ unzip ไว้ใน F:\Program_Files\ หลังการ unzip ไดรเรททอรี jakarta-tomcat จะถูกสร้างขึ้นภายใน F:\Program_Files\

จากนั้นให้ดาวน์โหลดไฟล์ ApacheModuleJServ.dll ของเวอร์ชัน 3.1.1 ได้ที่ <http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.1.1/bin/win32/i386/ApacheModuleJServ.dll> หรือของเวอร์ชันอื่น ๆ ได้ที่ <http://jakarta.apache.org/builds/jakarta-tomcat/release/> เลือกเวอร์ชันที่ต้องการแล้วลิงก์ต่อไปเรื่อยๆ เหมือน path ของเวอร์ชัน 3.1.1 นำไฟล์ที่ดาวน์โหลดมาไว้ที่ [Apache_HOME]\modules\ (โฮมไดเรททอรีของ Apache Web Server) ในคู่มือนี้จะเอาไว้ใน F:\Program_Files\Apache Group\Apache\modules\

ทำการแก้ไขไฟล์ F:\Program_Files\Apache Group\Apache\conf\http.conf ด้วยการเพิ่ม "include C:/Program_Files/jakarta-tomcat/conf/tomcat-apache.conf" ลงในบรรทัดสุดท้ายของไฟล์ ไฟล์ tomcat-apache.conf จะถูกสร้างขึ้นเองโดยอัตโนมัติเมื่อทำการ start Tomcat

```
.....
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *>
# ServerAdmin webmaster@dummy-host.example.com
# DocumentRoot /www/docs/dummy-host.example.com
# ServerName dummy-host.example.com
# ErrorLog logs/dummy-host.example.com-error_log
# CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
include C:/Program_Files/jakarta-tomcat/conf/tomcat-apache.conf
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการเซตค่าตัวแปรที่ใช้ระบุ path ของโสมไดเรกทอรีของ Tomcat และ Java ด้วยการแก้ไขไฟล์

F:\Program_Files\jakarta-tomcat\bin\tomcat.bat ดังนี้

```
@echo off
rem A batch file to start/stop tomcat server.
rem This batch file written and tested under Windows NT
rem Improvements to this file are welcome
rem Guess TOMCAT_HOME if it is not present

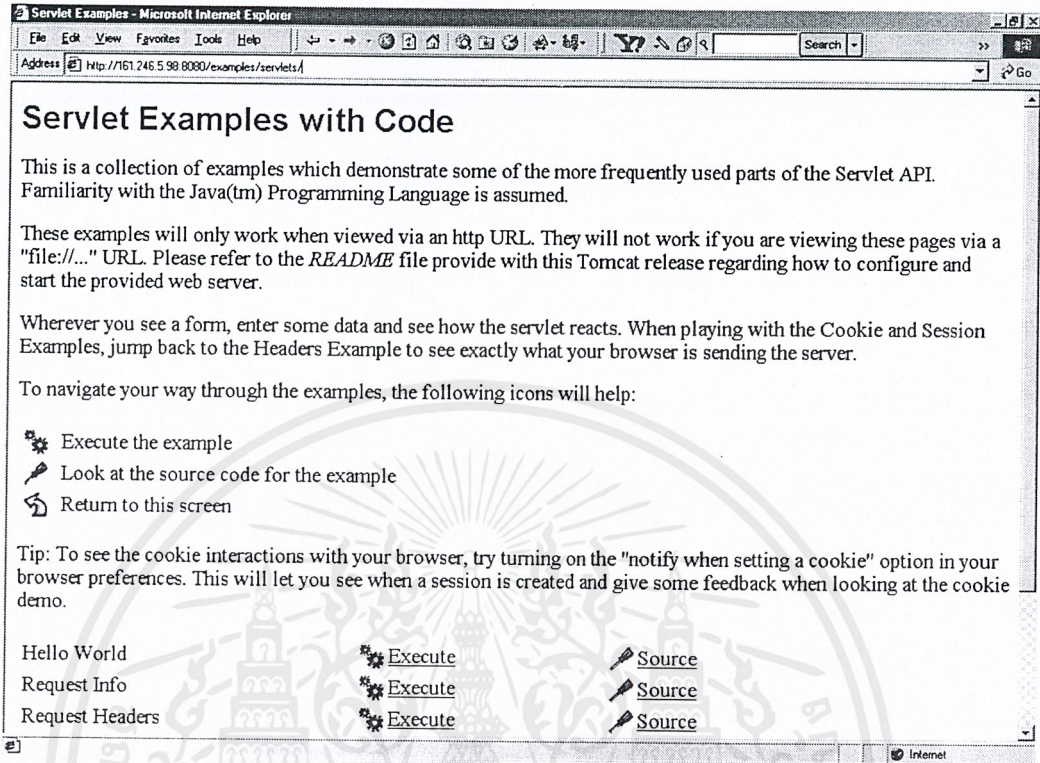
set TOMCAT_HOME=F:\Program_Files\jakarta-tomcat

set JAVA_HOME=F:\Program_Files\jdk1.3

.....
```

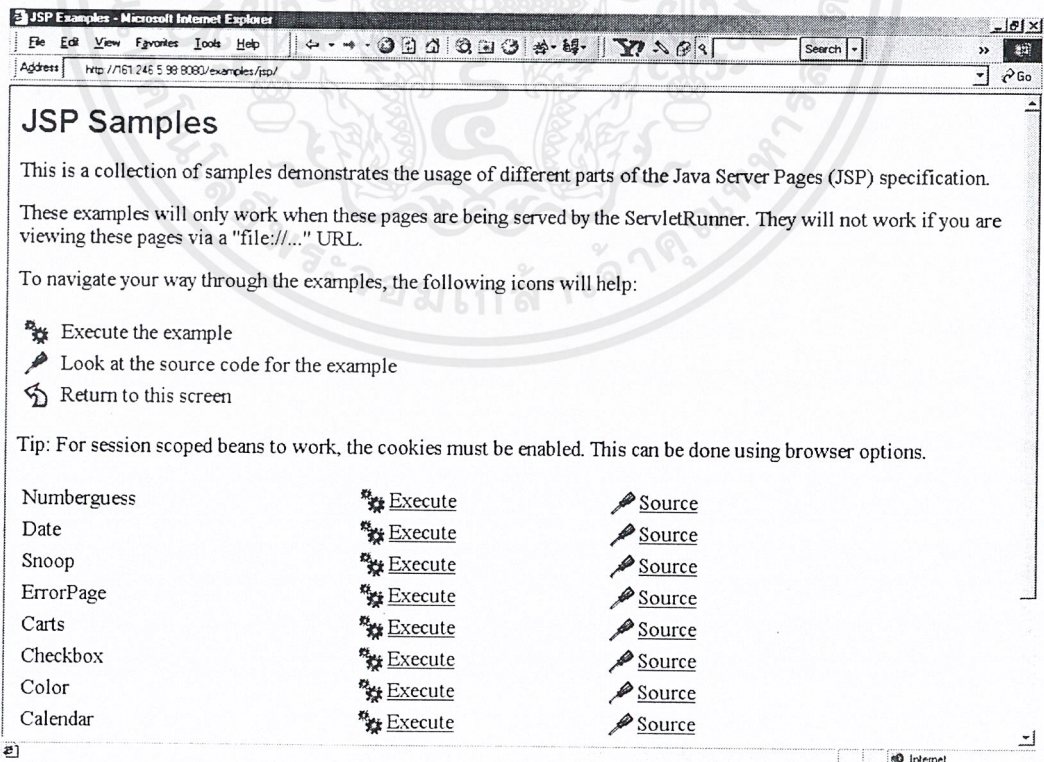
เมื่อทำการแก้ไขไฟล์ทั้งหมดเรียบร้อยแล้วให้ save แล้วรันโปรแกรม Tomcat ก่อน โดยไปที่คอมพิวเตอร์มาดไลน์ เข้ามาที่ [TOMCAT_HOME]\bin\ แล้วพิมพ์ startup เมื่อรัน Tomcat เรียบร้อย จึงค่อยทำการ start service ที่ start menu เน้นว่าจำเป็นอย่างยั้งที่ต้องทำการรัน Tomcat ก่อน Apache เพื่อให้เว็บเซิร์ฟเวอร์สามารถทำงานได้ถูกต้อง

หลังจากติดตั้งตามขั้นตอนแล้วทดสอบว่าการติดตั้ง Tomcat ประสบผลสำเร็จหรือไม่ด้วยการเปิด IE แล้วเข้ามาที่ URL `http://<localhost>:8080/examples/servlets/` เพื่อเป็นการทดสอบว่าสามารถใช้งาน เซิร์ฟเลตบนเว็บเซิร์ฟเวอร์ได้หรือไม่ จะได้หน้าจอดังนี้



รูปที่ ข.11 เว็บเพจแสดงการใช้งานเซิร์ฟเลตบน Tomcat

และทดสอบการใช้งาน JSP ด้วยการเข้า URL `http://<localhost>:8080/examples/jsp/` จะได้หน้าจอ



รูปที่ ข.12 เว็บเพจแสดงการใช้งาน JSP บน Tomcat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดสอบนั้นไม่ต้องคัดลอกไคลเรทอรี F:\Program_Files\jakarta-tomcat\webapps\examples มาไว้ที่ F:\Program_Files\Apache Group\Apache\htdocs เนื่องจากพอร์ต 8080 ที่ปรากฏอยู่บน URL นั้น ไฟล์ที่อยู่ในไคลเรทอรี F:\Program_Files\jakarta-tomcat\webapps\examples จะถูกอ้างอิงอย่างอัตโนมัติด้วย Jakarta-Tomcat

ข.2.4 การติดตั้ง Cocoon

สามารถดาวน์โหลด Cocoon เวอร์ชัน 1.8 ได้ที่ <http://xml.apache.org/dist/cocoon/Cocoon-1.8.zip> ทำการ unzip ไฟล์ ในที่นี้จะให้ unzip ใส่วิวที่ F:\Program_Files\cocoon-1.8\ จากนั้นก็อปไฟล์ *.jar ทั้งหมดที่อยู่ใน F:\Program_Files\cocoon-1.8\lib\ ลงใน F:\Program_Files\jakarta-tomcat\lib\ และ F:\Program_Files\cocoon-1.8\bin\cocoon.jar มาไว้ที่ F:\Program_Files\jakarta-tomcat\lib\ เช่นเดียวกัน จากนั้นเซต classpath ในไฟล์ F:\Program_Files\jakarta-tomcat\bin\tomcat.bat ดังภาพ

```

.....
set CLASSPATH=.
set CLASSPATH=%TOMCAT_HOME%\classes

set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\ant_1_1.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\cocoon.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\bsf.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\bsfengines.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\jop_0_13_0.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\servlet_2_2.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\stylebook-1.0-b2.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\turbine-pool.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\xalan_1_2_D02.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\xerces_1_2.jar

set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\webserver.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\jasper.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\xml.jar
set CLASSPATH=%CLASSPATH%;%TOMCAT_HOME%\lib\servlet.jar
set CLASSPATH=%CLASSPATH%;%JAVA_HOME%\lib\tools.jar
.....

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเซต classpath ของ Cocoon ต้องเอาไว้ก่อนหน้า Tomcat ของเดิมที่มีอยู่แล้ว จึงจะสามารถใช้งาน Cocoon ได้

แก้ไขไฟล์ F:\Program_Files\jakarta-tomcat\conf\web.xml เพื่อเพิ่มเติมคำสั่งสำหรับการคอนฟิก Cocoon และเพิ่ม mime-type สำหรับเว็บลงไป ดังรูป

```

.....
<session-config>
  <session-timeout>
    30
  </session-timeout>
</session-config>

<servlet>
  <servlet-name>org.apache.cocoon.Cocoon</servlet-name>
  <servlet-class>org.apache.cocoon.Cocoon</servlet-class>
  <init-param>
    <param-name>properties</param-name>
    <param-value>
      cocoon.properties
    </param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>org.apache.cocoon.Cocoon</servlet-name>
  <url-pattern>*.xml</url-pattern>
</servlet-mapping>

<mime-mapping>
  <extension>
    wml
  </extension>
  <mime-type>
    text/vnd.wap.wml
  </mime-type>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</mime-mapping>
<mime-mapping>
  <extension>
    wmlc
  </extension>
  <mime-type>
    application/vnd.wap.wmlc
  </mime-type>
</mime-mapping>
<mime-mapping>
  <extension>
    wmls
  </extension>
  <mime-type>
    text/vnd.wap.wmlscript
  </mime-type>
</mime-mapping>
<mime-mapping>
  <extension>
    wmlsc
  </extension>
  <mime-type>
    application/vnd.wap.wmlscriptc
  </mime-type>
</mime-mapping>
<mime-mapping>
  <extension>
    wbmp
  </extension>
  <mime-type>
    image/vnd.wap.wbmp
  </mime-type>
</mime-mapping>
<mime-mapping>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<extension>
  txt
</extension>
<mime-type>
  text/plain
</mime-type>
</mime-mapping>
.....

```

จากนั้นทำการก๊อปปี้ไฟล์ F:\Program_Files\cocoon-1.8\conf\cocoon.properties มาไว้ที่ F:\Program_Files\jakarta-tomcat\webapps\cocoon\WEB-INF\ ซึ่งต้องสร้างขึ้นใหม่ แก้ไขไฟล์ F:\Program_Files\jakarta-tomcat\conf\server.xml ด้วยการเพิ่มคำสั่งดังรูป

```

.....
<Context path="/test" docBase="webapps/test" debug="0" reloadable="true" >
</Context>

<Context path="/cocoon" docBase="webapps/cocoon" debug="0" reloadable="true" >
</Context>

</ContextManager>
</Server>

```

แก้ไขไฟล์ F:\Program_Files\jakarta-tomcat\conf\tomcat-apache.conf เพิ่มคำสั่ง

```

.....
<Location /examples/WEB-INF/ >
  AllowOverride None
  - deny from all
</Location>

Alias /cocoon C:\success\jakarta-tomcat\webapps\cocoon
<Directory "C:\success\jakarta-tomcat\webapps\cocoon">
  Options Indexes FollowSymLinks
</Directory>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ApJServMount /cocoon/servlet /cocoon

<Location /cocoon/WEB-INF/ >

    AllowOverride None

    deny from all

</Location>

Alias /test C:\success\jakarta-tomcat\webapps\test

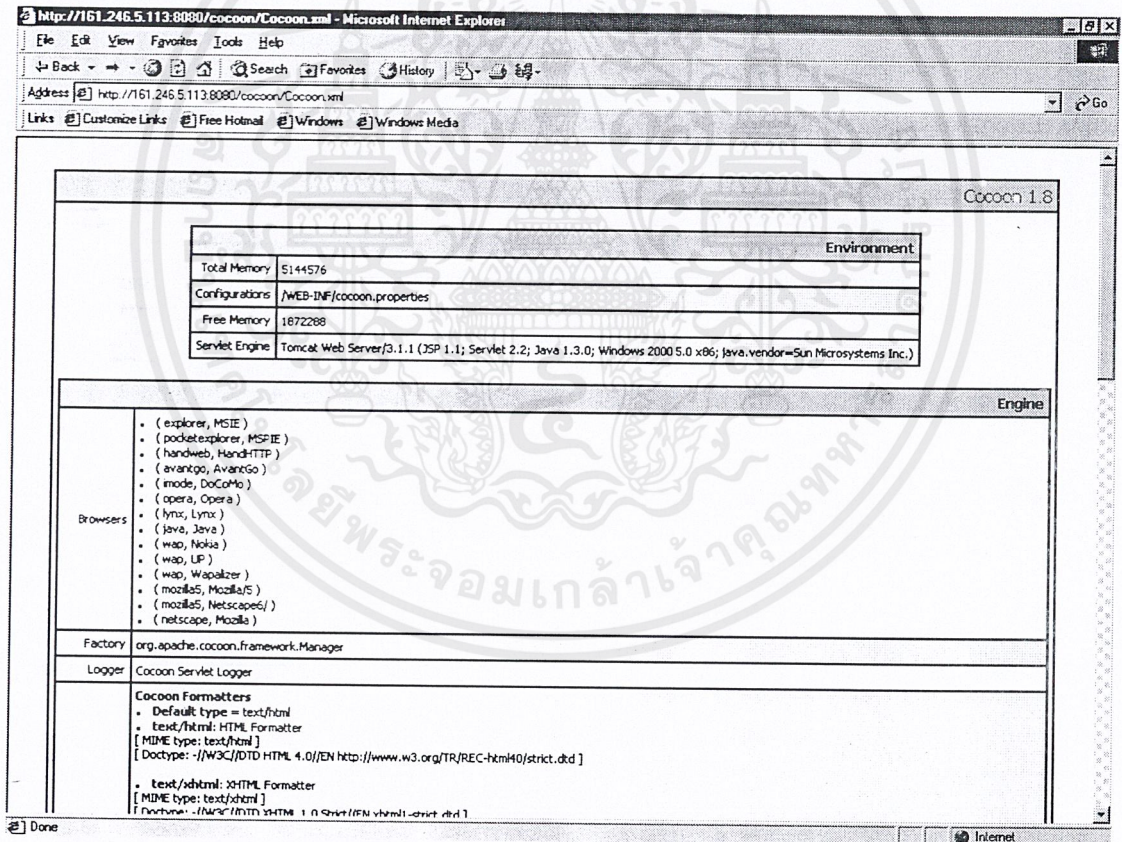
<Directory "C:\success\jakarta-tomcat\webapps\test">

    Options Indexes FollowSymLinks

</Directory>

.....
    
```

ทดสอบการลง Cocoon ว่าสำเร็จหรือไม่ด้วยการเข้า URL <http://localhost:8080/cocoon/Cocoon.xml> จะได้น้ำจืดดังนี้



รูปที่ ข.13 เว็บเพจแสดงการติดตั้ง Cocoon สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเซตระบบทั้งเรียบร้อยแล้วก็จะสามารถใช้งานตัวแปลงภาษาได้ด้วยการนำไฟล์

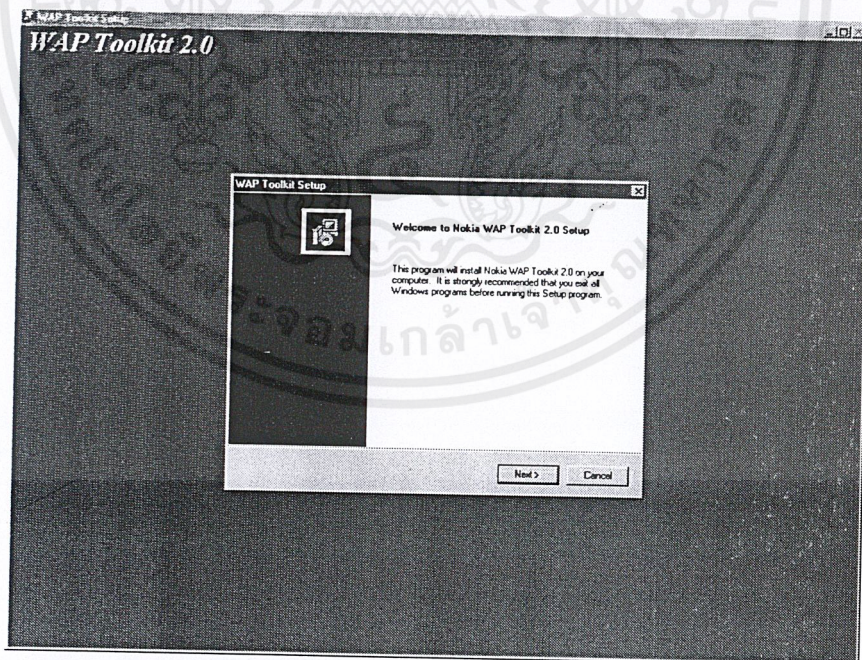
ProducerFromHTML.class	Buffer.class
HTMLExtractor.class	Service&ProcInstruct.class
Service.class	Service&TempStack
FormFilter	Type.class
RadioandCheckbox	

มาใส่ไว้ใน F:\Program_Files\cocoon-1.8\lib\com\poqit\wml\Producer\ (ให้สร้าง path ดังกล่าวขึ้นมาเอง) นำไฟล์ Wapform.class และ Handlerwap.class ไว้ใน F:\Program_Files\jakarta-tomcat\webapps\test\WEB-INF\classes\ ซึ่งมีอยู่แล้วไม่ต้องสร้าง และสุดท้ายนำไฟล์ convert2wml.wml มาใส่ไว้ใน F:\Program_Files\jakarta-tomcat\webapps\test\

ข.2.5 การดาวน์โหลดและติดตั้ง Nokia Toolkit

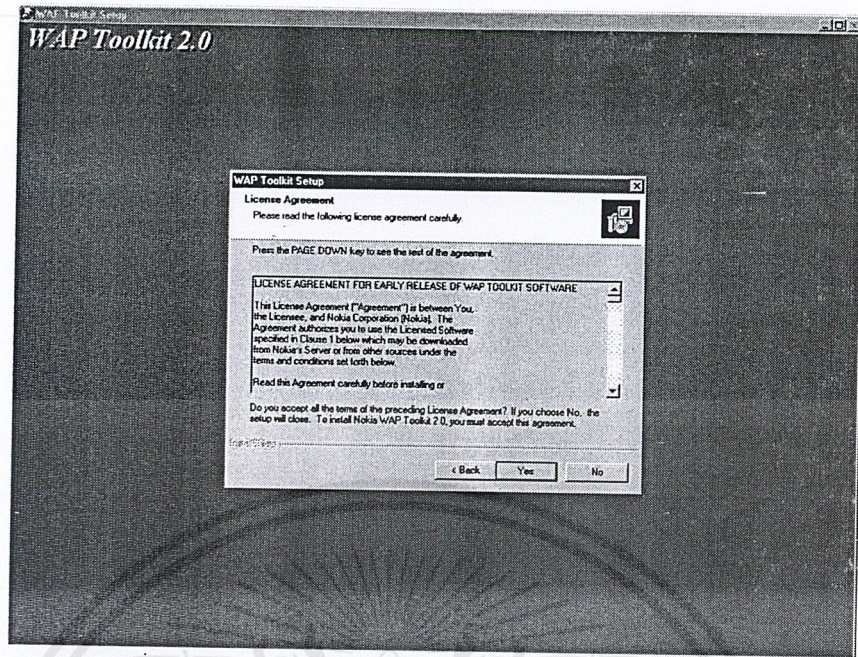
ดาวน์โหลดไฟล์ติดตั้ง จากเว็บไซต์ของบริษัทโนเกีย (www.nokia.com) โดยต้องทำการลงทะเบียนก่อนถึงจะมีสิทธิในการดาวน์โหลดโปรแกรมฟรีได้ จากนั้นจึงค่อยทำการดาวน์โหลด Nokia WAP Toolkit เวอร์ชัน 2.0 ชื่อ NokiaToolkit2_0.zip

ทำการ unzip ไฟล์ที่ถูกบีบอัดของ Nokia Toolkit ก่อน จากนั้นดับเบิลคลิกที่ไอคอน setup.exe ที่ unzip ออกมา แล้วโปรแกรมจะเริ่มทำงาน

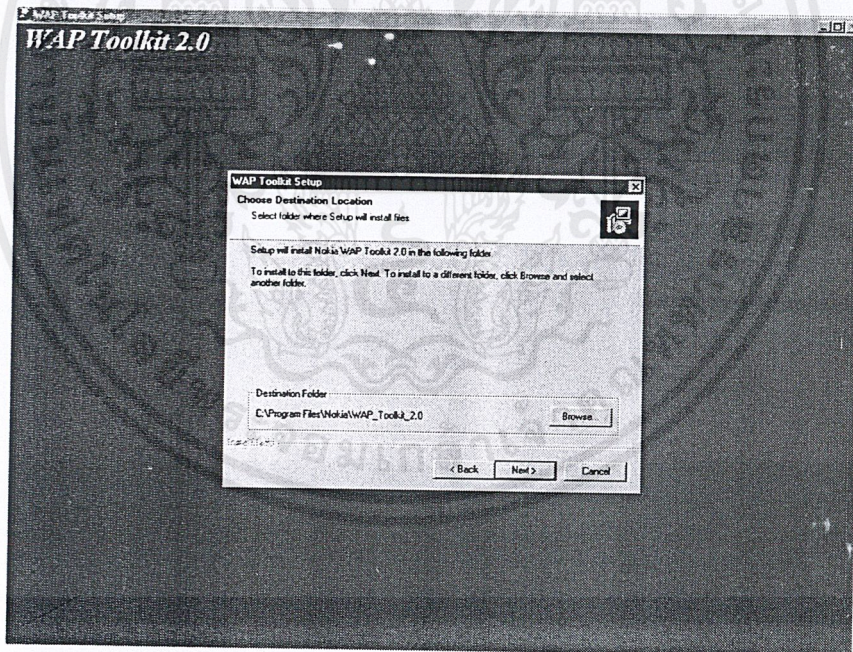


รูปที่ ข.14 แสดงหน้าต่างติดตั้ง Nokia Toolkit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

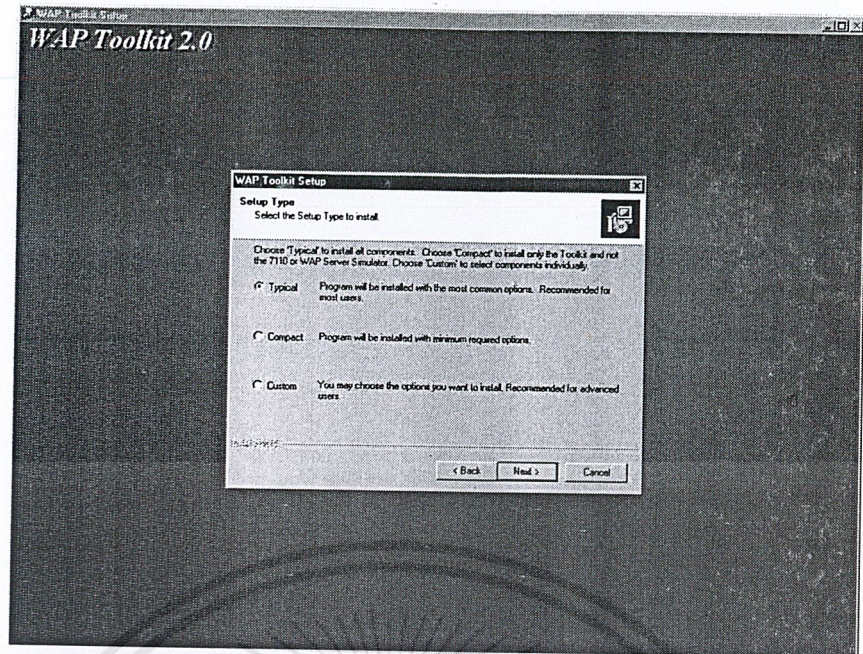


รูปที่ ข.15 แสดงข้อตกลงทางกฎหมาย

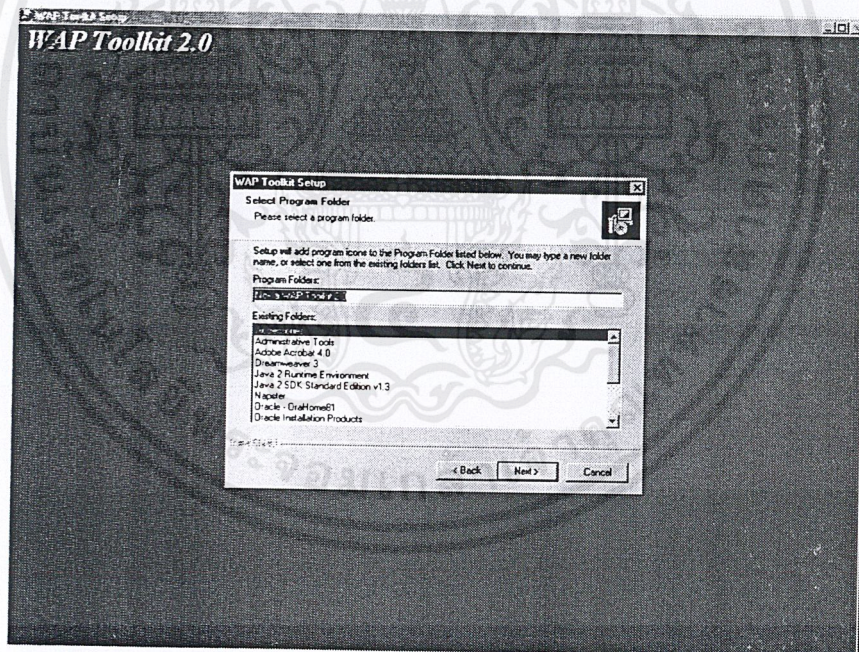


รูปที่ ข.16 แสดงขั้นตอนการระบุพาทในการติดตั้ง Nokia Toolkit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

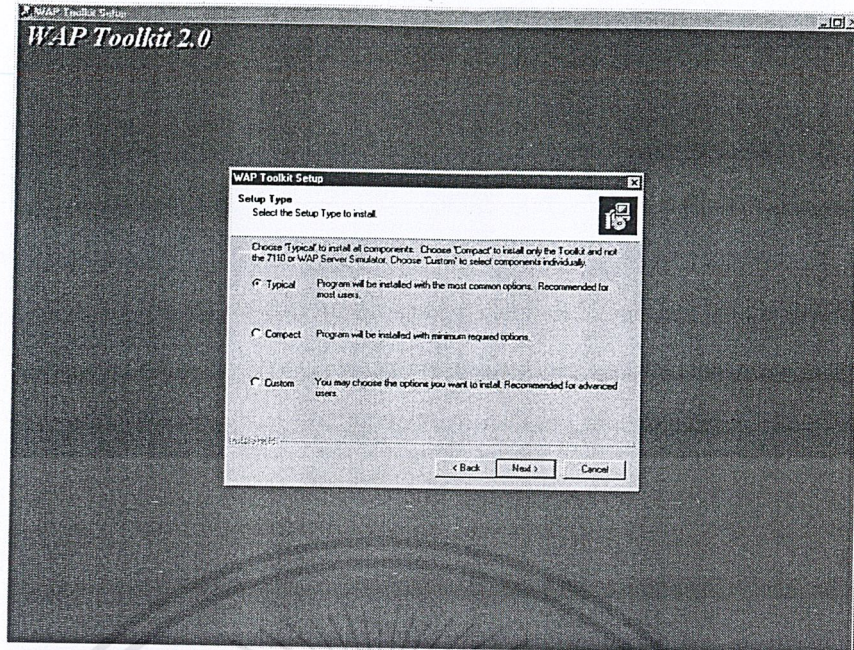


รูปที่ ข.17 วิธีติดตั้งโปรแกรม

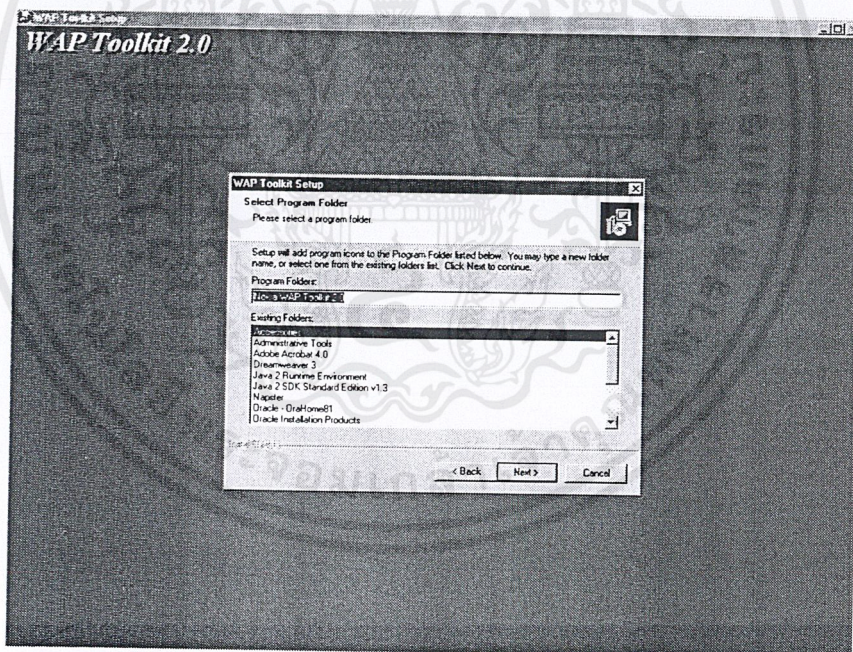


รูปที่ ข.18 สร้างกลุ่มของไอคอนบนปุ่ม Start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.19 วิธีติดตั้งโปรแกรม



รูปที่ ข.20 สร้างกลุ่มของไอคอนบนปุ่ม Start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ดาวน์โหลดโปรแกรม Java Development Kit (JDK) 1.3

"http://java.sun.com/j2se/1.3/j2sdk1_3_0win.exe" หรือ "<ftp://ftp.javasoft.com>"

ดาวน์โหลด Apache HTTP Server เวอร์ชัน 1.3.14 สำหรับวินโดวส์

"http://httpd.apache.org/dist/httpd/binaries/win32/old/apache_1_3_14_win32_r2.exe" หรือ
เวอร์ชันที่ใหม่กว่าได้ที่ "<http://httpd.apache.org/dist/httpd/>"

ดาวน์โหลด Jakarta-Tomcat v3.1.1

"<http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.1.1/bin/jakarta-tomcat-3.1.1.tar.gz>"
และสามารถดาวน์โหลดเวอร์ชันอื่นๆ ได้จาก "<http://jakarta.apache.org/builds/>"

ดาวน์โหลดไฟล์ ApacheModuleJserv.dll ของเวอร์ชัน 3.1.1

"<http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.1.1/bin/win32/i386/ApacheModuleJserv.dll>"

หรือของเวอร์ชันอื่น ๆ ได้ที่ "<http://jakarta.apache.org/builds/jakarta-tomcat/release/>"

ดาวน์โหลด Cocoon เวอร์ชัน 1.8

"<http://xml.apache.org/dist/cocoon/Cocoon-1.8.zip>"

ดาวน์โหลด Nokia Toolkit

"<http://www.nokia.com>"