



โปรแกรมนำเสนอภาพและตัวอักษรบนไมโครคอมพิวเตอร์
(IMAGE AND FONT PRESENTATION ON MICROCOMPUTER)



วัน เดือน ปี. 1 ๑๐ ๒๕๒๒
เลขทะเบียน... ๐๓๗๑๕๓
เลขเรียกหนังสือ... T ๐๓๖๒๔๖-๒๕๖ ๒

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาคตามหลักสูตรปริญญาวิศวกรรมบัณฑิต
สาขาวิศวกรรมการวัดคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา ๒๕๓๘

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสาร 037153 นำไปใช้

ปริญญาโทปีการศึกษา 2538

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมนำเสนอภาพและตัวอักษรบนไมโครคอมพิวเตอร์

(IMAGE AND FONT PRESENTATION ON MICROCOMPUTER)

ผู้จัดทำ

นาย รัฐพล กษมากรณ์ 36012112

นาย สุริยันต์ คงคำ 36012138


..... อาจารย์ที่ปรึกษา
(อาจารย์ ทรงชัย วีระทวีมาศ)

..... อาจารย์ที่ปรึกษา
(อาจารย์ อรัญญา ปรีชาไว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนำเสนอภาพและตัวอักษรบนไมโครคอมพิวเตอร์
(IMAGE AND FONT PRESENTATION ON MICROCOMPUTER)

นาย รัฐพล กษมากรณ์
นาย สุริยันต์ คงคำ

อาจารย์ ทรงชัย วีระทวีมาศ
อาจารย์ อรัญญา ปรีชาไว
อาจารย์ที่ปรึกษา
ปีการศึกษา 2538

บทคัดย่อ

โครงการสำหรับนำเสนอภาพและตัวอักษรบนไมโครคอมพิวเตอร์นี้ จะทำการนำเสนอบนโปรแกรมวินโดว ซึ่งวินโดวนี้มีความสามารถในการนำเสนอตัวอักษรได้หลากหลายเท่าที่ ต้องการ โดยโปรแกรมนำเสนอภาพและตัวอักษรนี้จะแบ่งเป็นสองส่วนใหญ่ คือ ส่วนของภาพ และ ส่วนของตัวอักษร ในส่วนของภาพจะมีลักษณะการนำเสนอในแบบต่างๆ ไปคือเป็นลักษณะ บิตแมพโดยอาศัยคุณสมบัติในด้านกราฟิกของวินโดวเข้ามาช่วย และในส่วนของตัวอักษรนั้นจะ ใช้คุณสมบัติของตัวอักษรในวินโดวที่เรียกว่า "TrueType font" ซึ่งสามารถนำเสนอรูปแบบของ ตัวอักษรได้หลากหลายรูปแบบ และ ไม่ผิดเพี้ยนเมื่อขนาดของตัวอักษรเปลี่ยนแปลงตามต้องการ แล้วนำส่วนของภาพและตัวอักษรนี้มารวมกันเพื่อนำเสนอพร้อมๆกัน โดยลักษณะการใส่ข้อมูล เพื่อกำหนดรูปแบบและลำดับของการแสดงนั้นถูกเขียนขึ้นเพื่อให้ง่ายและสะดวกต่อการใช้งาน โดยสามารถกำหนดสิ่งต่างๆที่ต้องการได้ในหน้าเดียว และสามารถแก้ไข ลบ หรือเพิ่มลำดับการ แสดงได้ในทันที

IMAGE AND FONT PRESENTATION ON MICROCOMPUTER

Ratthapol Kasamakorn

Suriyun Khongdum

Advisor

Songchai Weerataweemas

Arunyu Preechawai

1995

Abstract

The project presents for images and fonts on microcomputer. It's program operate on windows system which changes many fonts and size. By this program devided images and fonts, the images is bitmap by characteristic of graphic windows and the fonts use. The type font of windows system which changes fonts and size by not error. Data input to program is model and queue display is easy by user interfate in the screen.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

| | หน้า |
|---|------|
| บทนำ | |
| บทที่ 1 การเขียนโปรแกรมบน Windows | 1 |
| 1.1 ลักษณะของการเขียนโปรแกรมบน Windows | 1 |
| 1.2 การเขียนโปรแกรมเพื่อสร้าง Windows | 3 |
| 1.3 การสร้างกรอบข้อความ | 12 |
| 1.4 การสร้างเมนู | 14 |
| 1.5 การสร้างไฟล์ Include | 17 |
| 1.6 การสร้างไฟล์ Module-Definition | 18 |
| บทที่ 2 รูปภาพบิตแมพ(Bitmap)และการแสดงผล | 19 |
| 2.1 การสร้างรูปภาพบิตแมพ | 19 |
| 2.2 การแสดงรูปบิตแมพ | 27 |
| 2.3 การใส่สีให้กับรูปบิตแมพที่เป็นสีเดียว(ขาว-ดำ) | 32 |
| 2.4 การลบรูปบิตแมพ | 32 |
| 2.5 คอนเท็กซ์ออปกรณ์ | 33 |
| 2.6 สี | 34 |
| 2.7 งานสี | 34 |
| บทที่ 3 ตัวอักษร(Font) | 36 |
| 3.1 ส่วนประกอบของตัวอักษร | 36 |
| 3.2 หน่วยและการวัดตัวอักษร | 37 |
| 3.3 ตัวอักษรใน Windows | 40 |
| 3.4 ลักษณะเฉพาะของข้อความและตัวอักษร | 43 |
| 3.5 เทคโนโลยีของตัวอักษรแบบทรู ไทยปี | 46 |
| 3.6 การใช้ตัวอักษรบนแอปพลิเคชัน | 46 |
| 3.7 โครงสร้างและฟังก์ชันของตัวอักษรแบบทรู ไทยปี | 56 |
| บทที่ 4 ฟังก์ชันและโครงสร้างข้อมูลของ Windows ที่ใช้ในโปรแกรม | 58 |
| 4.1 ฟังก์ชันที่ใช้ในโปรแกรม | 58 |
| 4.2 โครงสร้างข้อมูลที่ใช้ในการเขียนโปรแกรม | 67 |

| | |
|--|------|
| | หน้า |
| บทที่ 5 บล็อกโคอะแกรมและการใช้งานโปรแกรม | 78 |
| 5.1 บล็อกโคอะแกรมของโปรแกรม | 78 |
| 5.2 การใช้งานโปรแกรม | 80 |
| บทที่ 6 สรุปผลและวิจารณ์ | 84 |
| ภาคผนวก | |

- เพิ่มข้อมูล IFPRESEN.C
- เพิ่มข้อมูล IFPRESEN.RC
- เพิ่มข้อมูล IFPRESEN.H
- เพิ่มข้อมูล IFPRESEN.DEF
- เพิ่มข้อมูล INIT.C
- เพิ่มข้อมูล ERROR.C

กิตติกรรมประกาศ
หนังสืออ้างอิง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ไม่ว่าจะเป็นด้าน การติดต่อสื่อสาร ด้านการศึกษา หรือแม้แค่การนำคอมพิวเตอร์มาใช้ในด้านการนำเสนอหรือแจ้งข่าว สารคั้งที่เห็นกันอยู่ทั่วไป

ในส่วนของตัวคอมพิวเตอร์เองนั้นก็ ได้มีการพัฒนาขึ้นมาเรื่อยๆเป็นลำดับไปทั้งในด้านของฮาร์ดแวร์ เช่น ความเร็ว หรือความสามารถในการเก็บข้อมูล และซอฟต์แวร์ ซึ่งได้มีการพัฒนาโปรแกรมเพื่อสนับสนุนในส่วนของฮาร์ดแวร์ด้วย ในส่วนของซอฟต์แวร์นี้ก็ ได้มีการพัฒนาโปรแกรมเรื่อยๆมา จนในปัจจุบันได้มีโปรแกรมที่พัฒนาจนได้รับความนิยมกันอย่าง มากได้แก่ โปรแกรม Windows ซึ่ง Windows นี้เองมีความสามารถต่างๆมากมายรวมทั้งความสามารถทางด้านกราฟิกในการนำเสนอภาพในลักษณะบิตแมพที่คล่องตัว และความสามารถในการนำเสนอตัวอักษร ซึ่ง Windows ได้มีการพัฒนารูปแบบของตัวอักษรให้สามารถนำเสนอได้หลากหลายและมากเท่าที่ ต้อง อีกทั้งรูปแบบของตัวอักษรใน Windows นั้นมีความคล่องตัว และความสวยงามไม่ว่าจะกำหนดขนาดของตัวอักษรให้เล็กหรือใหญ่เพียงใด โดยตัวอักษรที่กล่าวมาของ Windows นั้นถูกเรียกว่า “ TrueType ” ซึ่งได้นำเอาคุณสมบัติที่ดีของรูปแบบตัวอักษรต่างที่มีอยู่เข้าด้วยกัน

จากความสามารถของ Windows ดังที่กล่าวมาแล้วนั้น จึงได้มีการนำคุณสมบัติที่ทั้งทางด้านกราฟิกในการนำเสนอภาพที่มีลักษณะคล่องตัวและง่ายในการนำเสนอ และทางด้านตัวอักษรที่มีความหลากหลายและสวยงามที่เรียกว่า TrueType นี้มาใช้งานร่วมกัน จึงได้ “ โปรแกรมนำเสนอภาพและตัวอักษรบนไมโครคอมพิวเตอร์ ”

บทที่ 1

การเขียนโปรแกรมบน Windows

1.1 ลักษณะของการเขียนโปรแกรมบน Windows

ในการเขียนโปรแกรมบนคอมพิวเตอร์ โปรแกรมเมอร์ภาษา C ส่วนมากมักจะติดต่อกับ I/O และหน่วยความจำโดยใช้ฟังก์ชันมาตรฐานในไลบรารีภาษา C ฟังก์ชันเหล่านี้จะมอง I/O เป็นอุปกรณ์แบบตัวหนังสือเท่านั้น ส่วนการติดต่อกับหน่วยความจำก็จะติดต่อกับโดยตรงอย่างอิสระ การกระทำเหล่านี้จะทำได้ใน Windows ที่ซึ่งแอปพลิเคชันแต่ละแอปพลิเคชันจะต้องป้อนส่วนสิ่งต่างๆใช้ร่วมกัน เช่น คีย์บอร์ด หน่วยความจำ พอร์ตต่างๆ จอภาพ หรือแม้กระทั่งตัวประมวลผลหรือ CPU โดยจะถือว่าสิ่งเหล่านี้คือทรัพยากร(resource) ที่ต้องแบ่งกันใช้

ความแตกต่างระหว่างการเขียนโปรแกรมบนคอมพิวเตอร์กับ Windows ได้แก่

1. ระบบติดต่อกับผู้ใช้ จุดประสงค์หลักอันหนึ่งของการออกแบบ Windows ก็เพื่อที่จะให้ผู้ใช้ สามารถใช้งานแอปพลิเคชันหลายๆแอปพลิเคชันพร้อมกัน รวมทั้งยังให้ผู้ใช้สามารถมองเห็นการทำงานของทุกๆแอปพลิเคชันได้พร้อมกันอีกด้วย ด้วยเหตุที่ว่าใช้งานหลายๆแอปพลิเคชันพร้อมกันนั้น แต่ละแอปพลิเคชันจึงต้องแบ่งปันจอภาพเพื่อแสดงผลต่อผู้ใช้ สำหรับระบบงานที่คล้ายกันบางระบบจะเจาะจงจอภาพทั้งจอให้แก่แอปพลิเคชันเดียวใช้งาน การแสดงของแอปพลิเคชันอื่นๆจะถูกซ่อนไว้ข้างหลัง(หรือเรียกว่า Background Process) แต่ในระบบ Windows จะแตกต่างออกไปแอปพลิเคชันที่ทำงานบน Windows ทุกๆตัว จะได้ใช้จอภาพแสดงผลกันทั่วหน้า ใหญ่บ้าง เล็กบ้างตามแต่ผู้ใช้ต้องการ และสามารถแสดงให้แก่ผู้ใช้ได้ตลอดเวลา

แต่ละแอปพลิเคชันจะใช้จอภาพแสดงผลร่วมกับแอปพลิเคชันอื่นๆ โดยใช้งานเฉพาะในส่วนที่อยู่ในกรอบสี่เหลี่ยมของคนที่เรียกว่า วินโดว์(Windows) หากมองกันผิวเผินวินโดว์จะเป็นรูปสี่เหลี่ยมที่มีกรอบธรรมดาเท่านั้น แต่ที่จริงแล้ววินโดว์ไม่เพียงแต่จะเป็นอุปกรณ์แสดงผลขั้นพื้นฐานของแอปพลิเคชันเท่านั้น แต่ยังเป็นสิ่งที่ผู้ใช้สามารถติดต่อกับแอปพลิเคชันได้เป็นอย่างดีโดยผ่านส่วนประกอบต่างๆของวินโดว์ เช่น เมนู ปุ่มบังคับ สโครลบาร์(scroll bar) เป็นต้น

เนื่องจาก Windows มีวินโดว์ให้แอปพลิเคชันหลายแบบ Windows จึงไม่สร้างวินโดว์ล่วงหน้าให้กับแอปพลิเคชันใดๆ แอปพลิเคชันเหล่านั้นจะต้องสร้างวินโดว์ของตน ในแบบที่ตนต้องการเอง ซึ่งต่างจากอดีตที่จะเตรียมจอภาพพื้นฐานให้ใช้ได้ทันที แต่ Windows ก็จะมีระบบจัดการพื้นฐานสำหรับวินโดว์อำนวยความสะดวกสำหรับผู้ใช้อยู่แล้วเช่น การเคลื่อนย้าย การเปลี่ยนขนาด การลดขนาดให้เป็นไอคอน เป็นต้น

ข้อได้เปรียบของ Windows อีกประการคือ จากการที่แต่ละแอปพลิเคชันมีวินโดว์เป็นของตนเอง จึงสามารถแน่ใจได้ว่าแอปพลิเคชันจะไม่แย่งกันใช้งานจอภาพ จนเกิดความสับสนในการใช้งาน

2. ระบบการรับข้อมูลเข้าแบบใหม่ ในระบบคอมพิวเตอร์เมื่อแอปพลิเคชันจะรับข้อมูลจากผู้ใช้ เช่น ผ่านทางฟังก์ชัน getch() ก็จำเป็นต้องรอจนกว่าผู้ใช้จะใส่ข้อมูลจริงๆเสียก่อน หากในระหว่างนั้นผู้ใช้จะใส่ข้อมูลทางอื่น เช่น เม้าส์ ก็จะไม่สามารถรับข้อมูลเข้ามาได้ แต่ใน Windows จะแตกต่างออกไป Windows จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวบรวมข้อมูลเข้าจากทุกๆอุปกรณ์ แล้วเก็บไว้ใน คิวเมสเสจ(Message Que) เมื่อแอปพลิเคชันต้องการอ่านเข้า ก็เพียงแค่อ่านจากเมสเสจคิวเท่านั้น แอปพลิเคชันจะได้รับข้อมูลจากผู้ใช้โดยไม่มีการสูญหายของข้อมูล

ข้อมูลที่อยู่ในคิวเมสเสจจะอยู่ในรูปของเมสเสจที่มีขนาดในการเก็บเท่ากันและมีรูปแบบการใช้งานคล้ายกัน ในการอ่านค่าแต่ละครั้งการพิจารณาข้อมูลก็จะใช้โปรแกรมในลักษณะที่คล้ายกัน ทำให้การเขียนแอปพลิเคชันง่ายและสะดวกขึ้น

การอ่านข้อมูลจากอุปกรณ์ต่างๆก็เช่นกัน ในระบบ Windows การอ่านค่าจากเมสเสจคิวใช้ฟังก์ชันเพียงฟังก์ชันเดียวเท่านั้น คือ GetMessage() แต่ในคอสจจะยุ่งยากมากกว่า คือหากเราต้องการอ่านค่าคีย์บอร์ดเป็นคีย์บอร์ด ก็ต้องใช้ฟังก์ชัน keypressed() การที่ต้องใช้หลายฟังก์ชันในการทำงานที่มีลักษณะคล้ายกันนั้น เมื่อเทียบกับการใช้ฟังก์ชัน GetMessage() เพียงฟังก์ชันเดียว จะเห็นได้ชัดว่าการเขียนแอปพลิเคชันบน Windows นั้นเรียบง่ายและสะดวกสบายมากกว่า

นอกจากนี้ข้อมูลที่เกิดจากผู้ใช้เลือกเมนูหรือปุ่มบังคับต่างๆ ก็ยังบรรจุเอาไว้ในคิวเมสเสจด้วย ดังนั้นการรับรู้การกระทำทั้งหมดของผู้ใช้ทุกอย่างจึงใช้เพียงฟังก์ชัน GetMessage() เพียงฟังก์ชันเดียวเท่านั้น

3. ระบบกราฟิก จากระบบคอสจที่ไม่มีการสนับสนุนระบบกราฟิกเลย มาสู่ Windows ที่เต็มไปด้วยคำสั่งและวิธีการอันหลากหลาย ซับซ้อน และมีประสิทธิภาพ สำหรับการจัดการกับระบบกราฟิกไม่ว่าจะเป็น เส้น ครก วงกลม รูปเหลี่ยม รูปทึบ หรือเส้นโค้ง ไม่เพียงเท่านั้นคำสั่งทุกคำสั่งยังสามารถใช้ได้กับอุปกรณ์แสดงผลทุกชนิดที่สามารถแสดงผลเป็นกราฟิกได้ เราสามารถใช้คำสั่งเดียวกัน รูปแบบเดียวกัน วาดรูปต่างๆได้ทั้งบนจอภาพ เครื่องพิมพ์(กราฟิก) พลอตเตอร์ ฯลฯ

Windows ได้อาศัยดีไวซ์ไดรเวอร์ในการแสดงผลให้เหมาะสมกับอุปกรณ์ที่ใช้แต่ละแบบ ดีไวซ์ไดรเวอร์ อาจเรียกว่า พื้นผิวอุปกรณ์(device context) และการแสดงผลกราฟิกทุกครั้งจะต้องกระทำผ่านพื้นผิวอุปกรณ์นี้เสมอ

4. ระบบการจัดการ ในการทำงานในคอสนั้นจะไม่มีระบบจัดการปันส่วนทรัพยากรใดๆเมื่อแอปพลิเคชันหนึ่งๆทำงานก็จะถือเสมือนว่าแอปพลิเคชันนั้นเป็นเจ้าของทรัพยากรทั้งหมด เช่น คีย์บอร์ด จอภาพ หน่วยความจำ หรือแม้กระทั่งซีพียูแต่สำหรับ Windows จะมีการบังคับให้แอปพลิเคชันมีการปันส่วนทรัพยากรต่างร่วมกันเสมอ ซึ่งต่างจากแอปพลิเคชันในคอสที่แอปพลิเคชันสามารถนำหน่วยความจำที่ไม่ได้ยึดครองโดยคอสนำไปใช้ได้ทั้งหมดโดยไม่ต้องคำนึงถึงแอปพลิเคชันอื่นๆ

แต่ใน Windows อันเป็นระบบหลายงาน(multitasking) นั้นแตกต่างออกไป พื้นที่หน่วยความจำซึ่งเป็นทรัพยากรอันจำกัดนั้นจะต้องปันส่วนกันใช้ หากทุกแอปพลิเคชันต่างใช้อย่างสิ้นเปลือง หน่วยความจำอันจำกัดก็จะหมดไปอย่างรวดเร็ว จึงทำให้แอปพลิเคชันต่างๆทำงานพร้อมกันได้ไม่มาก เพียงหนึ่งหรือสองแอปพลิเคชัน ฉะนั้นในการทำงานจริงของ Windows จึงมีการเคลื่อนย้ายสลับเปลี่ยน(swap) หรือทิ้ง(discard) ส่วนต่างๆของหน่วยความจำตลอดเวลาเพื่อให้การทำงานทั้งระบบมีประสิทธิภาพสูงสุด กานใช้หน่วยความจำแต่ละครั้งมักจะพบว่ามืออยู่หลายครั้งที่ข้อมูลที่ใส่เข้าจะไม่อยู่ที่ตำแหน่งแอดเดรสเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกตัวอย่างคือการจัดจอภาพที่จะต้องแบ่งกันใช้ในหลายๆแอปพลิเคชัน ในระบบคอสมิกจะยกจอภาพทั้งจอไป ขึ้นอยู่กับการจัดการของแอปพลิเคชันเพียงแอปพลิเคชันเดียว ไม่ว่าแอปพลิเคชันนั้นจะทำการคิดแปลงจอภาพ อย่างไร เช่นเปลี่ยนความละเอียดในการแสดงผล เปลี่ยนสีที่ใช้ในการแสดงผล ฯลฯ แต่ในระบบ Windows จะเปลี่ยนไป แอปพลิเคชันจะต้องใช้จอภาพร่วมกัน เพราะฉะนั้นแอปพลิเคชันที่ทำงานบน Windows จะต้องไม่มีการคิดแปลงการใช้งานจอภาพโดยตรงต้องมีการขอใช้โดยผ่าน Windows เสมอ

1.2 การเขียนโปรแกรมเพื่อสร้าง Windows

การเขียนโปรแกรมบน Windows นั้น แม้ว่าหน้าต่างของโปรแกรมจะเหมือนกับภาษาซีที่คุ้นเคย แต่ว่าแนวความคิด รูปแบบ และขั้นตอนการเขียนจะแตกต่างกันอย่างมาก ทั้งนี้ก็เพราะ อย่างแรก แอปพลิเคชันที่ทำงานบน Windows ที่เป็น ระบบหลายงาน (multitasking) จะไม่ได้ควบคุมการใช้ทรัพยากรของเครื่องทั้งหมดแต่เพียงผู้เดียว ตัว Windows จะเป็นผู้คอยบอกว่าแอปพลิเคชันใดจะได้ทรัพยากรใดไปใช้ การเขียนโปรแกรมจึงต้องมีการคำนึงถึงส่วนนี้มาก (เช่น การวาดหน้าจอ ตัวแอปพลิเคชันไม่สามารถวาดเองได้ ต้องมีการส่งคำสั่งไปให้ Windows เป็นผู้วาดให้ การส่งข้อมูลไปเครื่องพิมพ์ก็ต้องผ่านตัวพริ้นต์แมนเนเจอร์ เป็นต้น) อีกอย่างหนึ่งคือ ตัวฟังก์ชันต่าง ๆ ที่มีให้ใช้จะแตกต่างไปจากภาษาซีมาตรฐาน

วินโดว์แอปพลิเคชัน

ทุกแอปพลิเคชันที่เป็นแอปพลิเคชันของ Windows จะต้องมีส่วนประกอบหลักสองส่วนคือ

1. ฟังก์ชันหลัก ซึ่งจะต้องมีชื่อว่า WinMain เสมอ
2. วินโดว์ฟังก์ชัน

WinMain จะทำหน้าที่คล้ายกับฟังก์ชัน main ในภาษาซีมาตรฐาน ส่วนฟังก์ชันอื่นก็มีลักษณะกับฟังก์ชันทั่วไปของภาษาซีแต่ฟังก์ชันทั่วไปของ Windows นั้นจะต่างกับของภาษาซีตรงที่ไม่ได้ถูกเรียกใช้งาน(call)จาก WinMainโดยตรงแต่ตัว Windows จะเป็นผู้เรียกฟังก์ชันเหล่านั้นขึ้นมาทำงาน

ส่วนประกอบของ WinMain

WinMain มีหน้าที่หลัก 3 อย่างคือ

1. การเตรียมงาน (Initialize) หน้าที่ของส่วนนี้ก็เช่น การรีจิสเตอร์วินโดว์ การสร้างวินโดว์ เป็นต้น
2. การสร้าง message loop เพื่อส่งเมสเสจไปยังวินโดว์ฟังก์ชันต่าง ๆ
3. การจบแอปพลิเคชันเมื่อได้รับ WM_QUIT

ลักษณะของ WinMain เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int PASCAL WinMain (hInstance, hPrevInstance, lpCmdLine, nCmdShow)
HANDLE hInstance;          /* current instance */
HANDLE hPrevInstance;     /* previous instance */
LPSTR lpCmdLine;          /* command line */
int nCmdShow;              /* show - window type (open/icon) */
{
}

```

อาร์กิวเมนต์ที่ตามมาเป็นอาร์กิวเมนต์ที่ Windows ส่งให้กับแอปพลิเคชันแต่ละตัวซึ่งมีความหมายดังนี้

- `hInstance` เป็นค่าประจำแอปพลิเคชันนี้ ซึ่งจะแตกต่างกันไปในแต่ละแอปพลิเคชัน
- `hPreInstance` เป็นค่าแอปพลิเคชันนี้ที่ถูกเรียกขึ้นมาใช้งานก่อน (แอปพลิเคชันหนึ่งๆสามารถถูกเรียกใช้งานได้หลายครั้ง) ถ้าไม่มีการรันแอปพลิเคชันนี้มาก่อนค่านี้จะเป็น NULL
- `lpCmdLine` พอยน์เตอร์ไปยังคอมมานด์ไลน์หากว่าผู้ใช้มีการคีย์คอมมานด์ไลน์ต่อเข้ามาแอปพลิเคชันสามารถหาสตริงเหล่านั้นได้โดยใช้พอยน์เตอร์ตัวนี้
- `nCmdShow` ลักษณะของแอปพลิเคชันที่ถูกเรียกขึ้นมาใช้งานว่าจะให้ขึ้นมาเป็นวินโดว์ แบบปกติหรือเป็นลักษณะของไอคอน

หากพิจารณาส่วนหัวของ `WinMain` ใดๆ แล้วจะสังเกตเห็นชนิดข้อมูล (Data Type) นั้นมีหน้าต่างาแปลกออกไปจากภาษาซีมาตรฐานจริงๆ แล้วชนิดข้อมูลใน Windows นั้นมีมากมายหลายชนิดมาก ที่พบบ่อยก็เช่น `HANDLE` หมายถึง ตัวชี้ขนาด 16 บิตที่ใช้ชี้ข้อมูลที่ชี้แทนวัตถุต่าง ๆ (เช่น วินโดว์ หรือเครื่องมือในการวาดภาพต่าง ๆ) `HWND` `LPSTR` เป็นต้น

การขึ้นทะเบียนวินโดว์

ก่อนการสร้างวินโดว์ใด ๆ ทุกครั้งจำเป็นต้องมีการสร้างวินโดว์คลาส (window class) ของวินโดว์นั้นเสียก่อน ซึ่งวินโดว์คลาสนี้จะเป็นตัวบอกลักษณะของตัววินโดว์ เช่น ตำแหน่งขนาดลักษณะเคอร์เซอร์ลักษณะไอคอน และอื่น ๆ ซึ่ง Window จะมีวินโดว์คลาส มาตรฐานต่าง ๆ ให้ใช้อยู่แล้ว แต่ส่วนมากนิยมกำหนดวินโดว์คลาสเองมากกว่า

การสร้างวินโดว์คลาสทำได้โดยการใส่ค่าที่ต้องการเข้าไปในพารามิเตอร์ต่างๆของข้อมูลแบบโครงสร้าง (struct) ที่ชื่อว่า `WNDCLASS` จากนั้นจึงส่งค่าเหล่านี้ไปให้ Windows เพื่อบอก Windows ให้รู้จักคลาสที่ถูกกำหนดขึ้นใหม่นี้ (โดยใช้ฟังก์ชัน `RegisterClass`) และเรียกวิธีการนี้ว่า "การขึ้นทะเบียนวินโดว์"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BOOL InitApplication(hInstance)
HANDLE hInstance;
{
1. WNDCLASS wc;
    /* Fill in window class structure with parameters that describe the main window. */
2. wc.style = NULL;
3. wc.lpfnWndProc = MainWndProc;      /* Function to retrieve messages for window */
    /* of this class. */
4. wc.cbClsExtra = 0;                /* No per - class extra data. */
    wc.cbWndExtra = 0;                /* No per - window extra data.*/
5. wc.hInstance = hInstance;         /* Application that owns the class. */
6. wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
7. wc.hCursor = LoadCursor(NULL, IDC_ARROW);
8. wc.hbrBackground = GetStockObject(WHITE_BRUSH);
9. wc.lpszMenuName = "GenericMenu"; /* Name of menu resource in RC file. */
10. wc.lpszClassName = "GenericWclass"; /* Name used in call to CreateWindow. */
    /* Register the window class and return success/failure code. */
    return (RegisterClass(&wc));
}

```

จากตัวอย่างโปรแกรมข้างต้น ส่วนต่าง ๆ มีความหมายดังนี้

1. ให้ `wc` มีโครงสร้างข้อมูลเป็นแบบ `WNDCLASS`
2. `style` บอกสไตล์ต่าง ๆ ของวินโดว์ เช่น มีการวาดใหม่หรือไม่เมื่อมีการเลื่อนหรือขยายวินโดว์ ซึ่งในที่นี้มีค่าเป็น `NULL`
3. `lpfnWndProc` เป็นพอยน์เตอร์ไปยังส่วนที่เป็นวินโดว์ฟังก์ชันหลักที่ใช้รับเมสเสจจากฟังก์ชันหลัก ในที่นี้ชื่อว่า `MainWndProc`
4. `cbClsExtra` บอกจำนวนไบต์เพิ่มเติมสำหรับโครงสร้างข้อมูลของคลาส ส่วน `cbWndExtra` จะบอกจำนวนไบต์เพิ่มเติมในโครงสร้างของวินโดว์ที่ใช้คลาสนี้ ซึ่งทั้งสองตัวนี้ถูกเซตให้เป็น 0
5. `hInstance` ให้ค่า `Instance` ของแอปพลิเคชันที่ใช้คลาสนี้ในที่นี้ก็จะเป็น `hInstance`
6. `hIcon` กำหนดไอคอนของแอปพลิเคชันนี้ ซึ่งจะถูกนำมาใช้เมื่อวินโดว์ถูกลด ขนาดในที่นี้ใช้ไอคอนมาตรฐานที่สร้างไว้ใน Windows อยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. `hCursor` กำหนดรูปร่างของเคอร์เซอร์เมื่อตัวชี้เมาส์มาตกอยู่ในวินโดวนี้ ในที่นี้ใช้เคอร์เซอร์มาตรฐาน
8. `hbrBackground` กำหนดสีของแปรงที่จะใช้ระบายสีแบ็คกราวด์ ในที่นี้ใช้แปรงมาตรฐานที่เป็นสีขาว
9. `IpszMenuName` แสดงชื่อเมนูในไฟล์ resource ที่ถูกเรียกใช้
10. `IpszClassName` ชื่อของวินโดวคลาส ในที่นี้ชื่อ `GenericWClass`

การสร้างวินโดว

เมื่อมีการรีจิสเตอร์วินโดวเรียบร้อยแล้วเราก็สามารถสร้างวินโดวโดยใช้คลาสที่สร้างไว้ได้ทันทีโดยใช้ฟังก์ชัน `CreateWindow` ซึ่งเป็นคำสั่งที่แทบทุกแอปพลิเคชันต้องมีพารามิเตอร์ต่าง ๆ ที่ต้องให้กับฟังก์ชันนี้ มีดังนี้

```

/* Create a main window for this application instance. */

hWnd = CreateWindow(
1. "GenericWClass", /* See Register Class() call. */
2. "Generic Sample Application", /* Text for window title bar. */
3. WS_OVERLAPPEDWINDOW, /* Window style. */
4. CW_USEDEFAULT, /* Default horizontal position. */
   CW_USEDEFAULT, /* Default vertical position. */
   CW_USEDEFAULT, /* Default width. */
   CW_USEDEFAULT, /* Default height. */
5. NULL, /* Overlapped windows have no parent. */
6. NULL, /* Use the window class menu. */
7. hInstance, /* This instance owns this window. */
8. NULL /* Pointer to needed. */
);

```

จากโปรแกรมข้างต้นมีรายละเอียดดังนี้

1. ชื่อของวินโดวคลาสที่วินโดวนี้ใช้ ในที่นี้คือ `GenericWClass`
 2. ข้อความที่จะให้ปรากฏตรงไตเติลบาร์ของวินโดว
- ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สไตล์ของวินโดว์ที่สร้าง ในที่นี้เป็นNON_OVERLAPPEDWINDOW

4. พารามิเตอร์ต่อไปอีก 4 ตัวจะเป็นตำแหน่งแกน X ตำแหน่งแกน Y ความกว้างและความสูงของวินโดว์ ซึ่งสามารถกำหนดเป็นตัวเลขที่แน่นอนลงไปได้ แต่หากใช้ CW_USEDEFAULT จะเป็นการใช้ค่าที่กำหนดไว้ล่วงหน้าแล้ว

5. บอกว่าวินโดว์นี้เป็นวินโดว์แบบ Child ที่เกิดจาก Parent ตัวไหน หากไม่มีก็ให้ค่าเป็น NULL

6. หากต้องการกำหนดชื่อเมนูใหม่ที่จะลงไปแทนชื่อเมนู ที่กำหนดไว้คอนริจิสเตอร์ คลาสสามารถใส่ได้ที่พารามิเตอร์ดังนี้

7. เป็นค่า Instance ของแอปพลิเคชันที่เป็นเจ้าของวินโดว์ตัวนี้

8. เป็นส่วนของข้อมูลเพิ่มเติมที่จะถูกใช้ในฟังก์ชันวินโดว์ หากไม่มีก็เป็น NULL

หากว่าการสร้างวินโดว์เป็นผลสำเร็จ ฟังก์ชัน CreateWindow นี้จะให้แฮนเดิลของวินโดว์นั้นกลับมา แต่หากไม่สำเร็จจะให้ค่าเป็น NULL ซึ่งจำเป็นต้องมีการตรวจสอบหากพบว่าวินโดว์ถูกสร้างไม่สำเร็จก็ให้จบแอปพลิเคชันแล้วกลับเข้าสู่ Windows เมื่อสร้างวินโดว์สำเร็จแล้วจะต้องมีการบอก Windows ให้แสดงวินโดว์นั้นขึ้นมาโดยใช้ฟังก์ชัน ShowWindow ซึ่งการที่วินโดว์นั้นจะถูกแสดงเป็นวินโดว์หรือไอคอนนั้นขึ้นอยู่กับอากิวเมนต์ nCmdShow ที่ส่งให้กับวินโดว์นั้น และหลังจากมีการแสดงวินโดว์แล้วก็ควรจะมีการส่งเมสเสจ WM_PAINT เพื่อให้วินโดว์นั้นวาดใหม่ (โดยเฉพาะวินโดว์หลักของแอปพลิเคชัน) ตัวอย่างการใช้เป็นดังนี้

```
ShowWindow(hWnd, nCmdShow);          /* Shows the windows */
UpdateWindow(hWnd);                  /* ส่ง WM_PAINT ไปยังวินโดว์ */
```

การสร้าง Message loop

ที่กล่าวมาข้างต้น (การริจิสเตอร์และการสร้างวินโดว์) เป็นส่วนการ Initialize ของ แอปพลิเคชัน หน้าที่ส่วนต่อมาของ WinMain คือการอ่านเมสเสจจากคิวของระบบ (System Queue) แล้วส่งต่อไปให้วินโดว์ต่าง ๆ เมสเสจที่เข้ามาในคิวของระบบจะมาจากส่วนที่เป็นอินพุตของเครื่อง เช่น จากการเลื่อนเมาส์ การคลิกเมาส์ การกดคีย์บอร์ดหรือแม้แต่จากไทเมอร์ในระบบ (ตัวอย่างเช่นนี้ เมื่อผู้ใช้ทำการคลิกปุ่มซ้ายของเมาส์ Windows ก็ส่ง WM_LBUTTONDOWN เข้าไปในคิวของระบบ จากนั้นเมื่อแอปพลิเคชันอ่านพบเมสเสจนี้ แอปพลิเคชันก็จะรู้ว่าผู้ใช้มีแอกชันอะไรและจะต้องตอบสนองเมสเสจนั้นอย่างไร

ฟังก์ชันในการอ่านเมสเสจจากคิวของระบบ GetMessage และฟังก์ชันในการส่งเมสเสจไปยังวินโดว์ต่างๆ คือ DispatchMessage เมสเสจที่เข้ามาในระบบจะเข้ามาอย่างแรนดอม คือไม่ทราบว่าจะมีมาเมื่อใด จึงต้องมีการตรวจสอบอยู่ตลอดเวลา (โดยใช้ while) เมื่อมีเมสเสจเข้ามาก็ จะต้องส่งต่อไปยังวินโดว์ฟังก์ชันที่เหมาะสมดังต่อไปนี้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MSG msg;
```

```
while(GetMessage(&msg, NULL, NULL, NULL) ) {
    DispatchMessage(&msg);
}
```

เราเรียกส่วนทั้งหมดของ while นี้ว่า message loop แต่หากแอปพลิเคชันต้องการรับข้อมูลจากคีย์บอร์ด เราจะต้องใช้ฟังก์ชัน TranslateMessage เพื่อทำการตรวจสอบ message WM_KEY และ WM_KEYDOWN (ซึ่งเป็น message ที่แสดงว่าผู้ใช้มีการกดคีย์บอร์ด) จากนั้นฟังก์ชันนี้จะส่ง WM_CHAR พร้อมกับค่าของคีย์ที่กดไปยังวินโดว์ฟังก์ชัน เมื่อมีการใช้ฟังก์ชันนี้เข้าช่วย ลักษณะโปรแกรมส่วนนี้จะกล่าวเป็นดังนี้

```
while(GetMessage(&msg, NULL, NULL, NULL) ) {
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

การจบแอปพลิเคชัน

ฟังก์ชันที่เป็นฟังก์ชัน main ของแอปพลิเคชัน Windows คือ WinMain เพราะฉะนั้นการจบแอปพลิเคชันก็ต้องจบที่ฟังก์ชัน หากเป็นการออกจากแอปพลิเคชันก่อนที่จะเข้า message loop ที่ทำได้โดยการ return บรรทัด (กรณีนี้ก็เช่น ตรวจสอบว่ามี แอปพลิเคชันนี้ลงไว้รึยังก่อนแล้วและไม่ต้องการรันซ้ำอีก) แต่หากเข้าไปใน message loop แล้วต้องการออกจากแอปพลิเคชัน ก็ทำได้โดยฟังก์ชัน PostQuitMessage ซึ่งเป็นฟังก์ชันที่จะวางเมสเสจ WM_QUIT ลงในคิวของระบบเมื่อ GetMessage อ่านได้เมสเสจ นี้ก็จะให้ค่า NULL ซึ่งก็เป็น การจบออกจากรูปและจบแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันการ Initialization

นอกจากฟังก์ชัน WinMain ซึ่งต้องมีเสมอแล้ว ประเพณีการเขียนโปรแกรมบน Windows ยังมีฟังก์ชันที่ต้องเขียนอีกสองส่วน ซึ่งเป็นส่วนในการ Initialize แอปพลิเคชัน (ซึ่งที่จริงแล้วใส่รวมกันใน WinMain ก็ได้ แต่นิยมเขียนแยกออกมาเพื่อให้ง่ายในการอ่าน) ส่วนแรกจะเป็นการรีจิสเตอร์วินโดว์คลาส ซึ่งมีตัวอย่างดังนี้

```

BOOL InitApplication(hInstance)
HANDLE hInstance;
{
    WNDCLASS wc;
    /* Fill in window class structure with parameters that describe the main window. */
    wc.style = NULL; /* Class style(s). */
    wc.lpfnWndProc = MainWndProc; /* Function to retrieve message for window of this class. */
    /*
    wc.cbClsExtra = 0; /* No per - class extra data. */
    wc.cbWndExtra = 0; /* No per - window extra data. */
    wc.hInstance = hInstance; /* Application that owns this class. */
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wc.hCursor = LoadCursor(NULL, IDC_ARROW);
    wc.hbrBackground = GetStockObject(WHITE_BRUSH);
    wc.lpszMenuName = "GenericMenu"; /* Name of menu resource in .RC file. */
    wc.lpszClassName = "GenericWClass"; /* Name used in call to CreateWindow. */
    /* Register the window class and return success/failure code. */
    return(RegisterClass(&wc);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่สองเป็นการสร้างวินโดวโดยใช้วินโดวคลาสที่สร้างขึ้น

```
BOOL InitInstance(hInstance, nCmdShow)
```

```

HANDLE hInstance;          /* Current instance identifier. */
int nCmdShow;              /* Param for first ShowWindow() call. */
{
    HWND hWnd;             /*Main window handle. */
    /* Save the instance handle in static variable, which will be used in many subsequence */
    /* calls from application to Windows. */
    hInst = hInstance;
    /* Create a main window fir this application instance. */
    hWnd = CreateWindow(
        1. "GenericWClass", /* See Register Class() call. */
        2. "Generic Sample Application", /* Text for window title bar. */
        3. WS_OVERLAPPEDWINDOW, /* Window style. */
        4. CW_USEDEFAULT, /* Default horizontal position. */
        CW_USEDEFAULT, /* Default vertical position. */
        CW_USEDEFAULT, /* Default width. */
        CW_USEDEFAULT, /* Default hight. */
        5. NULL, /* Overlapped windows have no parent. */
        6. NULL, /* Use the window class menu. */
        7. hInstance, /* This instance owns thia window. */
        8. NULL /* Pointer to needed. */
    );
    /* If window could not be create, return 'failure' */
    if(!hWnd)
        return(FALSE);
    /* make the window visible; update its olient area; and return "success" */
    ShowWindow(hWnd, nCmdShow); /* Show the window */
    UpdateWindow(hWnd); /* Sends WM_PAINT message */
    return(TRUE); /* Returns the value from PostQuitMessage */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วินโดว์ฟังก์ชัน

ส่วนที่กล่าวมาข้างต้นเป็นเรื่องของการเริ่มสร้างแอปพลิเคชันเท่านั้น คิวฟังก์ชันที่ใช้ทำงานจริงๆ จะแยกออกมาอีกส่วนหนึ่งเรียกว่า วินโดว์ฟังก์ชัน หรือ ฟังก์ชันประจำวินโดว์ ซึ่งมีหน้าที่ตอบสนองเมสเสจต่าง ๆ ลักษณะของวินโดว์ฟังก์ชันเป็นดังนี้

```
long FAR PASCAL MainWndProc(hWnd, message, wParam, lParam)
HWND hWnd; /* Window handle */
unsigned message; /* type of message */
WORD wParam; /* additional information */
LONG lParam; /* additional information */
{
    switch(message) {
        default; /* Passes it on if unprocessed */
            return(DefWindowProc(hWnd, message, wParam, lParam) );
    }
    return(NULL);
}
```

วินโดว์ฟังก์ชันต้องใช้ในการเรียกแบบ FAR เพราะการอ้างถึงแอดเดรสในหน่วยความจำจะใช้พอยน์เตอร์ขนาด 32 บิต และจะต้องเป็นการผ่านค่าตัวแปรแบบ PASCAL นอกจากนี้ชื่อของวินโดว์ฟังก์ชันนั้นจะต้องเป็นชื่อเดียวกับคอนเทร็ริบิวเตอร์วินโดว์คลาส (ในพารามิเตอร์ lpfnWndProc) หน้าที่ของวินโดว์ฟังก์ชันคือ รับ message จากทั้งคิว ฟังก์ชันหลัก (WinMain) และโดยตรงจากคิว Window โดยผ่านตัวแปรที่ชื่อว่า message จากนั้นก็ใช้ switch เพื่อแยกแยะว่าเมสเสจใดที่กำลังคอยอยู่และต้องทำอะไรบ้าง ส่วนเมสเสจอื่นก็จะส่งกลับไปยัง Windows โดยใช้ฟังก์ชัน DefWindowProc

หากเมสเสจใดที่ต้องมีข้อมูลอื่นๆ ส่งมาด้วย ส่วนมากจะมากับพารามิเตอร์ lParam หรือ ไม่ก็ wParam เช่น WM_MOUSEMOVE ซึ่งบอกว่าขณะนี้เมาส์กำลังมีการเคลื่อนที่ก็จะมีโคออดิเนตของเมาส์ส่งมาทาง lParam ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาแล้วว่าเมสเสจที่เข้ามาในวินโดว์ฟังก์ชันนั้นมาได้สองทางคือ จากตัว WinMainซึ่งเมสเสจเหล่านี้เป็นของการอินพุตต่าง ๆ เช่น การคลิก การเลื่อนเมาส์ การกดคีย์บอร์ด ส่วนที่มาโดยตรงจาก Windows จะเป็นเมสเสจที่พิเศษๆ บางอย่างที่ตัว WinMain อาจไม่สามารถรับรู้ได้เช่น เมื่อมีการเปลี่ยนขนาดของวินโดว์

ตัว Windows จะส่ง WM_PAINT มายังวินโดว์ที่ถูกเปลี่ยนขนาดนั้นเพื่อให้มีการวาดพื้นที่ใช้งาน (Client Area) ใหม่หรือ WM_DESTROY ที่ Windows จะส่งมาเมื่อมีการปิดวินโดว์นั้นซึ่งส่วนมากผู้ใช้ต้องการให้เป็นการจบแอปพลิเคชันเลยจึงนิยมมีการตรวจสอบเมสเสจด้วยตัวอย่างเช่น

```
case WM_DESTROY:
```

```
    PostQuitMessage(0);
```

```
    break;
```

เมื่อวินโดว์ฟังก์ชันทำคำสั่ง PostQuitMessage ก็จะมีการใส่เมสเสจ WM_QUIT ลงในคิวของระบบ และเมื่อ WinMain อ่านขึ้นมา ก็จะเป็นการจบแอปพลิเคชันในทันที

1.3 การสร้างกรอบข้อความ

กรอบข้อความ (dialog box) เป็นวินโดว์อย่างหนึ่งที่ถูกสร้างขึ้นมาเพื่อติดต่อกับผู้ใช้ เช่น แจ้งข่าวสารบางอย่าง แจ้งความคิดพลาด บอกชื่อแอปพลิเคชัน หรือให้ผู้ใช้ใส่ชื่อไฟล์ที่ต้องการโหลดขึ้นมาทำงาน และเมื่อทำงานเสร็จแล้วมักจะหายไป (มาตรฐาน SAA ยังแนะนำว่าควรมีกรอบข้อความที่บอกเรื่องราวของแอปพลิเคชันหรือที่เรียกว่า About Dialog Box ในทุก ๆ แอปพลิเคชัน)

การสร้างกรอบข้อความมีขั้นตอนดังนี้

1. สร้าง template ของกรอบข้อความในไฟล์ resource
2. ใช้คำสั่ง DialogBox ในส่วนโปรแกรมภาษาซี
3. ใส่ชื่อกรอบข้อความในส่วน EXPORT ของไฟล์ module-definition
4. เพิ่มเมนูในไฟล์ resource ให้มีคำสั่งให้แสดงกรอบข้อความ
5. ตรวจสอบ WM_COMMAND เพื่อแสดงกรอบข้อความในช่วยที่ต้องการ

การสร้าง Template ของกรอบข้อความ

Template ในที่นี้หมายถึงลักษณะของกรอบข้อความนั้น ทั้งตำแหน่ง ขนาด และสิ่งที่บรรจุอยู่ข้างใน การสร้าง template ทำได้สองทาง คือ ใช้ แอปพลิเคชัน DialogEditor หรือสร้างขึ้นมาเองในไฟล์ resource ซึ่งจะ

เป็นวิธีการที่ใช้ต่อไปนี้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะ template ของกรอบข้อความในไฟล์ resource เป็นดังนี้

1. AboutBox DIALOG 22, 17, 144, 75
 2. STYLE DS_MODALFRAME | WS_CAPTION | WS_SYSMENU
CAPTION "About Generic" -1, 0,, 5, 144, 8
 3. CTEXT "Microsoft Windows" -1, 0, 14, 144, 8
 4. CTEXT "Version 3.0" -1, 0, 34, 144, 8
 5. DEFPUSHBOTTON "OK" IDOK, 53, 59, 32, 14, WS_GROUP
- END

1. ส่วนแรกเป็น ชื่อของกรอบข้อความ ตามด้วยคีย์เวิร์ด DIALOG และตำแหน่งทางแกน X และแกน Y (อ้างอิงกับพื้นที่ใช้งานของวินโดวที่เป็น Parent) ส่วนสองตัวสุดท้ายเป็นความกว้างและความสูง หน่วยทางแกน X จะเป็น 1/4 ของหน่วยมาตรฐาน ของกรอบข้อความ ส่วนหน่วยทางแกน Y จะเป็น 1/8 ของหน่วยมาตรฐาน ของกรอบข้อความซึ่งหน่วยนี้สามารถเรียกดูได้จากฟังก์ชัน GetDialogBaseUnits

2. ส่วนนี้บอกสไตล์ของกรอบข้อความว่าเป็นอย่างไร

3. BEGIN และ END บอกจุดเริ่มและสิ้นสุดของส่วนที่เป็น template ของกรอบข้อความ

4. TEXT เป็นคำสั่งเขียนข้อความในกรอบข้อความ ส่วนความหมายของตัวเลขข้างหลังครุรายละเอียดในเรื่อง กรอบข้อความ

5. DEFPUSHBUTTON จะสร้างกรอบสี่เหลี่ยมที่มีคำว่า "OK" อยู่ภายใน ทำหน้าที่ในการปิดกรอบ ข้อความ

การสร้างฟังก์ชันของกรอบข้อความ

กรอบข้อความเป็นวินโดวที่เรียกว่า built-in คือผู้ใช้ไม่ต้องรีจิสเตอร์เอง แต่กรอบข้อความทุกอันที่จะต้องมีฟังก์ชันที่ใช้ในการประมวลผลเมสเสจที่เข้ามายังกรอบข้อความ นั้น เรียกว่า ฟังก์ชันของกรอบข้อความ (Dialog Function)

หน้าตาของฟังก์ชันของกรอบข้อความเป็นอย่างนี้

```
BOOL FAR PASCAL About(hDlg, message, wParam, lParam)
```

```
HWND hDlg; /* window handle of the dialog box.*/
```

```
unsigned message; /* type of message */
```

```
WORD wParam; /* message-specific information */
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LONG lParam;

```
{
    switch(message) {
    case WM_INITDIALOG:                /* message: initialize dialog box */
        return(TRUE);
    case WM_COMMAND:                   /* message: received a command */
        if(wParam == IDOK             /* "OK" box selected? */
        || wParam == IDCANCEL) {      /* System menu close command? */
            EndDialog(hDlg, TRUE);    /* Exits the dialog box */
            return (TRUE);
        }
        break;
    }
    return(FALSE);                    /* Didn't process a message */
}
```

ลักษณะโดยทั่วไปจะเหมือนกับฟังก์ชันของวินโดวส์ (กำหนดเป็น FAR ผ่านค่าตัวแปร แบบPASCAL และต้องกำหนดในส่วน EXPORT ของไฟล์ module-definition) แต่ส่วนที่ต่างคือไม่ต้องใช้ฟังก์ชัน DefWindowProc ในการจัดการกับเมสเสจที่ไม่ได้ทำการประมวลผล ซึ่งเมสเสจที่กรอบข้อความในงานมี 2 ข้อความคือ WM_INITDIALOG เป็นเมสเสจที่ส่งมายังกรอบข้อความก่อนที่จะแสดงออกมาทางหน้าจอ เพื่อให้มีการเตรียมการต่าง ๆ เช่น ในตัวอย่าง return (TRUE) หมายถึง ให้เลื่อนโฟกัสไปที่คอนโทรลตัวแรก หากเป็น return (FALSE) ก็ให้ไม่โฟกัสที่คอนโทรลใด ๆ

ส่วนเมสเสจอีกข้อความที่กรอบข้อความจะให้ประมวลผลคือ WM_COMMAND เป็นเมสเสจที่เกิดขึ้นเพื่อตอบสนองขณะที่ผู้ใช้เินพุตเข้าทางกรอบข้อความ เช่น กดปุ่ม หรือคีย์คีย์บอร์ด เป็นต้น ซึ่งปกติต้องมีปุ่มหนึ่งที่เมื่อผู้ใช้กดแล้วให้กรอบข้อความทำฟังก์ชัน EndDialog ได้ ไม่เช่นนั้นก็จะไม่สามารถปิดกรอบข้อความและกลับไปทำงานตามปกติได้

1.4 การสร้างเมนู

เมื่อสร้างตัวกรอบข้อความในไฟล์ resource และสร้างฟังก์ชันที่จะประมวลผลเมสเสจของกรอบข้อความแล้ว อีกสิ่งหนึ่งคือ ต้องหาวิธีที่จะให้ตัวแอปพลิเคชันรู้ว่าตอนนี้ผู้ใช้ต้องการให้แสดงกรอบข้อความออกมา ซึ่งคำสั่งไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้แสดงกรอบข้อความ About นี้มักจะเป็นคำสั่งสุดท้ายในเมนู Help หรือไม่กี่ในเมนู File แต่แอปพลิเคชัน Generic นี้จะมีเมนู Help เมนูเดียว ดังนั้นจึงจะใส่ About ในเมนู Help นี้

การสร้างเมนูของแอปพลิเคชันต่าง ๆ ก็เป็นการสร้างในไฟล์ resource เหมือนกับการสร้าง template ของกรอบข้อความ ลักษณะการสร้างเมนูเป็นดังนี้

```
GenericMenu MENU
```

```
BEGIN
```

```
POPUP      "&Help"
```

```
BEGIN
```

```
    MENUITEM      "&About Generic...", IDM_ABOUT
```

```
END
```

```
END
```

GenericMenu เป็นชื่อเมนูที่แอปพลิเคชันสามารถเรียกใช้ได้ ส่วน MENU เป็นคีย์เวิร์ดบอกส่วนต่อไปนี้เป็น การกำหนดเกี่ยวกับเมนู BEGIN และ END ก็จะเป็นตัวบอกจุดเริ่มและจุดสิ้นสุดในการกำหนดเมนูคำว่า POPUP เป็นคีย์เวิร์ดที่บอกว่าชื่อที่ตามมานี้ให้เป็นชื่อที่ปรากฏในเมนูบาร์(เรียกว่าเป็น เมนูหลัก) ส่วนเครื่องหมาย "&" จะเป็นการสั่งว่าให้ บิดเส้นใต้ตัวอักษรต่อไปที่ตามมา (ในที่นี้คือตัว H) และเมื่อผู้ใช้กด Alt+H ก็จะเป็นการ แอคทีฟเมนูนี้

สำหรับบรรทัดที่อยู่ระหว่าง BEGIN และ END ด้านในจะเป็นเมนูย่อยในเมนูหลักนั้น คีย์เวิร์ด MENUITEM เป็นคำที่บอกว่า ข้อความที่ตามหลังมาให้แสดงออกเป็นหนึ่งเมนูย่อยและสุดท้ายสำคัญที่สุดเป็น คำที่แสดงเลขประจำเมนูย่อยนั้น เรียก Menu ID ซึ่งมีหน้าที่ในการบอกตัวแอปพลิเคชันว่าเมนูย่อยใดที่ผู้ใช้เลือก ในที่นี้จะเป็น IDM_ABOUT

1.4 การสร้างส่วนประมวลผลเมสเสจ

เมื่อผู้ใช้เลือก About Generic... ในเมนู Help แล้ว Windows จะใส่ WM_COMMAND เข้าไปในคิวของระบบ เมื่อ WinMain ดึง message นั้นขึ้นมา ก็จะส่งต่อไปให้วินโดว์ฟังก์ชันดังนั้นในวินโดว์ฟังก์ชันจึงต้องมีการตรวจสอบเมสเสจ ที่เข้ามาว่าเป็น WM_COMMAND หรือไม่

การที่จะบอกว่า WM_COMMAND ที่ได้มานั้นมาจากไหน ก็สามารถตรวจสอบได้จากพารามิเตอร์สองตัวที่ส่งมาด้วยคือ lParam และ wParam ในที่นี้จะได้ Menu ID มาทาง wParam ดังนั้นตัวโปรแกรมจึงออกมาเป็นลักษณะดังนี้

```

FARPROC lpProcAbout;                                /* pointer to the "About" function */

case WM_COMMAND:                                    /* message: command from application menu */
    if(wParam == IDM_ABOUT) {
        1. lpProcAbout = MakeProcInstance(About, hInst);
        2. DialogBox(hInst,                          /* current instance */
            "AboutBox",                               /* resource to use */
            hWnd,                                     /* parent handle */
            lpProcAbout);                             /* About() instance address */
        3. FreeProcInstance(lpProcAbout);
        break;
    }
else
    return(DefWindowProc(hWnd, message, wParam, lParam) );

```

1. ก่อนที่จะแสดงกรอบข้อความนั้นจะต้องมีการให้ค่าแอดเดรสที่แน่นอนของฟังก์ชันของกรอบข้อความมาก่อน ซึ่งฟังก์ชัน `MakeProcInstance` นี้จะใช้ค่า `Instance` ของแอปพลิเคชัน (`hInst`) นั้นรวมกับพอยน์เตอร์ของฟังก์ชัน (`About`) เพื่อรีเทิร์นค่าแอดเดรสของฟังก์ชันนั้นซึ่งมีลักษณะข้อมูลเป็น `FARPROC`

2. ฟังก์ชัน `DialogBox` เป็นฟังก์ชันใช้ในการแสดงกรอบข้อความ โดยมีพารามิเตอร์ที่ต้องการคือ ค่า `Instance` ของแอปพลิเคชัน ชื่อของ `template` ของกรอบข้อความนั้น แชนเคล็ดของวินโดว์ `Parent` และค่าแอดเดรสของฟังก์ชันของกรอบข้อความ (`lpProcAbout`)

ข้อควรระวัง คือ ฟังก์ชันนี้จะไม่รีเทิร์นการควบคุมกลับมาถ้ากรอบข้อความนั้นยังไม่ถูกปิดคั้งนั้น การออกแบบกรอบข้อความจะต้องมีอย่างน้อยหนึ่งปุ่มสำหรับปิดกรอบข้อความนั้นเพื่อให้การทำงานกลับมายังวินโดว์เดิม

3. เมื่อ `DialogBox` รีเทิร์นค่ากลับมาแล้ว `lpProcAbout` ก็ไม่จำเป็นต้องใช้ จึงควรมีการฟรีทิ้งไป โดยใช้ฟังก์ชัน `FreeProcInstance` ไม่เช่นนั้นแล้วหน่วยความจำส่วนนั้นก็จะไม่สามารถไปใช้งานอย่างอื่นหรือโดยแอปพลิเคชันอื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 การสร้างไฟล์ Include

ประเพณีของการเขียนโปรแกรมด้วยภาษาซีมักจะแยกการกำหนดค่าคงที่ และโปรโตไทป์ ของฟังก์ชันต่างๆ ในไฟล์ที่แยกออกมา เรียกว่า include file (หรือ headerfile) ในการสร้างแอปพลิเคชันบน Windows ไฟล์ ๆ นี้ จะถูกใช้ทั้งในส่วนที่เป็นโปรแกรมภาษาซีและส่วนที่เป็นไฟล์ resource

ตัวอย่างไฟล์ include ของแอปพลิเคชัน Generic จะเป็นดังนี้

```
#define IDM_ABOUT 100

int PASCAL WinMain(HANDLE, HANDLE, LPSTR, int);

BOOL InitApplication(HANDLE);

BOOL InitInstance(HANDLE, unsigned, WORD, LONG);

long FAR PASCAL MainWndProc(HWND, unsigned, WORD, LONG);

BOOL FAR PASCAL About(HWND, unsigned, WORD, LONG);
```

จะเห็นว่าในบรรทัดแรกเป็นการกำหนดค่าคงที่ของเมนู ที่เหลือจะเป็นการโปรโตไทป์ ของ WinMain, MainWndProc, InitApplication และ InitInstance การใช้งานไฟล์ Include นี้กับภาษาซีทั่วไป คือ

```
#include "WINDOWS.H" /* ไฟล์ include ของ Windows */
#include "GENERIC.H" /* ไฟล์ include ของ แอปพลิเคชัน Generic */
```

1.6 การสร้างไฟล์ Module-Definition

ไฟล์นี้เป็นอีกไฟล์ที่แอปพลิเคชันของ Windows จะต้องมีเพื่อกำหนดชื่อ เซกเมนต์และหน่วยความจำที่ต้องการ ซึ่งข้อความจำเป็นต้องมี มีดังนี้ NAME, STACKSIZE, HEAPSIZE, EXETYPE และ EXPORTS ไฟล์ module-definition ของแอปพลิเคชัน Generic มีลักษณะดังนี้

```
; module - definition for file generic - - used by LINK.EXE
NAME Generic;          application's module name
DESCRIPTION 'Sample Microsoft Windows Application'
EXETYPE WINDOWS      ; required for all Windows applications
STUB 'WINSTUB.EXE' ; Generates error message if application is run with Windows
CODE PRELOAD MOVEABLE DISCARDABLE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ; DATA must be MULTIPLE if program can be invoked more than once
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA PRELOAD MOVEABLE MULTIPLE

HEAPSIZE 1024

STACKSIZE 5120 ; recommended minimum for Windows applications

EXPORT

MainWndProc @1 ; name of window processing function

About @2 ; name of "About" processing function



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บทที่ 2

รูปภาพบิตแมพ (Bitmap) และการแสดงผล

บิตแมพ (bitmap) หากจะแปลกันตรง ๆ ก็หมายถึง "การแทนจุดต่าง ๆ ของรูปภาพด้วยการเรียงตัวของบิต" ใน Windows จะมีรูปบิตแมพอยู่สองอย่าง คือ

1. รูปบิตแมพที่ขึ้นกับอุปกรณ์ (device-dependent bitmap) เป็นรูปบิตแมพที่ใช้แต่ละบิตในหน่วยความจำ แสดงผลแทนจุดในภาพจริงๆ ดังนั้นหากใช้จอที่มีความละเอียดอย่างหนึ่งก็จะได้รูปที่มีขนาดหนึ่ง จึงได้ชื่อว่า "ขึ้นกับอุปกรณ์"
2. รูปบิตแมพแบบไม่ขึ้นกับอุปกรณ์ (device-independent bitmap-DIB) รูปแบบนี้จะเป็นการอธิบายลักษณะของภาพนั้น แทนที่จะใช้แต่ละบิตแทนแต่ละจุด ดังนั้นหากรู้จักลักษณะของจอจะสามารถแสดงรูปชนิดนี้ให้เหมาะกับจออื่น ๆ ได้

2.1 การสร้างรูปบิตแมพ

การสร้างรูปบิตแมพแบ่งได้เป็น 4 วิธีดังนี้

1. สร้างและโหลดรูปบิตแมพ

วิธีนี้เป็นวิธีที่สะดวกที่สุด โดยที่การสร้างนั้นหมายถึง ใช้แอฟพลิเคชันภายนอกสร้างรูปบิตแมพขึ้นมาก่อน และรวมภาพบิตแมพเป็นรีซอร์สของแอฟพลิเคชันและแอฟพลิเคชันจึงโหลดรูปเข้ามาแสดง แอฟพลิเคชันเหล่านี้ก็เช่น SDKPaint ของ SDK หรือ WhiteWater Resource Toolbox (WRT) และอื่น ๆ แต่ในที่นี้จะขออ้างอิงกับ SDKPaint เท่านั้น

การสร้างและโหลดรูปบิตแมพมีขั้นตอนดังนี้

1. เรียกแอฟพลิเคชัน SDKPaint ในกลุ่มของ SDK
2. ใช้เครื่องมือที่มีให้ เช่น การลากเส้น การพ่นสี หรือการระบายสีในการสร้างรูปจากนั้นให้เซฟไว้โดยให้มีนามสกุลเป็น BMP

3. การทำให้แอฟพลิเคชันรู้จักไฟล์บิตแมพที่สร้างไว้ให้เพิ่มคำบอก BITMAP เข้าไปในไฟล์รีซอร์ส เช่น ถ้ารูปบิตแมพถูกเก็บไว้ในไฟล์ dog.bmp ก็ให้เพิ่มบรรทัดนี้เข้าไป ในไฟล์ RC

```
dog BITMAP DOG.BMP
```

4. การเรียกรูปบิตแมพเข้ามาในแอฟพลิเคชันนั้น จะต้องมีการโหลดรูปเข้ามาในหน่วยความจำด้วยฟังก์ชัน LoadBitmap ซึ่งถ้าสามารถโหลดเข้ามาสำเร็จ ฟังก์ชันนี้ก็จะให้ค่าแอสเคลที่ชี้ไปยังรูปบิตแมพนั้นกลับมา ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037153

```
hDogBitmap = LoadBitmap(hInstance, "dog");
```

5. จากนั้นก็ใช้ฟังก์ชัน `SelectObject` เพื่อเลือกรูปเข้าไปในคอนเท็กซ์อุปกรณ์

```
SelectObject(hMemoryDC, hDogBitmap);
```

6. สุดท้ายก็เป็นการแสดงรูปออกทางวินโดวโดยใช้ฟังก์ชัน `BitBlt` ดังนี้

```
BitBlt(hDC, 10, 10, 100, 150, hMemDC, 0, 0, SRCCOPY);
```

จากตัวอย่างมี `hDC` เป็นแฮนเดิลของคอนเท็กซ์อุปกรณ์จริง ๆ พิกัด 10,10 เป็น ตำแหน่งที่จะวางรูป ตำแหน่ง 100,150 เป็นความกว้างและยาวของรูปที่ต้องการแสดงตำแหน่ง 0,0 ตัวหลังเป็นตำแหน่งของรูปใน `hMemoryDC` ที่จะเริ่มต้นก๊อปปี้ไปยัง `hDC` และพารามิเตอร์สุดท้าย `SRCCOPY` จะเป็นการบอกว่าให้ก๊อปปี้จากรูปต้นทางไปปลายทางโดยตรง (แต่หากเป็นวิธีอื่น เช่น ตัวเลือก `SRCAND` จะเป็นการทำโลจิกแอนด์กันระหว่างต้นทางปิดต่อบิต)

2. วาดรูปบนพื้นที่ว่างที่สร้างขึ้น

วิธีนี้เป็นการสร้างพื้นที่ว่างขึ้นมาก่อน แล้วจึงใช้ฟังก์ชันของ GDI ในการวาดรูปลง ไปการสร้างวิธีนี้มีขั้นตอน คือ

1. สร้างพื้นที่ว่างสำหรับบิตแมพโดยใช้ฟังก์ชัน `CreateBitmap` หรือ `Create CompatibleBitmap`
2. เลือกบิตแมพเข้าไปในคอนเท็กซ์อุปกรณ์ด้วยฟังก์ชัน `SelectObject`
3. ใช้ฟังก์ชัน GDI วาดภาพ

ตัวอย่างต่อไปนี้เป็น การสร้างรูปดาวโดยใช้ฟังก์ชัน `Polygon` ของ GDI

```
HDC hDC;
```

```
HDC hMemoryDC;
```

```
HBITMAP hBitmap;
```

```
HBITMAP hOldBitmap;
```

```
POINT Points[5] = [ 32, 0, 16, 63, 63, 16, 0, 16, 48, 63];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. `hDC = GetDC(hWnd);`
2. `hMemoryDC = CreateCompatibleDC(hDC);`
3. `hBitmap = CreateCompatibleBitmap(hDC, 64, 64);`
4. `hOldBitmap = SelectObject(hMemoryDC, hBitmap);`
5. `PatBit(hMemoryDC, 0, 0, 64, 64, WHITENESS);`
6. `Polygon(hMemoryDC, Points, 5);`
7. `BitBlt(hDC, 0, 0, 64, 64, hMemoryDC, 0, 0, SRCCOPY);`
8. `SelectObject(hMemoryDC, hOldBitmap);`
`DeleteDC(hMemoryDC);`
9. `ReleaseDC(hWnd, hDC);`

จากตัวอย่าง

1. ฟังก์ชัน `GetDC` ใช้ในการหาค่าแฮนเดิลคอนเท็กซ์อุปกรณ์ของวินโดว์นั้น
2. ใช้ `CreateCompatibleDC` สำหรับสร้างคอนเท็กซ์อุปกรณ์ในหน่วยความจำที่เหมือนกับคอนเท็กซ์อุปกรณ์

จริง

3. สร้างรูปบิตแมพเปล่าโดยใช้ฟังก์ชัน `CreateCompatibleBitmap` โดยที่ในที่นี้มีขนาดเป็น 64×64 จุดสี แต่จำนวนบิตที่ใช้จริงนั้นขึ้นอยู่กับชนิดของจอ ถ้าเป็นจอสี ในหนึ่งจุดสีก็จะประกอบด้วยหลายบิต
4. ฟังก์ชัน `SelectObject` จะเป็นการเลือกรูปบิตแมพที่สร้างไว้ เข้าไปในคอนเท็กซ์อุปกรณ์ โดยที่แฮนเดิลของรูปเก่าจะถูกเก็บไว้ที่ตัวแปร `hOldBitmap`
5. ฟังก์ชัน `PatBit` จะเซตทุกจุดสีในรูป ให้เป็นสีขาว
6. ฟังก์ชัน `Polygon` จะใช้สำหรับสร้างรูปเหลี่ยมขึ้นมา ค่าพารามิเตอร์ที่ฟังก์ชันนี้ต้องการคือ ตำแหน่งของจุดปลายต่าง ๆ ในที่นี้ก็จะอยู่ในอระเรย์ `Points`
7. ฟังก์ชัน `BitBlt` เป็นการก๊อปปี้รูปจากหน่วยความจำไปยังพื้นที่ใช้งานของวินโดว์
8. ฟังก์ชัน `SelectObject` อีกครั้งจะคืนรูปบิตแมพเก่าเข้าไปยังคอนเท็กซ์อุปกรณ์ดั้งเดิม และ `DeleteDC` จะทำการลบคอนเท็กซ์อุปกรณ์ที่สร้างจำลองไว้ในหน่วยความจำทิ้งไป
9. `ReleaseDC` จะเป็นการปล่อยการยึดครองคอนเท็กซ์อุปกรณ์เพื่อให้แอปพลิเคชันอื่นสามารถเรียกใช้ได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สร้างรูปบิตแมพด้วยวิธีกำหนดทีละจุด

วิธีนี้จะใช้สำหรับการสร้างรูปบิตแมพที่มีขนาดเล็ก เช่น รูปแบบของการระบายสี เพราะจะต้องกำหนดลงไปตรง ๆ ว่าบิตนี้ของไบต์นั้นมีค่าเป็น 0 หรือ 1 ฟังก์ชันที่ใช้กับวิธีการนี้คือ CreateDIBitmap

ส่วนของโปรแกรมต่อไปนี้ เป็นการสร้างรูปบิตแมพขนาด 64x32 พิกเซล โดยเก็บไว้ในตัวแปรอะเรย์ชื่อ Square

```
HBITMAP hBitmap;
```

```
HDC hDC;
```

```
BYTE Square[] = {
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```



```

pDibInfo -> bmiColor[1].rgbRed = 0xff;
pDibInfo -> bmiColor[1].rgbGreen = 0xff;
pDibInfo -> bmiColor[1].rgbBlue = 0xff;

hDC = GetDC(Hwnd);

hBitmap = CreateDIBitmap(hDC, (LPBITMAPINFOHEADER)&(pDibInfo -> bmiHeader),
    CBM_INIT, (LPSTR)Square, (LPBITMAPINFO)pDibInfo, DIB_RGB_COLORS);

ReleaseDC(Hwnd, DC);

DeleteObject(hBitmap);

LocalFree((HANDLE)pDibInfo);
}

```

ฟังก์ชัน `CreateDIBitmap` เป็นฟังก์ชันใช้ในการสร้างและใส่ค่าเริ่มต้นให้กับบิตแมพจากนั้นก็ให้ค่าแชนเคลิลของรูปบิตแมพนั้นกลับมา โดยที่มีความกว้างเป็น 64 จุดและความยาว(ความสูง) เป็น 32 จุด แต่ละจุดแทนด้วยหนึ่งบิต ดังนั้นรูปนี้จึงเป็นรูปสีเดียว (monochrome) และจุดเหล่านี้เก็บไว้ในอะเรย์ชื่อ `Square`

ตัวแปรโครงสร้างแบบ `BITMAP` จะเป็นตัวกำหนดว่าค่าที่อยู่ในอะเรย์นั้นจะถูกแปลความหมายอย่างไร ค่าเหล่านี้ก็คือ ความกว้าง ความยาว จำนวนบิตที่จะใช้แปลเป็นหนึ่งจุด (1,4,8, หรือ 24 บิต) และตารางสีของแต่ละจุด แต่ในตัวอย่างที่ปรากฏอยู่ข้างต้นเป็นรูปภาพสองสี ดังนั้นจึงใช้หนึ่งบิตแทนหนึ่งจุด และมีข้อมูลในตารางสีเพียงสองตัว(สำหรับค่า 0 จะเป็นสีดำและค่า 1 เป็นสีขาว)

การใช้ฟังก์ชันในการสร้างรูปบิตแมพนี้มีดังนี้

```
hBitmap = CreateBitmap(64, 32, 1, 1, (LPSTR)Square);
```

โดยที่ `hBitmap` จะเก็บแชนเคลิลของรูปบิตแมพที่สร้างขึ้น หากต้องการเปลี่ยนแปลงรูปที่ใช้แชนเคลิลนี้อยู่สามารถใช้ฟังก์ชัน `SetDIBits` มาช่วยได้ ตัวอย่างข้างล่างเป็นการนำข้อมูลในอะเรย์ `Circle` มาใช้งานแทน

```
BYTE Circle[ ] = {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 SetDIBit(hDC, hBitmap, 0, 32, (LPSTR)Circle, (LPBITMAPINFO)&myDIBInfo, DIB_RGB_COLORS);
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งทุกกรณีโดย

การสร้างรูปบิตแมพด้วยวิธีนี้เหมาะสำหรับรูปภาพเล็ก ๆ เท่านั้น เพราะว่าถ้ารูปใหญ่ขึ้นตัวเลขหนึ่งและศูนย์ก็จะมากขึ้นตาม ทำให้สับสนได้ง่ายเวลาใส่ข้อมูล

4. การสร้างรูปบิตแมพสี

หลักการสร้างรูปบิตแมพสีคือ สร้าง compatible บิตแมพว่างขึ้นมาก่อน จากนั้นก็สร้างแปรงทาสีสีต่างๆ (แดง เขียว และน้ำเงิน) มาวาดรูปลงในบิตแมพนั้น หากขอแสดงผลเป็นแบบโมนโครม Windows ก็ จะสร้างรูปแบบของจุดที่ใช้แทนสีต่าง ๆ เหล่านี้ขึ้นมา

ตัวอย่างการสร้างรูปบิตแมพเป็นดังนี้

```
HDC hDC;
HDC hMemoryDC;
HBITMAP hBitmap;
HBITMAP hOldBitmap;
HBRUSH hRedBrush;
HBRUSH hGreenBrush;
HBRUSH hBlueBrush;
HBRUSH hOldBrush;

hDc = GetDC(hWnd);
hMemoryDC = CreateCompatibleDC(hDC);
hBitmap = CreateCompatibleBitmap(hDC, 64, 32);
hOldBitmap = SelectObject(hMemoryDC, hBitmap);

hRedBrush = CreateSolidBrush(RGB(255, 0, 0));
hGreenBrush = CreateSolidBrush(RGB(0, 255, 0));
hBlueBrush = CreateSolidBrush(RGB(0, 0, 255));
```

```
PatBlt(hMemoryDC, 0, 0, 64, 32, BLACKNESS);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
hOldBrush = SelectObject(hMemoryDC, hRedBrush);
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PatBlt(hMemoryDC, 0, 0, 24, 11, PATORDEST);
PatBlt(hMemoryDC, 40, 10, 24, 12, PATORDEST);
PatBlt(hMemoryDC, 24, 22, 24, 11, PATORDEST);
SelectObject(hMemoryDC, hGreenBrush);
PatBlt(hMemoryDC, 24, 0, 24, 11, PATORDEST);
PatBlt(hMemoryDC, 0, 10, 24, 12, PATORDEST);
PatBlt(hMemoryDC, 40, 22, 24, 11, PATORDEST);
SelectObject(hMemoryDC, hBlueBrush);
PatBlt(hMemoryDC, 40, 0, 24, 11, PATORDEST);
PatBlt(hMemoryDC, 24, 10, 24, 12, PATORDEST);
PatBlt(hMemoryDC, 0, 22, 24, 11, PATORDEST);

BitBlt(hDC, 0, 0, 64, 32, hMemoryDC, 0, 0, SRCCOPY);

SelectObject(hMemoryDC, hOldBrush);
DeleteObject(hRedBrush);
DeleteObject(hGreenBrush);
DeleteObject(hBlueBrush);
SelectObject(hMemoryDC, hOldBrush);
DeleteDC(hMemoryDC, hOldBitmap);
ReleaseDC9hWnd, hDC);

```

จากตัวอย่างฟังก์ชัน `CreateSolidBrush` ทำหน้าที่สร้างแปรงทาสี สีต่างๆที่ต้องการใช้ในการวาดรูป โดยที่พารามิเตอร์ 3 ตัวที่ฟังก์ชันนี้ต้องการเป็นความเข้มของสีแดง เขียว น้ำเงิน ตามลำดับ จากนั้นก็ใช้ `SelectObject` เพื่อเลือกแปรงทาสีที่ต้องการ และใช้ `PatBlt` สำหรับทำการวาด โดย `PATORDEST` จะเป็นตัวที่บอกถึงลักษณะการวาดสี โดยจะมีสีใหม่ลงไปปนกับสีที่อยู่เดิมโดยการ `OR`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การแสดงรูปปิดแมพ

เมื่อทำการสร้างรูปปิดแมพได้ตามวิธีทั้ง 4 อย่างดังที่กล่าวมาแล้ว ก็มาถึงการแสดงรูปปิดแมพนั้น Windows ได้จัดวิธีแสดงรูปออกเป็นอีก 4 วิธีดังนี้

1. โดยการใช้ฟังก์ชัน BitBlt

ก่อนใช้ฟังก์ชันผู้ใช้จะต้องสร้างคอนเท็กซ์อุปกรณ์ในหน่วยความจำเสียก่อน แล้วจึงใช้ฟังก์ชันนี้คือบปี่รูปลงในหน่วยความจำส่วนที่สร้างขึ้น

```
HDC hDc, hMemoryDC;
```

```
hDC = GetDC(hWnd);
```

```
hMemoryDC = CreateCompatibleDC(hDC);
```

```
hOldBitmap = SelectObject(hMemoryDC, hBitmap);
```

```
if(hOldBitmap)
```

```
{
```

```
    BitBlt(hDC, 100, 30, 64, 32, hMemoryDC, 0, 0, SRCCOPY);
```

```
    SelectObject(hMemoryDC, hOldBitmap);
```

```
}
```

```
DeleteDC(hMemoryDC);
```

```
ReleaseDC(hWnd, hDC);
```

การทำงานของสามฟังก์ชันแรก ก็เหมือนกับที่เคยอธิบายในตัวอย่างที่กล่าวมาแล้ว ส่วน BitBlt จะทำการก๊อปปี้รูปจากคอนเท็กซ์อุปกรณ์ในหน่วยความจำไปยังคอนเท็กซ์อุปกรณ์ของจอ (จาก hMemoryDC ไปยัง hDC) โดยพิกัดตำแหน่งที่จะนำไปแสดงคือ (100,30) ความกว้างและยาวของรูปคือ (64,32) ส่วน SRCCOPY จะบอกให้เป็นการก๊อปปี้แบบเหมือนต้นฉบับทุกประการ

การบอกความกว้างและยาวของรูปนั้น นอกจากกำหนดลงไปโดยตรงดังตัวอย่างข้างต้นแล้วยังสามารถใช้ฟังก์ชัน `GetObject` มาช่วยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BITMAP Bitmap;

GetObject(hBitmap, sizeof(BITMAP), (LPSTR)&Bitmap);

ตัวแปรโครงสร้าง Bitmap จะเก็บคุณสมบัติต่างๆของรูปบิตแมพนั้นไว้รวมทั้งความกว้างและความยาวด้วย ทำให้เมื่อนำไปใช้ ในฟังก์ชัน BitBlt ก็จะได้เป็นดังนี้

BitBlt(hDC, 100, 30, Bitmap.bmWidth, Bitmap.bmHeight, hMemoryDC, 0, 0, SRCCOPY);

การใช้ฟังก์ชันนี้นั้นไม่ต้องคำนึงว่ารูปเป็นรูปกลายสีหรือสีเดียว แต่หากชนิดของรูปที่ต้นทาง (hMemoryDC) ไม่เหมือนกับอุปกรณ์ที่แสดงผล (hDC) แล้ว ก็จะทำให้เกิดการแปลงค่าบางประการให้เช่น ถ้ารูปเป็นสีแต่จอเป็นโมโนโครม รูปที่ปรากฏออกมาจะมีฉากหลังเป็นสีขาว และจุดอื่น ๆ เป็นสีดำทั้งหมด

2. โดยการใช้ฟังก์ชัน StretchBlt

รูปภาพที่แสดงออกทางอุปกรณ์ต่าง ๆ นั้นไม่จำเป็นที่จะต้องมียุทธศาสตร์เหมือนกับต้นฉบับ หากมีการใช้ฟังก์ชัน StretchBlt แทนฟังก์ชัน BitBlt แล้วก็สามารถที่จะย่อหรือขยายรูปนั้นได้ง่ายเพียงแค่เพิ่มพารามิเตอร์ที่ BitBlt ไม่มีเข้าไปอีกสองตัว คือความกว้างและยาวของรูปที่เอาต์พุต เช่น

StretchBlt(hDC, 100, 30, 128, 64, hMemoryDC, 0, 0, 64, 32, SRCCOPY);

ค่าพิกัด (128,64) ที่ปรากฏในตัวอย่างจะเป็นตัวบอกความกว้างยาวของรูปที่จะออกมาส่วน (64,32) จะเป็นความกว้างยาวจริงของรูป

ในการย่อขนาดรูปนั้น แน่นอนว่ารายละเอียดต่างๆ ของรูปจะต้องหดหายไปด้วย การลดการป้ายเบนครั้งนี้สามารถแก้ไขได้โดยบอก StretchBltให้ใช้วิธีการแสดงผลต่างๆกันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| โหมด | จุดประสงค์ |
|--------------|-----------------------------------|
| WHITEONBLACK | สร้างรูปที่เป็นสีขาวโดยมีพื้นสีดำ |
| BLACKONWHITE | สร้างรูปที่เป็นสีดำโดยมีพื้นสีขาว |
| COLORONCOLOR | สร้างรูปสี |

ฟังก์ชันที่ใช้ตั้งโหมดเหล่านี้ให้ StretchBlt รู้จักคือ SetStretchBltMode เช่น ถ้าต้องการให้ใช้โหมด WHITEONBLACK ก็ทำได้ดังนี้

```
SetStretchBltMode(hDC, WHITEONBLACK);
```

3. โดยการสร้างเป็นลักษณะของแปรงทาสี

อีกทางหนึ่งที่สามารถใช้รูปบิตแมพได้คือ นำไปสร้างเป็นลักษณะหนึ่งของแปรงทาสี ไว้ สำหรับทาสี (fill) ในรูปบิตต่าง ๆ เมื่อสิ่งเหล่านี้ลงในรูปบิตนั้น Windows จะก็อปปีรูปบิตแมพนั้นไปทางตั้งและทางนอน จนกว่าจะเต็มรูปบิต

บิตแมพที่ต้องการนำมาใช้งานอย่างนี้ควรสร้างให้มีความกว้างยาว 8x8 จุดสี ทั้งนี้เนื่องจาก Window จะใช้เพียงรูปขนาด 8x8 จุดสีที่ตำแหน่งมุมบนซ้ายเท่านั้น และการสร้างอาจเป็นการกำหนดทีละจุด (hard code) โดยการสร้างและวาด หรือจะใช้การโหลดด้วยไฟล์ซอร์สก็ได้ ไม่ว่าจะเป็นการสร้างด้วยวิธีใดก็ตาม จะได้เป็นแชนเคลลของ รูปบิตแมพนั้นกลับมา จากนั้นก็นำแชนเคลลนั้นผ่านไปให้กับฟังก์ชัน CreatePatternBrush ดังตัวอย่างต่อไปนี้

```
hBitmap = LoadBitmap(hInstance, "Checks");
```

```
hBrush = CreateCompatibleBrush(hBitmap);
```

hBrush เป็นแชนเคลลของแปรงที่สร้างขึ้น การนำไปใช้งานให้ใช้ฟังก์ชัน SelectObject ดังนี้

```
hOldBrush = SelectObject(hDc, hBrush);
```

ลักษณะการใช้ฟังก์ชันนี้จะถือว่าบิตแมพที่เลือกใหม่นั้นเป็นส่วนหนึ่งของแปรงทาสีดังนั้น จะไม่กระทบกับรูปบิตแมพอื่นที่ถูกเลือกไว้ก่อนแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นก็เป็นการนำไปใช้งาน ตัวอย่างต่อไปนี้เป็นารระบายสีที่มุมบนซ้ายของวินโดว์

```
PatBlt(hdc, 0, 0, 100, 100, PATORDEST);
```

ส่วนที่เป็นพื้นของวินโดว์ของแอปพลิเคชัน ก็สามารถขีดให้เป็นรูปแบบที่ต้องการได้โดยเปลี่ยนในโครงสร้างของวินโดว์คลาส ดังนี้

```
pWndClass -> hbrBackground = CreatePatternBrush(hBitmap);
```

หรือไม่ก็ใช้ฟังก์ชัน SetClassWord

```
SetClassWord(hWnd, GCW_HBRBACKGROUND, hBrush);
```

แต่วิธีนี้ก็มีข้อเสียคือ การเปลี่ยนรูปแบบของพื้นนั้นจะเกิดขึ้นกับทุกวินโดว์ที่ใช้คลาสนี้แต่หากต้องการเปลี่ยนเฉพาะบางวินโดว์ก็จะต้องมีการกำหนดลงไปอย่างชัดเจนโดยจะกระทำต่อเมื่อได้รับเมสเสจ WM_ERASEBKGD

```
RECT Rect;
```

```
HBRUSH hOldBrush;
```

```
case WM_ERASEBKGD:
```

```
UnRealizeObject(hMyBkgndBrush);
```

```
hOldBrush = SelectObject(wParam, hMyBkgndBrush);
```

```
GetUpdateRect(wParam, (LPRECT)&Rect, FALSE);
```

```
PatBlt(wParam, Rect.left, Rect.top, Rect.right - Rect.left, Rect.bottom -  
Rect.top, PATCOPY);
```

```
SelectObject(wParam, hOldBrush);
```

```
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะที่ได้รับเมตาดেja WM-ERASEBKGD ก็จะได้แชนเดิลของคอนเท็กส์อุปกรณมาในคัวแปร wParam ด้วย ฟังก์ชัน GetUpdateRect จะให้รายละเอียดของพื้นที่ที่ต้องถูกลบเพื่อวาดใหม่กลับมาในโครงสร้าง Rect ซึ่งถูกนำไปใช้ในฟังก์ชัน PatBit เพื่อวาดรูปบิตแมพลงในพื้นที่นั้น ส่วนฟังก์ชัน UnRealizeObject ทำหน้าที่ในการปรับจุดเริ่มต้นของบิตแมพที่วาดใหม่เมื่อวินโดว์นั้นถูกเคลื่อนย้าย

4. การแสดงรูปบิตแมพแบบไม่ขึ้นกับอุปกรณ์

ประโยชน์อย่างหนึ่งของการแสดงผลแบบนี้คือ ไม่จำเป็นต้องสร้างคอนเท็กส์อุปกรณ์ จำลองในหน่วยความจำก่อน ทำให้หน่วยความจำที่ต้องการลดลงไปเล็กน้อย ฟังก์ชันที่นำมาใช้คือ SetDIBitsToDevice ส่วนค่าพารามิเตอร์ต่าง ๆ ที่ต้องการ คือ

- คอนเท็กส์อุปกรณ์ของอุปกรณ์แสดงผลที่ต้องการแสดง
- ตำแหน่งที่ต้องการแสดงรูป
- ขนาดของรูปภาพที่ต้องการแสดง
- จำนวนเส้นสแกนของรูปต้นทาง
- ตำแหน่งของจุดสีแรก(มุมซ้ายบน) ในรูปต้นทางที่เริ่มแสดง
- โครงสร้างที่เก็บข้อมูลของรูปบิตแมพ รวมทั้งบัฟเฟอร์ที่เก็บรูปต้นทางอยู่ด้วย
- ค่าที่บอกว่าข้อมูลที่อยู่ในจานสีหรือตารางสี (palette table) ที่ใช้นั้นเป็นการบอกค่าสี RGB จริง

หรือเป็นดัชนีชี้ค่าของจานสีที่ใช้อยู่ในขณะนั้น

ตัวอย่างการใช้ฟังก์ชันเป็นดังนี้

```
SetDIBitsToDevice(hdc, 0, 0, lpbi -> bmiHeader.biWidth, lpbi -> bmiHeader.biHeight, 0, 0, 0,
lpbi -> bmiHeader.biHeight, pBuf, (LPBITMAPINFO) lpbi, DIB_RGB_COLORS);
```

พารามิเตอร์ตัวแรกเป็นแชนเดิลของคอนเท็กส์อุปกรณ์ สองตัวต่อมาเป็นจุดเริ่มต้นของรูปที่จะแสดง ในกรณีนี้ ก็จะเป็นมุมบนซ้ายของอุปกรณ์นั้นๆตัวแปร โครงสร้างสองตัวถัดมาเป็นความกว้างและความสูงของรูปที่ต้องการ แสดง

เลข 0,0 ที่ตามมาเป็นจุดเริ่มของรูปต้นทาง (0,0 จะเป็นการเริ่มต้นตั้งแต่มุมบนซ้ายของภาพ) เลข 0 ตัวต่อมาเป็นจุดเริ่มต้นของการทำ banding (ศึกษาเพิ่มเติมได้เรื่อง "การพิมพ์") ส่วนตัวที่ตามมาเป็นความกว้างของแบนด์ที่ต้องการ ในที่นี้ก็หมายถึง ในหนึ่งแบนด์ครอบคลุมทั้งรูป

lpbi เป็นบัฟเฟอร์ที่เก็บรูปต้นทาง บัฟเฟอร์นี้อยู่ในคัวแปร โครงสร้าง BITMAPINFO ส่วนพารามิเตอร์ตัวสุดท้ายเป็นไปได้อสองอย่างคือ DIB_RGB_COLORS (ตามตัวอย่าง) จะบอกว่าตารางสีที่ใช้นั้นเก็บเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลการผสมสีระหว่าง แดง เขียว และน้ำเงิน จริง ๆ แต่หากเป็น DIB_PAL_COLORS จะเป็นดัชนี (อะเรย์ 16 บิต) ซึ่งตำแหน่งในจานสี (palette) ที่ใช้งานอยู่ในขณะนั้น

2.3 การใส่สีให้กับรูปบิตแมพที่เป็นสีเดียว (ขาวดำ)

สำหรับรูปบิตแมพที่เป็นรูปสีเดียว เราสามารถที่จะกำหนดสีจุดที่เป็นสีดำ (ตัวรูป) กับจุดที่เป็นสีขาว (พื้น) ได้ว่าให้แสดงด้วยสีอะไรแทน การตั้งสีของรูปนั้นจะใช้ฟังก์ชัน SetTextColor และการเซตสีของพื้นใช้ฟังก์ชัน SetBkColor

ตัวอย่างต่อไปนี้เป็นการเล่นการเซตสีเพื่อให้ตัวรูปเป็นสีแดง ส่วนพื้นที่เป็นสีเขียว

```
SetTextColor(hDc, RGB(255, 0, 0));
```

```
SetBkColor(hDC, RGB(0, 255, 0));
```

ฟังก์ชัน RGB จะรับค่าความเข้มของสีแดง เขียว และน้ำเงิน (เรียงตามลำดับของพารามิเตอร์) มาสร้างรหัสของสีนั้นกลับมา โดยค่าความเข้มที่น้อยที่สุดคือ 0 และความเข้มที่มากที่สุดเป็น 255 ดังนั้น RGB (255,255,0) จะมีสีแดงเข้มเท่ากับสีเขียว สีที่ออกมาจึงเป็นสีเหลือง

2.4 การลบรูปบิตแมพ

รูปบิตแมพก็เหมือนกับวีซอร์สอย่างอื่น คือ ต้องมีการจองหน่วยความจำมาเรียกใช้ ดังนั้น เมื่อเลิกใช้ก็ต้องมีการลบทิ้ง ฉะนั้นเพื่อให้หน่วยความจำส่วนนั้นว่างลง และสามารถนำไปใช้งานอย่างอื่นได้ การลบต้องให้คอนเท็กซ์อุปกรณ์เลือกรูปบิตแมพเดิมเสียก่อน โดยใช้

```
SelectObject(hMemoryDC, hOldBitmap);
```

แล้วจึงลบแอสแตลที่ซีรูปบิตแมพนั้นทิ้ง

```
DeleteObject(hBitmap);
```

การที่ต้องเลือกรูปอื่นเข้าไปในคอนเท็กซ์อุปกรณ์ก่อน เพราะ Windows จะไม่ยอมให้ลบแอสแตลที่กำลังถูกใช้อยู่ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 คอนเท็กซ์อุปกรณ์

คอนเท็กซ์อุปกรณ์เป็นสะพานเชื่อมข้อมูลระหว่างแอปพลิเคชัน ไดรเวอร์และอุปกรณ์เข้าด้วยกัน เมื่อแอปพลิเคชันที่ต้องการแสดงผลทางอุปกรณ์ต่างๆก็สามารถเรียกใช้งานฟังก์ชันใช้งานกราฟิก(GDI)โดยไม่ต้องสนใจลักษณะของอุปกรณ์หรือสนใจเพียงเล็กน้อย ซึ่งฟังก์ชันเหล่านี้ก็จะผ่านข้อมูลเหล่านั้น ไปให้ไดรเวอร์ของอุปกรณ์ และไดรเวอร์ของอุปกรณ์นั้นก็จะไปเรียกใช้งานอุปกรณ์ต่างๆอีกที

การติดต่อกับอุปกรณ์แสดงผล

แอปพลิเคชันของ WINDOWS จะสามารถเรียกใช้ฟังก์ชันใน GDI สำหรับติดต่อกับอุปกรณ์แสดงผล ฟังก์ชันของGDI (ซึ่งทำงานไม่ขึ้นกับอุปกรณ์)จะผ่านค่าต่างๆที่ได้จากแอปพลิเคชัน ไปยังไดรเวอร์โดยพิจารณา ลักษณะของอุปกรณ์ และแปลงการเรียกใช้ฟังก์ชันของ GDI ให้เป็นกระบวนการสำหรับอุปกรณ์นั้น

การเก็บคอนเท็กซ์อุปกรณ์

ในบางครั้งอาจจำเป็นต้องมีการเก็บค่าต่างๆที่คอนเท็กซ์อุปกรณ์นั้นกำลังใช้งานอยู่ เพื่อนำมาใช้ภายหลัง ฟังก์ชันที่ทำหน้าที่ในการเก็บและการนำกลับสถานะของดีไวซ์คอนเท็กซ์คือ SaveDC และ RestoreDC

การลบคอนเท็กซ์อุปกรณ์

เราสามารถลบคอนเท็กซ์อุปกรณ์ที่สร้างขึ้นหลังจากที่ไม่ต้องการใช้งานแล้วได้ด้วยฟังก์ชัน DeleteDC ซึ่งจะเป็นการคืนทรัพยากรต่างๆกลับให้ WINDOWS

คอนเท็กซ์อุปกรณ์ในหน่วยความจำ

การสร้างคอนเท็กซ์อุปกรณ์ในหน่วยความจำ จะเป็นการกันหน่วยความจำไว้ส่วนหนึ่งเป็นอุปกรณ์เทียม โดยให้มีลักษณะและการปฏิบัติต่อ เหมือนกับคอนเท็กซ์อุปกรณ์ที่ต่ออยู่กับอุปกรณ์จริง จากนั้นเมื่อมีการสร้างรูปบิตแมพลงไปในคอนเท็กซ์อุปกรณ์แบบนี้ ก็สามารถใช้ฟังก์ชันเพื่อก๊อปปี้ข้อมูล ไปลงยังคอนเท็กซ์อุปกรณ์จริงๆได้ทันที หรือในทางกลับกันจะนำข้อมูลของคอนเท็กซ์อุปกรณ์จริงมาลงในคอนเท็กซ์อุปกรณ์ในหน่วยความจำก็ได้ ฟังก์ชันที่ใช้ในการสร้างดีไวซ์คอนเท็กซ์แบบนี้คือ CreateCompatibleDC แต่อุปกรณ์ที่จะถ่ายเทข้อมูลกับคอนเท็กซ์อุปกรณ์แบบหน่วยความจำนี้ได้ต้องสนับสนุนการสร้างรูปบิตแมพด้วยฟังก์ชัน BitBlt เท่านั้น

2.6 สี

ฟังก์ชัน GDI ที่มีการสร้างปากกาหรือแปรงทาสีส่วนมากต้องการข้อมูลที่ใช้ในการระบุสีของเครื่องมือ โดยอยู่ในโครงสร้างข้อมูล COLORREF วิธีในการบอกสีมี 3 วิธีคือ

- ระบุค่า RGB(Red Green Blue) โดยตรง
- ระบุจานสี(Color Palette) ที่ต้องการใช้
- ระบุค่า RGB ที่สัมพันธ์กับจานสี

วิธีแรกนั้นจะกำหนดเป็นความเข้มของสีแดง เขียวและน้ำเงินที่จะมาผสมเป็นสีที่ต้องการ โดยค่าความเข้มต่ำสุดคือ 0 และสูงสุดคือ 255 ไบต์ที่หนึ่ง(ไบต์ต่ำสุด)จะเป็นค่าความเข้มของสีแดง ไบต์ที่สองเป็นค่าความเข้มของสีเขียว ไบต์ที่สามเป็นค่าความเข้มของสีน้ำเงิน และไบต์ที่สี่ต้องมีค่าเป็น 0 เสมอ ตัวอย่างเช่น ถ้าต้องการสีน้ำเงิน ก็จะมีค่าเป็น 0X00FF0000 หรือถ้าเป็นสีเขียวก็จะเป็น 0X0000FF00 เป็นต้น การกำหนดสีวิธีนี้ฟังก์ชัน GDI จะส่งไปยังใครเวอร์ของอุปกรณ์โดยตรง ซึ่งใครเวอร์จะเป็นตัวเลือกสีที่ใกล้เคียงกับที่ต้องการมากที่สุดมาใช้ การบอกค่าของสีที่ใกล้เคียงค่าของสีที่ต้องการให้ใช้ฟังก์ชัน GetNearestColor

แต่ถ้าอุปกรณ์ที่ต้องการเป็นพล็อตเตอร์ ตัวใครเวอร์ก็จะแปลงสีที่ต้องการให้เป็นหมายเลขของปากกาพล็อตเตอร์ที่มีสีใกล้เคียงแทน

ในบางอุปกรณ์ที่ไม่สามารถแสดงสีได้มากนัก ก็จะใช้วิธี "dither" คือ การสร้างรูปแบบของจุดเข้าไปในสีด้วย เพื่อสร้างความแตกต่างได้ด้วยสีเดียวกัน ถ้าอุปกรณ์นั้นเป็นโมโนโครมลักษณะที่แสดงออกจะเป็นสามลักษณะคือ ขาว ดำ และรูปแบบของจุด โดยที่ค่าผลรวมของค่า RGB เป็น 0 สีที่แสดงออกมาจะเป็นสีดำ(เลือกใช้แปรงสีดำ) ถ้ามีค่าเป็น 765 ก็จะเป็นสีขาว แต่ถ้าค่าอยู่ระหว่าง 0 และ 765 ก็จะเป็นรูปแบบของจุด

ฟังก์ชันที่ดึงค่าสีแดง เขียว และน้ำเงินออกจากโครงสร้าง COLORREF คือ GetRValue, GetBValue และ GetGValue

2.7 จานสี

ปกติแล้วจอภาพส่วนมากสามารถแสดงสีได้จำนวนมาก แต่จำนวนสีที่แสดงได้พร้อมกันในหนึ่งหน้าจอจะมีไม่มากนัก เช่น จอที่แสดงสีได้ 262,000 สี อาจแสดงสีได้พร้อมกันเพียง 256 สีเท่านั้น ทั้งนี้ก็เนื่องมาจากข้อจำกัดของการแสดงผล ในกรณีส่วนแสดงผลจะมี "จานสี" (Palette) เพื่อเก็บว่ามีสีอะไรที่กำลังถูกแสดงอยู่ เมื่อใดก็ตามที่มีการร้องขอใช้สีมากกว่าที่จานสีจะรับได้ สีที่ร้องขอมาใหม่ก็จะไปทับของเดิม ทำให้ภาพที่กำลังแสดงอยู่บนจอมีสีที่ผิดเพี้ยนไปได้

ในสภาวะของ WINDOWS จะมีหลายแอปพลิเคชันทำงานพร้อมๆกัน ซึ่งแต่ละแอปพลิเคชันก็จะมีวินโดว์ของตนเอง ปัญหาอยู่ตรงที่ว่าวินโดว์อาจไม่สามารถใช้สีมากเท่าที่อยู่ในจานสีได้ ทั้งนี้ก็เพราะมีวินโดว์อื่นกำลังใช้งานสีต่างกันไปอยู่เช่นกัน ดังนั้น WINDOWS จึงให้วินโดว์ที่แอกทีฟนั้น สามารถใช้สีมากเท่าที่จอ นั้นจะแสดงได้ในครั้งหนึ่งๆ(คือตามขนาดของจานสี) ส่วนวินโดว์ที่ไม่ได้แอกทีฟนั้นจะถูกเปลี่ยนสีไปให้ใกล้เคียงกับสีที่มีอยู่ในจานสีปัจจุบัน(ที่วินโดว์ที่แอกทีฟใช้งานอยู่)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่วารณใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่องของจานสี

จานสีเป็นเครื่องมือ ไม่ขึ้นกับอุปกรณ์แสดงผลอีกอย่างหนึ่ง ทำหน้าที่เกี่ยวกับการจัดการให้แอปพลิเคชันสามารถใช้สีที่มีอยู่จำกัดในงานสีของระบบ(ของส่วนแสดงผลของเครื่อง)ได้อย่างเต็มที่

การที่จะให้แต่ละแอปพลิเคชันสามารถใช้งานงานสีของระบบได้เต็มที่นั้น WINDOWS จานสีทางตรรก (logical palette) ขึ้นมาให้แก่วินโดว์ของแอปพลิเคชันและเมื่อต้องการใช้งานสีของแอปพลิเคชันก็จะมาเพิ่มสีในงานสีทางตรรกของวินโดว์นี้แทน แต่ว่าอาจจะมีหลายวินโดว์ทำงานอยู่พร้อมกันในขณะหนึ่งๆ ทำให้ผลรวมของจำนวนสีในแต่ละงานสีของวินโดว์อาจมีมากกว่างานสีของระบบก็ได้ WINDOWS จึงต้องมีการจัดการในเรื่องนี้ด้วย โดยจะพยายามจับคู่สีที่มีอยู่ในงานสีของระบบเข้ากับงานสีของวินโดว์ ถ้ามีไม่ตรงกันก็จะเพิ่มสีนั้นลงไปในงานสีของระบบ แต่ถ้างานสีของระบบมีสีอยู่เต็มแล้วก็จะใช้วิธีหาสีที่ใกล้เคียงกันมากที่สุด(จะมีสี 20 สีมาตรฐานที่สามารถจับคู่ได้เสมอ) โดยปกติ WINDOWS จะพยายามให้วินโดว์ที่แอกทีฟอยู่ได้สีที่ใกล้เคียงความต้องการของคนมากที่สุด ส่วนที่ให้ความสำคัญรองลงมาคือวินโดว์ที่ได้อินพุทโฟกัสอยู่

* **เพิ่มเติม** * จานสีทางตรรก(logical palette)นั้นเป็นงานสีที่ให้วินโดว์ใช้อย่างเต็มที่และสามารถใส่สีได้ตามความพอใจ แต่จะมีใช้งานสีจริงที่อยู่บนการ์ดแสดงผลซึ่งจะมีเพียงอันเดียว WINDOWS จะคอยจัดการให้ทุกวินโดว์รู้สึกเหมือนกันว่าได้ใช้งานสีเพียงผู้เดียวโดยใช้วิธีการหาสีที่จะมาทดแทน ได้ตามวิธีที่กล่าวมาแล้ว

การใช้งานงานสี

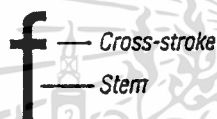
ก่อนที่จะวาดสิ่งต่างๆลงบนอุปกรณ์แสดงผล ต้องมีการสร้างงานสีของวินโดว์ขึ้นมาด้วยฟังก์ชัน CreatePalette และเลือกเข้าไปในคอนเท็กซ์อุปกรณ์ด้วย SelectPalette เสียก่อน ฟังก์ชันที่สามารถรับพารามิเตอร์เป็นค่าสีได้ก็สามารถรับเป็นดัชนีของงานสีได้เช่นกัน การกำหนดเป็นดัชนีของงานสีจะใช้ตัวแปรแบบ long โดยที่บิตแรกของไบต์สูงมีค่าเป็นหนึ่งในห้าและดัชนีของงานสีก็จะเก็บอยู่ในไบต์ค่า เช่น 0X01000005 จะเป็นการเลือกใช้สีของงานสีในช่องที่ 5 จากนั้นใช้มาโครที่ชื่อ PALETTEINDEX ในการเปลี่ยนดัชนีงานสีให้เป็นค่าของสี COLORREF เพื่อใช้เป็นพารามิเตอร์ให้กับฟังก์ชัน GDI ต่างๆ

บทที่ 3

ตัวอักษร (Font)

3.1 ส่วนประกอบของตัวอักษร

Windows graphics device interface (GDI) แบ่งตัวอักษรเป็นกลุ่ม (family) โดย แต่ละกลุ่มประกอบด้วย ตัวอักษรซึ่งมีการออกแบบร่วมกัน กลุ่มจะถูกแบ่งโดย “Stroke” และ “Serif” โดย stroke ถึง เส้นทางแนวนอน หรือแนวตั้ง โดยเส้นในทางแนวนอนเรียกว่า “Cross-stroke” เส้นหลักทางแนวตั้งนั้นเรียกว่า “Stem”



f — Cross-stroke
— Stem

Serif คือ เส้นสั้นๆ ที่อยู่ตอนปลายของ Stroke หลักของตัวอักษร ตัวอักษรที่ไม่มี Serif เรียกว่า “San-Serif”



Serif — M M — Sans-serif

ภายใน ครอบครัว(family)ของตัวอักษรจะแบ่งโดย รูปแบบที่ปรับเปลี่ยนไป ซึ่งทั่วไป จะเกี่ยวกับ น้ำหนัก(Weight) และความลาดเอียง(Slant) โดยที่น้ำหนัก (weight) จะแยกโดยสิ่งที่ประกอบกัน เช่น “Extralight”, “Light”, “Demi”, “Demi bold”, “Book”, “Bold”, “Heavy bold”, “Extra bold” และ “Black” และ ความเอียงแบ่งแยกโดย “Roman”, “Italio” และ “Oblique” โดยที่ Roman font คือ ตัวอักษรที่ตั้งตรง

Oblique font จะเอียง และ Italic font จะรวมเอาคุณสมบัติของทั้ง Roman และ Oblique มารวมกัน กลุ่มของตัวอักษรโดยปกติ จะไม่ได้รวม Italic และ Oblique font ไว้

GDI จะมีกลุ่มของชื่อที่ใช้อยู่ 5 กลุ่ม โดยแบ่งตามแบบของตัวอักษร สำหรับชื่อที่ 6 (FF_DONTCARE) จะใช้ได้กับ Default font โดยพิจารณาแต่ละชื่อดังนี้

| ชื่อ | ลักษณะ |
|---------------|--|
| FF_DECORATIVE | เป็นตัวอักษรที่มีความแปลกใหม่ เช่น Old English |
| FF_DONTCARE | เป็นชื่อของกลุ่มต่างๆ ไปจะใช้เมื่อรูปแบบของตัวอักษรไม่ได้ถูกบรรจุไว้ |
| FF_MODERN | เป็นตัวอักษรที่มีความกว้างของ Stroke คงที่ โดยไม่มี Serif เป็นตัวอักษรที่มีระบบคงที่ และใช้ในปัจจุบัน เช่นจะประกอบไปด้วย Pica, Elite และ Courier New |
| FF_ROMAN | เป็นตัวอักษรที่ความกว้างของ Stroke สามารถเปลี่ยนแปลงได้โดยมี Serif เช่น Times New Roman |
| FF_SCRIPT | ถูกออกแบบให้มีลักษณะเหมือนเขียนด้วยมือ เช่น Script และ Cursive |
| FF_SWISS | เป็นตัวอักษรที่ความกว้างของ Stroke เปลี่ยนแปลงได้แต่ไม่มี Serif เช่น Arial |

กลุ่มของชื่อใน GDI ไม่ได้มีลักษณะเช่นนี้เสมอไป เนื่องมาจากการแบ่งของ การพิมพ์(Typographic)

3.2 หน่วยและการวัดตัวอักษร

ทั้งส่วนที่สามารถมองเห็นได้และไม่ได้ของตัวอักษรมีผลต่อการวัดทั้งคู่ ส่วนที่สามารถมองเห็นได้ เรียกว่า "Glyph" และส่วนที่เรามองไม่เห็น มีลักษณะเป็น สีเหลี่ยมจัตุรัส และประกอบด้วยตัวอักษรเรียกว่า "Character Cell" โดย Character cell จะเริ่มที่มุมบนด้านซ้าย เมื่อจะแสดงตัวอักษร ตำแหน่งของ GDI อยู่ที่ จุดเริ่มต้นของ Character cell แรก (นี่คือคุณสมบัติของ GDI โดยสามารถเปลี่ยนได้ตลอดเวลาโดยใช้ ฟังก์ชัน SetTextalign) หน่วยในการวัดต่างๆไป ของตัวอักษรคือ จุดในอุตสาหกรรม คอมพิวเตอร์ 1 จุดมีค่า 1/72 นิ้ว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ความสูงของตัวอักษรใน Windows กำหนดโดย "Twips" ซึ่งมีขนาด 1/20 ของจุด (1/1440)นิ้ว ขนาดของจุด เมื่อการวัดต่างๆนั้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงใจ ของเอกสารทุกครั้งที่มีการนำไปใช้

จะใช้อ้างอิงขนาดของ Character cell และจะใช้เป็นขนาดย่อยๆของตัวอักษรเท่านั้น Glyphs ที่แตกต่างกัน 12 จุด จะทำให้มีความสูงแตกต่างกันอย่างเห็นได้ชัด ดังตัวอย่างแสดงความสูงที่ไม่เท่ากันของตัวอักษรจาก Times New Roman, Palatino , และ Arial ต่างกัน 12 จุดในแต่ละตัว

AAAaaaBBBbbbCCCcccDDDddd

เครื่องมือวัดของ Character-cell จะมีผลในการสร้างตัวอักษร

เครื่องมือวัด

ลักษณะ

Ascent

คือระยะห่างจาก Baseline ถึงส่วนสูงของตัวอักษร Ascender ของตัวอักษรคือส่วนของตัวอักษรที่สูงกว่า Baseline สำหรับ Window นั้น Ascent คือระยะ ห่างจาก Base line ถึง Character cell ที่สูงที่สุด โดยจะรวม Whitespace เข้าไปด้วย Typographic Ascent ในคำจำกัดความคือตัวอักษรที่สูงที่สุด

Base line

คือเส้นหลักของตัวอักษรทั้งหมด โดย Base line จะเป็น จุดที่ตั้งตรงที่สุดของตัวอักษร Desender คือ ส่วนของตัวอักษรที่ต่ำกว่า Base line เช่น หางของ "g" สำหรับใน cell ล่างสุดโดยรวม White space ด้วย สำหรับ Typography Desent คือ ส่วนที่เพิ่มออกมาจาก Base line

Height

คือส่วนในทางแนวตั้งที่ของตัวอักษร โดยเป็นผลรวมของ Ascent, Descent และ ส่วน ที่สำคัญภายในของตัวอักษร

Width

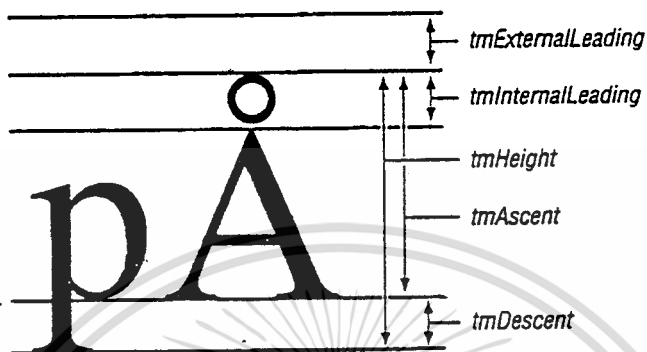
คือ ส่วนในทางแนวนอน สำหรับ Character cell ในตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า GDI จะส่งค่าความกว้าง สำหรับเฉลี่ย Character cell ใน ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|------------------|--|
| External leading | <p>ตัวอักษร หรือ สำหรับความกว้างของ Character cell ความกว้างของตัวอักษรนั้นสามารถ แก้ไขได้ในแต่ละตัวอักษร คือ ช่องว่างระหว่างแถวของตัวอักษร External leading จะไม่ใช่ส่วนของ Character cell เมื่อ Character line spacing คือ External leading ที่เพิ่มเข้าไปใน Internal leading ใน Windows นั้น External leading ไม่สามารถมีค่าเป็นลบได้</p> |
| Internal leading | <p>คือความแตกต่างระหว่างความสูงของ Character glyph สำหรับตัวอักษร และ ความสูงของ Character cell สำหรับตัวอักษร โดยจะใช้ Internal leading ในการกำหนดขนาดของจุดของตัวอักษร ขนาดของจุด คือความสูงของ Character cell ที่ต่างออกจาก Internal leading</p> <p>ในการใช้งานบางอย่างมีการใช้งานที่ไม่ถูกต้องของ Internal leading โดย Internal leading จะใช้เฉพาะ Diaritical mark ไม่ควรใช้ในการเว้นช่องว่างในการเคลื่อนย้ายจากหน้าแรกใน กระดาษ</p> |
| Overhang | <p>คือลักษณะสมบัติของ glyph บางตัวที่อยู่ในช่องว่างทางแนวนอนเดียวที่อยู่ติดกับ glyph ทุกตัวอักษรทุกตัว ในตัวอักษรแบบ Italic ใช้ overhang ในการเก็บตัวอักษรที่สัมพันธ์กันเข้าด้วยกัน</p> |
| Pitch | <p>คือ ชนิดโดยทั่วไปของการเว้นช่องว่างทางแนวนอนของตัวอักษร โดย Character cell ใน Fixed-pitch จะมีขนาดเดียวกันหมด แต่ใน Variable-pitch จะปรับเปลี่ยนตามความกว้างของ glyph</p> |

External leading สำหรับตัวอักษรจะมีลักษณะเฉพาะโดยการออกแบบของผู้ออกแบบตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากรูปแสดง Internal และ External leading และส่วนอื่นๆที่เกี่ยวข้องกับความสูง (height) ของตัวอักษรซึ่งในการเริ่มต้นคือตัวอักษร 'tm' คือสมาชิกของโครงสร้างข้อมูล TEXTMETRIC

3.3 ตัวอักษรใน Windows

ตัวอักษรที่ใช้งานของ Windows มี 3 ชนิด ตัวอักษรแบบ Raster, Vector และ TrueType โดยใน Version ก่อนของ Window มี font 2 ชนิด คือ raster และ Vector และ Window 3.1 จะมีการใช้ TrueType font เพิ่มเข้ามา

Raster font เป็นการเก็บในลักษณะบิตแมพ บิตแมพนี้ออกแบบสำหรับอุปกรณ์เอาต์พุตของโดยเฉพาะ GDI จะใช้สร้างตัวอักษร แบบ Bold, Italic, Underline และ Strikethrough สำหรับ Raster font อย่างไรก็ตามผลของมันจะไม่สวยงามเสมอไป เมื่อ GDI ต้องเปลี่ยนขนาดของ raster font การเปลี่ยนชื่อก็เป็นปัญหาเกี่ยวกับความสวยงามของ font ด้วย

Raster font ใช้สำหรับ กรณีพิเศษ เมื่อ TrueType font ไม่สามารถทำได้ สิ่งที่เป็นไปได้ก็อย่างหนึ่งในการใช้ Raster font ได้มาจาก ค่าจำนวนมาก ของ Raster font ซึ่งพบบ่อย ๆ บนระบบของผู้ใช้ ซึ่งการใช้ Raster font ควรจะใช้สำหรับชื่อของ font ชนิดพิเศษ และ TrueType font ไม่สามารถแสดงได้

เมื่อการร้องขอ Italic หรือ Bold ซึ่งไม่สามารถทำได้ GDI จะสร้างตัวอักษร โดยการแปลงตัวอักษรในลักษณะบิตแมพเมื่อมีการใช้งาน raster font เท่านั้น การร้องขอขนาดของจุดจะไม่สามารถทำได้ GDI จะแปลงเมื่วารมณ์ใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตแมพในการสร้างตัวอักษรด้วย เพราะ กลุ่มของ TrueType ประกอบด้วย Bold, Italic และ Bold Italic และ เพราะ TrueType จะมีหน่วยทุกการร้องขอขนาดของจุด และ GDI จะไม่สร้างตัวอักษรบ่อย ๆ เหมือน Window version ก่อน

Windows version 3.1 ประกอบด้วย Raster font ชุดใหม่ ซึ่งชุดนี้ ถูกเรียก small Font สำหรับใช้ที่ Resolution น้อยกว่า 8 จุด แม้ว่า TrueType สามารถระบุขนาดที่ต่ำกว่า 8 จุดได้ glyph ซึ่งมีขนาดเล็ก จะทำให้การอ่านยากสำหรับการใช้งานที่จำเป็น เพราะ glyph ที่เล็ก จะประกอบด้วย ส่วนอื่น ๆ หรือรายละเอียดที่เล็กมาก ดังนั้น raster small font จะมีประสิทธิภาพกว่า TrueType font

Vector font จะถูกเก็บเป็นกลุ่ม GDI มันเป็น Time-consuming ในการกำหนด แต่มันจะใช้สำหรับอุปกรณ์ เช่น Plotter ซึ่งบน Bitmap character ไม่สามารถใช้ได้ (โดยการลากเส้น GDI สามารถ จำลอง Vector font บน อุปกรณ์ซึ่ง ไม่สามารถ คอบสนองได้โดยตรงได้) ในข้อแนะนำก่อนของ TrueType font , Vector font สามารถใช้เพื่อตัวอักษรที่มีขนาดใหญ่มากหรือมีการบุคเบี้ยว หรือต้องการตัวอักษร ที่ทำมุมต่างๆ กับเส้น Base line

TrueType font ถูกเก็บในกลุ่มของจุดและส่วนที่เกี่ยวข้องซึ่งจะเป็นตัวกำหนด Outline ของตัวอักษรเมื่อมีการร้องขอ TrueType font, TrueType rasterizer จะใช้ Outline และ Hints ในการสร้าง Bitmap ของขนาดที่ถูก ร้องขอ

Default font สำหรับคำอธิบาย System font ในการกำหนดสัดส่วนของ Raster tool ที่ใช้แสดงตัวอักษร ใน Windows character set โดย Windows จะใช้ System font สำหรับ Menu, Window title และคำอธิบายอื่น

มันเป็นไปได้ที่จะมีตัวอักษรหลาย ๆ อย่าง ในระบบซึ่งมีสิ่งเดียว (เช่น Courier device font และ Courier GDI raster font) อย่างไรก็ตามการใช้งานจะถูกเรียกซึ่งได้ไม่เกินครั้งเดียว แทนที่มันจะลบอีกชื่อหนึ่งทิ้งไป การใช้งานไม่สามารถควบคุมตัวอักษรที่ถูกแสดงเมื่อมีการเรียกตัวอักษร ที่มีชื่อเหมือนกันโดยการใช้ IfOutPrecision ส่วนอยู่ในโครงสร้างข้อมูล LOGFONT

ตัวอักษรแบบทวิไทป์(TrueType)

ความสามารถของตัวอักษรแบบทวิไทป์ใน Window 3.1 จะมีมากกว่า version ก่อนๆ โดยมีความสามารถในการกำหนดสัดส่วนและการวนรอบ มันจะยอมให้มีการใช้ตัวอักษรเพียงแบบเดียวบนจอภาพและเครื่องพิมพ์ และมันจะมี รูปแบบของตัวอักษร ในเครื่องพิมพ์และอุปกรณ์ใช้งานอื่น 13 รูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| กลุ่มของตัวอักษร | ชื่อ | ลักษณะ |
|------------------|-----------------------------|---------------------------|
| Arial | Arial | San serif, variable pitch |
| | Arial Bold | San serif, variable pitch |
| | Arial Italic | San serif, variable pitch |
| | Arial Bold Italic | San serif, variable pitch |
| Courier New | Courier New | Serif, Fixed pitch |
| | Courier New Bold | Serif, Fixed pitch |
| | Courier New Italic | Serif, Fixed pitch |
| | Courier New Bold Italic | Serif, Fixed pitch |
| Symbol@ | Symbol | N/A |
| Times New Roman | Times New Roman | Serif, variable pitch |
| | Times New Roman Bold | Serif, variable pitch |
| | Times New Roman Italic | Serif, variable pitch |
| | Times New Roman Bold Italic | Serif, variable pitch |

เทคโนโลยีของตัวอักษรแบบทรูไทป์นั้นจะมีข้อพิเศษคือจะไม่ต้องเสียเวลาเพิ่มในการออกแบบและจะไม่บังคับในการแก้ไขการเขียนในฟังก์ชัน

| ฟังก์ชัน | หน้าที่ |
|----------------------------|---|
| CreateScalableFontResource | สร้าง Font resource file สำหรับตัวอักษรแบบทรูไทป์ |
| EnumFontFamilies | แก้ไขตัวอักษรบนอุปกรณ์ |
| GetCharABCWidths | แก้ไขความกว้างของส่วนที่ติดกันของตัวอักษร |
| GetFontData | แก้ไข Font-metric data จาก TrueType font file |
| GetGlyphOutline | แก้ไขลักษณะข้อมูลของตัวอักษร |
| GetOutlineTextMetrics | แก้ไข Font metric สำหรับตัวอักษรแบบทรูไทป์ |
| GetRasterizerCaps | กำหนดในการติดตั้ง TrueType |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ลักษณะเฉพาะของข้อความและตัวอักษร

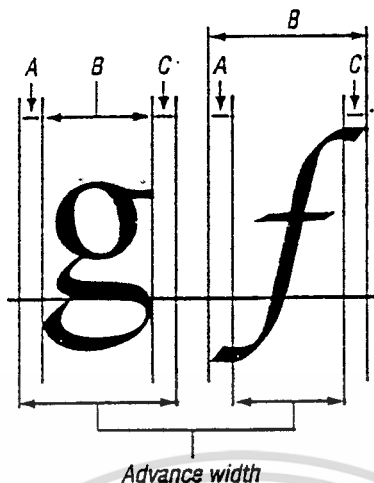
การเว้นช่องว่างของเส้นและตัวอักษร

เป็นการยากที่จะกำหนดตำแหน่งที่แน่นอนของตัวอักษรโดยเฉพาะในรูปแบบตัวหนาและตัวเอียง ใน Windows จะมีการแก้ไขความกว้างของตัวอักษรซึ่งจะประกอบด้วยช่องว่างข้างทั้งสองของ Glyph ดังรูป



เราสามารถควบคุมระยะห่างของตัวอักษรได้จากการใช้ ABC characters spacing โดย "A" คือ ความกว้างที่เพิ่มเข้ามาทางด้านหน้าของตำแหน่ง glyph "B" คือความกว้างของ glyph และ "C" จะอยู่ทางขวาของ glyph ดังนั้น ความกว้างทั้งหมด คือ $A+B+C$ โดยที่ค่าของ A และ C มีค่าเป็นลบได้ ดังนั้นตัวอักษรสามารถ ยืดเกินออกมาหรือ หดเข้าไปได้ เช่น จากรูป A,B และ C เป็นบวกสำหรับตัวอักษร "E" แต่ A และ C จะเป็นลบเมื่อเป็นตัวอักษร "F"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เราจะใช้ฟังก์ชัน `GetCharABCWidths` แก่ไขค่า A และ B และ C และสามารถกำหนดตามขนาดของตัวอักษรให้แตกต่างกันในแต่ละบรรทัดในการใช้ `Ascender` และ `Descender` ของตัวอักษร โดยสามารถแก้ไขค่านี้ได้การเรียกฟังก์ชัน `GetTextMetrics` และมีการตรวจสอบที่ `tmAscent` และ `tmDescent` ซึ่งเป็นสมาชิกของโครงสร้างข้อมูล `NEWTEXT METRIC` ค่าสูงสุดของ `ascent` จะแตกต่างจาก `typographic ascent` และ `descent` ในตัวอักษรแบบทรูไทป์ ซึ่งส่วนสูงสุดของ `glyph` คือ "f" และ ล่างสุดคือ "g" ส่วนที่ยื่นเกินออกมาโดยรอบตัวอักษรจะถูกจำกัดโดย `Straight edge` เราสามารถแก้ไขค่า `typographic ascent` และ `descender` โดยการเรียกฟังก์ชัน `GetOutlineTextMetric` และการตรวจสอบจาก `otmAscent` และ `otmDescent` ซึ่งเป็นสมาชิกของโครงสร้างข้อมูล `OUT LINE TEXT METRIC`



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงค่าที่แตกต่างกันในทางแนวนอน ใช้การส่งค่ากลับของโครงสร้างข้อมูล NEWTEXTMETRIC และ OUTLINETEXTMETRIC (ซึ่งขึ้นต้นด้วย "otm" เป็นสมาชิกของ OUTLINETEXTMETRIC)

หน่วยนิ้วทางตรรกและทางกายภาพ

Logical inch คือ หน่วยของตัวอักษรสำหรับ Windows ที่อ่านง่าย พบจอภาพ โดยทั่วไปมีขนาดใหญ่กว่า physical inch 30-40% 10-point font บนจอภาพจะใหญ่กว่า 10-point font ที่เครื่องพิมพ์ เพื่อให้มีขนาดตัวอักษรบนจอใหญ่พอที่จะอ่านได้ และการอ่านข้อความจากจอภาพจะใกล้กว่าอ่านจากกระดาษที่พิมพ์ข้อความแล้วโดยเครื่องพิมพ์ แต่การแสดงบนจอภาพจะกระทำในรูปแบบของสากลได้หากมีการแสดงร่วมกับของข้อความและรูปภาพ

เราสามารถแก้ไขขนาดทางกายภาพ(Physical)จากการเรียก GetOutlineTextMetrics ฟังก์ชัน การกำหนดของอุปกรณ์ทำได้โดยฟังก์ชัน GetDeviceCaps และมันจะส่งค่ากลับไปส่งยังตัวอักษรทั้งทางกายภาพและตรรก.

ขนาดของตัวอักษร

เราจะใช้ MM_Text mapping mode แทน MM_TWIPS เพราะ MM_TEXT สามารถแปลงขนาดทางตรรกได้ แต่ หน่วยทางตรรกจะถูกแปลงเป็นพิกเซล

การกำหนดขนาดของจุดสำหรับตัวอักษรอันดับแรกคือแปลงค่าส่งกลับไปโครงสร้างข้อมูล NEWTEXTMETRIC หรือ OUTLINETEXTMETRIC ในการใช้ขนาดของนิ้วทางตรรก (Logical inch) สำหรับอุปกรณ์ เช่น MM_TEXT unit ใช้ตัวอักษรที่มีความสูงของ Character cell (tmHeight) 12 และ internal leading (tmInternalLeading) เท่ากับ 2 เราจะลดขนาดความสูงของ Character cell โดยแปลงได้เป็น 10 unit (pixel)

การแปลงค่าเป็นขนาดจุดทาง typographic (มีค่า $1/72$ นิ้ว) โดยการใช้ GetDeviceCaps ฟังก์ชันกำหนดขนาดทางแนวดิ่ง และ ความละเอียดของจอและจำนวนของ พิกเซลต่อหนึ่งนิ้วทางตรรก ตัวอย่าง เช่น ค่าใน MM_TEXT mapping mode ต้องการ 12-point font มันจะใช้ค่าซึ่งเป็น algorithm ใน IEMHeight ซึ่งเป็นสมาชิกของโครงสร้างข้อมูล LOGFONT

$$-1 \times ((\text{LODPixelsY} \cdot 12) / 72)$$

สาเหตุในการใช้ค่าลบใน IEMHeight เพราะ ค่าความสูงของ glyph ไม่ใช่ของ character cell ค่าของ LOGPIXELSY จะถูกส่งกลับโดยเรียก GetDeviceCaps ฟังก์ชัน ขนาดของจุดที่ถูกร้องขอ คือ 12 และ สมาชิกของจุดในหนึ่งนิ้วทางกายภาพ ถึง 72 ในแนวเดียวกัน การใช้ algorithm ในการกำหนดขนาดของจุดของตัวอักษรจะถูกส่งกลับไปโครงสร้างข้อมูล NEWTEXTMETRIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกหนึ่งที่มีผลเห็นแต่เพียงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 เทคโนโลยีของตัวอักษรแบบทรูไทป์

การออกและการกำหนดสเกลของตัวอักษร

Raster fonts ถูกออกแบบให้มีความสวยงาม และอ่านง่าย โดยการกำหนดอัตราส่วนของรูปร่างโดยเฉพาะ aspect ratio คือ อัตราส่วนของความกว้าง และความสูงของพิกเซล รูปร่างทาง Digital ของตัวอักษร คือ ลักษณะทางอุดมคติของตัวอักษร

Windows เอง จะกำหนดให้ Aspect-ratio filter ในการเลือกออกแบบของตัวอักษรจากที่มีอยู่ทั้งหมด ฟังก์ชัน GetAspectRatioFilter จะใช้แก้ไขการตั้งค่าของ Current aspect-ratio Filter และสามารถใช้ ฟังก์ชัน SetMapperFlags ในการเปลี่ยน algorithm จากตัวอักษรทางกายภาพเป็นตัวอักษรทางตรรก

อัตราส่วนรูปร่างหน้าตา(aspect-ratio) จะไม่มีการวิเคราะห์สำหรับการกำหนดสัดส่วนของตัวอักษร ขนาดของ raster font ขนาด em square ของตัวอักษรแบบทรูไทป์ถูกใช้เมื่อมีการกำหนดสัดส่วนขนาดจุดของตัวอักษร (em square คือ สี่เหลี่ยมจัตุรัสที่มีความกว้างเท่ากับความกว้างของตัว "M") เพราะความกว้างของ em square กำหนดใน พิกเซล มันสามารถส่งค่าของขนาดของจุดใน device unit เช่น ตัวอักษรอ้างอิงที่ 50-ppem (pixel per emsquare) โดย ขนาดของพิกเซล เป็นตัวกำหนดขนาดของจุดทางกายภาพ เช่น 75 -ppem ขนาดตัวอักษรบน 300 - DPI device เป็น 18-point font โดยบน 150-DPI device เป็น 36 -point font ค่าของ pixel ใช้กำหนด ขนาดของจุดในคอมพิวเตอร์ โดย การใช้ความละเอียดของอุปกรณ์เอาท์พุท และ ขนาดของ em square จากสูตร

$$PPEM = (\text{Point Size} / 72) * \text{Device Resolution}$$

12-point font บน 72-DPI screen คือ 12 PPEM โดยที่ 300-DPI device เป็น 50 PPEM

ตัวอักษรแบบทรูไทป์สามารถกำหนดสเกลให้ สม่่าเสมอและไม่สม่่าเสมอได้ ในการออกแบบการกำหนดสเกลที่สม่่าเสมอ คือ ความกว้างและรอบๆ ของตัวอักษรถูกกำหนดสเกล ในหน่วย PPEM ที่เหมาะสม

3.6 การใช้ตัวอักษรบนแอปพลิเคชัน

การใช้ Stock Font

ในการใช้งานต่างๆของตัวอักษรทั้งหมด จะจำเป็นต้องมี stock font สำหรับการแสดงข้อความ การใช้ stock font จะกำหนดชนิดของตัวอักษรเมื่อมีการเลือก Handle ให้กับ Device context font mapper ใช้ตัวอักษรทางตรรกสร้างตัวอักษรทางกายภาพ สามารถเลือก และใช้ตัวอักษรทางกายภาพ สำหรับแสดงข้อความได้

ตัวอักษร

ลักษณะ

| | |
|---------------------|---|
| ANSI_FIXED_FONT | กำหนด Fixed-pitch font พื้นฐานบน Windows character set Courier font คือ ชนิดที่ถูกใช้ |
| ANSI_VAR_FONT | กำหนด Variable-pitch font พื้นฐานบน Windows character set MS Sans serif คือ ชนิดที่ถูกใช้ |
| DEVICE_DEFAULT_FONT | กำหนดตัวอักษรให้กับอุปกรณ์ เพราะมันจะเกี่ยวข้องกันบน GDI font mapper ซึ่งจะแปลการร้องขอตัวอักษร โดยยอมให้ปรับเปลี่ยนความกว้างจากอุปกรณ์ไปยังอุปกรณ์ |
| OEM_FIXED_FONT | กำหนด Fixed-pitch font พื้นฐานบน OEM character set OEM character set จะปรับเปลี่ยนจากระบบไปยังระบบสำหรับ IBM computer และ compatibles, OEM font จะเป็นพื้นฐาน IBM PC character set ข้อความใน dialboxes ใน System font สังเกตว่าเราจะเห็นตัวอักษรอื่นๆ |

ตัวอย่าง การแก้ไข handle ของ Windows variable stock font โดยเลือกมันเข้าสู่ดีไวซ์คอนเท็กซ์

```

HFont hfnt, hOldFont;
hfnt = GetStockObject(ANSI_VAR_FONT);
if (hOldFont = SelectObject(hdc, hfnt) ;
TextOut(hdc, 10, 50, "Sample ANSI_VAR_FONT text", 26);
SelectObject(hdc, hOldFont) ;

```

ถ้า stock font อื่นๆไม่สามารถหาได้ GetStockObject จะส่งค่า handle ไปยัง System font (SYSTEM_FONT)

ฟังก์ชัน GetStockObject ใช้แก้ไข handle ของตัวอักษรทางตรงที่การทำงานใน MM_TEXT unit handle จะส่งค่ากลับโดย GetStockObject ยอมให้กำหนดความสูงที่ไม่เข้ากับความสูงของการร้องขอจากตัวอักษรทางตรงเมื่อมีการทำงานใน mapping mode อื่น ๆ นอกจาก MM_TEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างตัวอักษรทางตรรก(Logical font)

logical font คือ รายการของตัวอักษรซึ่งมีความสูง, ความกว้าง, character set และ รูปแบบ การสร้างตัวอักษรทางตรรกจะต้องสร้างตัวอักษรที่มีความเหมาะสมที่สุดในการใช้งาน font mapper จะใช้ตัวอักษรทางตรรกในการเลือก ตัวอักษรทงกายภาพที่เข้ากันที่สุด กับคุณสมบัติที่กำหนด

เราสามารถใช้ฟังก์ชัน CreateFont หรือ CreateFontIndirect ในการสร้างตัวอักษรทางตรรก เราใช้ CreateFontIndirect ในการกำหนดค่าให้กับโครงสร้างข้อมูล LOGFONT ฟังก์ชันที่จะส่งค่าให้ handle ของตัวอักษรทางตรรก สำหรับเลือก คิวส์คอนเท็กซ์ และ อุปกรณ์ที่ใช้

ตัวอย่างนี้เป็นฟังก์ชันใช้ handle ของคิวส์คอนเท็กซ์ชื่อของตัวอักษร และขนาดของจุดเป็นอินพุทในการสร้างตัวอักษรทางตรรกของการร้องขอ ขนาดและ ชื่อของรูปแบบ และ เลือกตัวอักษรในการกำหนดคิวส์คอนเท็กซ์

```

BOOL FAR PASCAL CratLogFont(hdc, pszFace, PointSize)
HDC hdc;
PSTR pszFace;
int PointSize;
{
    HFONT hfnt, hfntOld;
    PLOGFONT plf = (PLOGFONT) LocalAlloc(GPTR, sizeof(LOGFONT));
    if(GetMapMode(hdc) != MM_TEXT){
        TextOut(hdc, 100, -200, "Mapping mode must be MM_TEXT", 28);
        return FALSE;
    }
    plf -> lfHeight = -MulDiv(PointSize, GetDeviceCaps(hdc, LOGPIXELSY, 72);
    lstrcpy(plf -> lfFaceName, pszFace);
    hfnt = CreateFontIndirect(plf);
    hfntOld = SelectObject(hdc, hfnt);

    /* Use font for text output */

    LocalFree((LOCALHANDLE) plf);
    SelectObject(hdc, hfntOld);
    if(DeleteObject(hfnt))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return TRUE;

else

return FALSE;

}

```

หน่วยความจำสำหรับตัวอักษรทางตรงในตัวอย่างนี้กำหนดและเริ่มต้นที่ศูนย์ (โดยใช้ฟังก์ชัน LocalAlloc โดย GPTR constant) ตัวอักษรทางตรงถูกสร้างโดยฟังก์ชัน CreateFontIndirect และใช้ค่าเริ่มต้นสำหรับสมาชิกทั้งหมด ยกเว้น IHeight และ IFaceName ในตัวอย่างนี้ใช้ฟังก์ชัน MulDiv ในการแปลงขนาดของจุดเป็นค่าที่แตกต่างกันไปทางลบและกำหนดค่าของ IHeight ในการแปลงค่านี้สำคัญมาเพราะหนึ่งนิ้วทางตรงนั้นใหญ่กว่า หนึ่งนิ้วทางกายภาพ ฟังก์ชัน MulDiv ถูกร้องขอค่าขนาดของจุดโดยผลของการหารค่าของพิกเซลต่อหนึ่งนิ้วทางตรง โดยค่าของจุดใน หนึ่งนิ้วทางกายภาพ (72) ค่าลบที่กำหนดสำหรับ IHeight ที่แสดงในระบบนี้จะแปลงค่านี้เป็นค่าความสูงของ Character glyph ในตัวอักษร เมื่อค่าบวกนี้ถูกกำหนดถูก GDI แปลงเป็นความสูงของตัวอักษรของ Character cell มีใช้ใน Internal leading ค่าบวกที่ใช้สำหรับ IHeight ในการเลือกตัวอักษรที่พอดีกับความสูงที่กำหนด และใช้แก้ไขความสูงของเครื่องพิมพ์ ตัวอักษร จาก tmHeight ของโครงสร้างข้อมูล NEWTEXTMETRIC สัดส่วนความสูงของความละเอียดจะใช้ค่านี้สำหรับ IHeight สูตรข่งล่างนี้ คือ

$$IHeight = \frac{tmHeight * DPE \text{ of screen}}{DPI \text{ of printer}}$$

เมื่อกำหนด handle ของตัวอักษรทางตรงในการเรียกฟังก์ชัน SelectObject นั้น Font mapper ส่งค่ากลับไป handle ของตัวอักษรทางกายภาพที่เหมาะสมที่สุดสำหรับการร้องขอ

เมื่อต้องการ Raster font ทำโดยเรียกฟังก์ชัน EnumFontFamilies และตรวจสอบจาก Raster-Fonttype bit การใช้งานสามารถกำหนดชื่อของรูปแบบ(Typeface name)ในโครงสร้างข้อมูล LOGFONT เหมือนกับ Vector font ถูกเลือกโดยการตรวจสอบที่บิต RASTER_FONTTYPE และ TRUETYPE_FONTTYPE การใช้งานเหมือนกับการกำหนด Vector font โดยกำหนดที่ OEM_CHARSET ใน CharSet ของโครงสร้างข้อมูล LOGFONT

การแก้ไขข้อมูลของตัวอักษรนอกเหนือจากที่เลือกไปแล้ว

เราสามารถแก้ไขข้อมูลของตัวอักษรจากคีย์บอร์ดคอนเท็กซ์โดยการใช้ฟังก์ชัน GetTextMetrics, GetTextFace และ GetOutlineTextMetrics เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นาไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน `GetTextMetrics` จากโครงสร้างข้อมูล `TEXTMETRIC` ไปยังบัพเฟอร์ โครงสร้างข้อมูล `TEXTMETRIC` เป็น ลักษณะของตัวอักษรทางกายภาพซึ่งประกอบด้วยค่าขนาดและค่าเฉลี่ยของ `Character cell` ภายในตัวอักษร การเว้นช่องว่างระหว่างเส้นของข้อความ, ค่าของ `Character cell` ในตัวอักษร และ `Character set` บนตัวอักษรพื้นฐาน

การใช้งานตัวอักษรแบบทรูไทป์สามารถเขียนฟังก์ชัน `GetOutlineTextMetrics` เพื่อแก้ไขค่า ในโครงสร้างข้อมูล `OUTLINETEXTMETRIC`

มีการใช้โครงสร้างข้อมูล `TEXTMETRIC` บ่อยๆในการกำหนดช่องว่างระหว่างเส้นของข้อความสำหรับ ทำอย่างการคำนวณหาค่า `Single-line spacing` และ บวกค่าของ `tmHeight` และ `tmExternalLeading` โดย `tmHeight` กำหนดความสูงของแต่ละ `Character cell` และ `tmExternalLeading` กำหนดช่องว่างร่วมกันของการ ออกแบบตัวอักษรระหว่างส่วนล่างสุดของ `Character cell` และ ส่วนบนสุดของ `Character cell` คัดไป (ค่าที่ ถูกต้องสามารถแก้ไขสำหรับตัวอักษรแบบทรูไทป์จากโครงสร้างข้อมูล `OUTLINETEXIMETRIC` ในส่วนที่ สามารถเพิ่มค่าของ `otmAscent`, `otmDescent` และ `otmLineCap`

ตัวอย่างต่อไปนี้เป็นการเว้นช่องว่างต่างของ `single-spaced text`

```
TEXTMETRIC tm;
int LineSpacing , i, YIncrement;
GetTextMetrics (hdc, &tm);
LineSpacing = tm.tmHeight+tm.tmExternalLeading;
YIncrement = 50;
for.( i =0 ; i<4 ; i++)
TextOut(hdc, 10, YIncrement, "Single-line spacing", 19);
YIncrement += LineSpacing;
```

`GetTextFace` จะนำชื่อที่เหมือนกับรูปแบบของตัวอักษรที่ถูกเลือกไปไว้ในบัพเฟอร์และสามารถใช้ค่าที่ใน `Dialog box` และเมนู.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไขข้อมูลอื่นๆของตัวอักษรทางตรรก

เราสามารถแก้ไขข้อมูลของตัวอักษรโดยการกำหนด handle ในการเรียกฟังก์ชัน GetObject จะส่งค่าของตัวอักษรทางตรรกไปให้โครงสร้างข้อมูล LOGFONT

ตัวอย่างต่อไปนี้จะใช้ฟังก์ชัน GetObject แก้ไขค่าของตัวอักษรทางตรรก และ ตรวจสอบความเอียง(Italic) ของตัวอักษร

```
LOGFONT lf;
GetObject( hfnt, sizeof(LOGFONT) , &lf );
if ( lf.lfItalic )
    return TRUE ;
else
    return FALSE;
```

การแสดงความ

สามารถใช้ฟังก์ชัน ในการแสดงความดังนี้

ฟังก์ชัน

หน้าที่

| | |
|---------------|--|
| DrawText | รูปแบบของการเขียนข้อความในสี่เหลี่ยม รูปแบบของ DrawText โดยการขยายเข้าไปในช่องว่างให้เหมาะสม การวางข้อความที่เข้าไปในเส้นที่พอดีกับข้างในสี่เหลี่ยมมันไปใช้ฟังก์ชัน GDI และ อยู่ใน USER.EXE |
| ExtTextOut | เขียนข้อความเข้าไปใน กรอบสี่เหลี่ยม โดยสี่เหลี่ยมนี้สามารถ กำหนดให้ทึบได้ (โดยกำหนด สีของ background) และสามารถ ตัด บางส่วนออก |
| GrayString | แสดง Graytext โดยการเขียน Text memory bitmap, graying bitmap ลื่อนค่าจากรูปบิตแมพไปไปยังอุปกรณ์ GrayString เป็น ผลมาจากการเลือกแปรงทาสี(Brush)และพื้นหลัง(Background) มัน ไม่ใช่ฟังก์ชันของ GDI มันอยู่ใน USER.EXE |
| TabbedTextOut | แสดงข้อความ, วางขยาย Tab ที่กำหนดในแถวลำดับ(Array)ของ ตำแหน่งของ Tab-stop |
| TextOut | แสดงข้อความที่ตำแหน่งที่กำหนด |

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาเบเซอร์ระเขยนตนการค้ำ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน `ExtTextOut` เป็นฟังก์ชันในการแสดงข้อความที่เร็วที่สุด และฟังก์ชัน `DrawText` เป็นฟังก์ชันแสดงข้อความที่ช้าที่สุด (แม้ว่าจะมีรูปแบบให้เลือกมากที่สุด) การแทนการใช้ฟังก์ชัน `GrayString` ทำได้โดยการกำหนดให้สีของตัวอักษร(`Text color`) เป็นสีเทา(`Gray`) ดังนี้

```
dwColorPrevious = SetTextColor( hdc, GetSysColor (COLOR_GRAYTEXT);
```

การใช้สี

อันดับแรกคือการสร้างคิไวซ์คอนเท็กซ์ให้สีของข้อความ(`Text color`) คือสีตัว และ สีของพื้นหลัง(`Background`) คือสีขาว เราสามารถเพิ่มสีให้กับ ข้อความโดยทางกำหนดข้อความ และ สีพื้นหลังของคิไวซ์คอนเท็กซ์

เราสามารถตั้งค่าของสีได้โคการใช้ฟังก์ชัน `SetTextColor` และ `SetBkColor`

ตัวอย่างต่อไปนี้เป็นารตั้งสีของข้อความเป็นสีแดง และสีพื้นหลังเป็นสีเขียว

```
SetTextColor( hdc, RGB( 0, 255, 0 ) );
```

```
SetBkColor( hdc, RGB ( 0, 255, 0 ) );
```

เราสามารถกำหนดสีของพื้นหลังได้ก็ต่อเมื่อโหมดของพื้นหลังเป็น `opaque` (ทึบ) เท่านั้น โหมดของพื้นหลังกำหนดสีของพื้นหลังใน `character cell` จะเป็นโหมด `opaque` สำหรับสิ่งใดบนจอ ถ้าโหมดเป็นแบบโปร่งใส (`Transparent`) สิ่งใด ๆ บนจอจะเป็นอีกอย่างหนึ่ง โดยพื้นหลังจะยังคงอยู่สีของพื้นหลังสำหรับตัวอักษรแบบเอียง ถูกสร้างขึ้นโดย `GDI`

การใช้งานสามารถตั้งและแก้ไขโหมดของพื้นหลังโดยใช้ฟังก์ชัน `SetBkMode` และ `GetBkMode` และเมื่อต้อง การที่จะแก้ไขข้อความและสีของพื้นหลังในปัจจุบันก็จะใช้ฟังก์ชัน `GetTextColor` และ `GetBkColor`

การใช้ตัวอักษรต่างๆในบรรทัดเดียวกัน

สไตล์ต่างๆในกลุ่มของตัวอักษรมีความกว้างต่างกัน เช่น `Bold` และ `Italic` จะใช้ได้กว้างกว่า `Roman` เสมอ เราสามารถแสดงผลต่างๆบนบรรทัดเดียวกันได้ และสามารถแก้ไขความกว้างได้โดยใช้ ฟังก์ชันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ฟังก์ชัน | หน้าที่ |
|---------------------|---|
| GetTabbedTextExtent | คำนวณความกว้างและความสูงของตัวอักษร ถ้ามีหนึ่ง Tab character หรือมากกว่า ความกว้างของตัวอักษรจะเป็น ตัวกำหนดแถวลำดับของตำแหน่ง Tab-stop |
| GetTextExtent | ขนาดความกว้าง และความสูงของบรรทัดของข้อความ |

เมื่อมีความจำเป็นที่ GDI จะต้องสร้างตัวอักษรโดยการเปลี่ยนรูปปิดแม่พของตัวอักษร ในการสร้างตัวอักษร แบบตัวหนา GDI จะแสดงระหว่างจุดเริ่มต้น และพิกเซลที่ติดกันทางขวามือของจุดเริ่มต้น และในการสร้าง Cell อันล่างสุด และเคลื่อนจุดเริ่มต้น ต่อมายังพิกเซลที่อยู่ทางขวาแล้วแสดงสองแถวต่อไป ทำเช่นนี้จนกระทั่งแสดงตัวอักษรเสร็จ เส้นฐาน(Base line)ของตัวอักษรแบบเอียง(Italio)ที่ถูกสร้างจะเลื่อนไปทางขวามือ โดยจำนวนที่กำหนดโดยความสูงของ character cell จำนวนของ base line ที่ถูกเลื่อนไปทางขวานี้จะนำมาใช้ร่วมกับการคำนวณด้วย โดยจะแก้ไขค่าที่ ฟังก์ชัน GetTextMetrics

$$\text{units base line shifted right} = (\text{tmDescent} * \text{tmOverhang}) / \text{tmAscent}$$

มีวิธีเดียวในการเขียนตัวอักษรต่างๆในบรรทัดเดียวกันโดยการใช้ฟังก์ชัน GetTextExtent หลังจากการเรียกฟังก์ชัน TextOut ในแต่ละครั้ง และความยาวที่ตำแหน่งปัจจุบันตามอย่างข้างล่างนี้ "This is a sample string" โดยใช้ ตัวอักษรแบบตัวหนา สำหรับคำว่า "This is a", ตัวอักษรแบบเอียงสำหรับ "Simple" และ System default สำหรับ "String"

```
int XIncrement ;
TEXTMETRIC tm ;
HFONT hfntDefault, hfntItalio, hfntBold;

XIncrement = 10;
hfntDefault = SelectObject(hdc, hfntBold);
TextOut(hdc, XIncrement, 50, "This is a", 10);
```

```

GetTextMetrics(hdc, hfntBold);
XIncrment -= tm.tmOverhang;
SelectObject(hdc, hfntItalic);
GetTextMetrics(hdc, &tm);
XIncrment -= tm.tmOverhang;
TextOut(hdc, XIncrment, 50, "sample ", 7);

XIncrment += LOWORD(GetTextExtent(hdc, "sample ", 7));
SelectObject(hdc, hfntDefault);
TextOut(hdc, XIncrment - tm.tmOverhang, 50, "string.", 7);

```

ในตัวอย่างนี้ฟังก์ชัน `GetTextExtent` ส่งค่าขนาด 32 บิต ทั้งความยาวและความสูงของตัวอักษร `LOWORD` จะแก้ไขความยาวของตัวอักษรซึ่งถูกนำไปบวกกับตำแหน่งปัจจุบันฟังก์ชัน `GetTextMetrics` แก้ไขส่วนที่เกินออกมา ของ ตัวอักษรปัจจุบัน ถ้าเป็นตัวอักษรแบบทรูไทป์ค่านีจะเป็นศูนย์ ส่วนที่เกินออกมาจะไม่มี การเปลี่ยนตำแหน่งของข้อความ ถ้าในกรณีที่เป็น Raster font ค่า `overhang` มีความสัมพันธ์มาก `overhang` จะถูก ลบจากตัวอักษรแบบหนา ที่ล้าคับต่าง ๆ ในตอนท้ายของข้อความถ้าตัวอักษรเป็นแบบ Raster font เพราะ `overhang` มีผลต่อการเริ่มต้น และปลายของตัวอักษรแบบเอียงใน Raster font, glyph จะเริ่มต้นทางขวาของ ตำแหน่งที่กำหนดและ ตอนปลายทางซ้ายมือของจุดสุดท้ายของ Character cell สุดท้าย (ฟังก์ชัน `GetTextExtent` แก้ไขส่วนที่ขยายของ character cell ไม่ใช่ของ glyph) พิจารณา `overhang` ของ Raster Italic string, ในตัวอย่างนี้ ลบ `overhang` ก่อนการวางตำแหน่งของข้อความ และลบมันอีกครั้งก่อนการวาง subsequent character การกำหนดตำแหน่งที่ถูกต้องสามารถใช้ฟังก์ชัน `GetCharWidth` หรือ `GetCharABCWidths` แก้ไขความกว้างของตัวอักษร โดยเฉพาะฟังก์ชัน `GetCharABCWidths` มีความแน่นอน กว่าฟังก์ชัน `GetCharWidth` แต่จะเฉพาะเมื่อใช้กับตัวอักษรแบบทรูไทป์เท่านั้น

ฟังก์ชัน `SetTextJustification` เพิ่มขอบเขตของฟังก์ชันที่ Break character ในบรรทัดของข้อความ เรา สามารถใช้ ฟังก์ชัน `GetTextExtent` กำหนดส่วนขยายของข้อความ โดยลบส่วนที่ขยายจากจำนวนทั้งหมดของ ช่องว่าง ในบรรทัดที่จะแทนที่และใช้ฟังก์ชัน `SetTextJustification` ขอบเขตของช่องว่างระหว่าง Break character ในข้อความ สำหรับฟังก์ชัน `SetTextCharacterExtra` จะเพิ่มขอบเขตของช่องว่างทุกๆ Character cell ในการเลือกตัวอักษรโดยจะรวม Break character เข้าไปด้วย (ฟังก์ชัน `GetTextCharacterExtra` กำหนดจำนวน ปัจจุบันของขอบเขตที่เพิ่มเข้าไปใน Character cell โดยการตั้งค่าเริ่มต้นไว้ที่ศูนย์)

การเว้นช่องว่างของ ABC จะมีการทำงานที่แน่นอนมาก เช่น เมื่อ Raster roman font ทางขวามือ ไม่มีการ ใช้ ABC spacing ความกว้างจะถูกคำนวณโดยรวม white space ทางขวามือด้วย การใช้ ABC width สามารถ ออกมาเป็นเอกสารที่ส่งมอบไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อาจนำไปใช้ประโยชน์ด้านการค้า ได้จึงขอได้มากในการจัดตำแหน่งและเคลื่อนย้าย whitespace เมื่อ มีการจัดตั้งข้อความ ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความแบบหมุนรอบ

เราสามารถหมุนตัวอักษรแบบทรงแปดเหลี่ยมที่จุดใด ๆ ดังตัวอย่างต่อไปนี้ จะเป็นการหมุนข้อความ โดยจะมี องศาเพิ่มขึ้นทุก ๆ 10 องศา รอบจุดศูนย์กลาง โดยการเปลี่ยนค่าของ `lEEscapement` ของในโครงสร้างข้อมูล `LOGFONT`

```
RECT ro;
int angle;
HFONT hfnt, hfntPrev;
LPSTR lpszRotate = "String to be rotated.";
PLOGFONT plf = (PLOGFONT) LocalAlloc(LPTR, sizeof(LOGFONT));

lstrcpy(plf->lFaceName, "Arial");
plf->lWeight = 700;

GetClientRect(hwnd, &ro);
SetBkMode(hdc, TRANSPARENT);

for( angle = 0 ; angle < 3600 ; angle += 100 ) {
    plf->lEEscapement = angle;
    hfnt = CreateFontIndirect(plf);
    hfntPrev = SelectObject(hdc, hfnt);
    TextOut(hdc, ro.right / 2, ro.bottom / 2, lpszRotate, strlen(lpszRotate));
    SelectObject(hdc, hfntPrev);
    DeleteObject(hfnt);
}

SetBkMode(hdc, OPAQUE);
LocalFree( (LOCALHANDLE) plf);
```

`lEEscapement` ของโครงสร้างข้อมูล `LOGFONT` ไม่ถูกสนใจโดย `GDI` ซึ่งการสมมุติค่าของ `lEEscapement` แต่ `lEOrientation` จะเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 โครงสร้างและฟังก์ชันของตัวอักษรแบบทวิไทยปี

การแก้ไขรูปร่างของตัวอักษร

เราใช้ฟังก์ชัน GetGlyphOutline แก้ไขรูปร่างของ glyph จากตัวอักษรแบบทวิไทยปี GetGlyphOutline ส่งค่าของรูปร่างที่เหมือนกันในทางบิตแมพ แต่โดยต่อจาก Polyline และ Spline.

เมื่อมีการแก้ไขรูปร่างของ glyph ค่าต่างๆจะถูกส่งกลับไปในโครงสร้างข้อมูล TTPOLYGONHEADER ด้วย โดย โครงสร้างข้อมูล TTPOLYCURVE ซึ่งเป็นที่ต้องการของ glyph จุดทั้งหมดที่ถูกส่งค่ากลับไปทีโครงสร้างข้อมูล POINTFX และ แสดงตำแหน่งสมบูรมโดยไม่สัมพันธ์กับการเคลื่อนที่ของจุดเริ่มต้นที่กำหนดโดย pfxStart ของโครงสร้างข้อมูล TTPOLYGONHEADER เป็นจุดที่รูปร่างสำหรับจุดเริ่มต้นของเส้นแสดงรูปร่างโครงสร้างข้อมูล TTPOLYCURVE ซึ่งสามารถทำได้ทั้ง Polyline record หรือ Spline record polyline record เป็นเส้นวากระหว่างจุดที่มีตัวอักษรรูปร่างของ Character Spline record แสดงในลักษณะ Quadratic curve นี้ถูกใช้โดยทวิไทยปี(Quadratic b-spline)

แต่ละ Polyline และ Spline record ประกอบด้วยจุดที่วางต่อกันมากมายแต่ค่าค่าสุดของ record จะส่งกลับไปจุดเริ่มต้น ที่กำหนดในโครงสร้างข้อมูล TTPOLYGONHEADER จะเป็นรูปร่างของ glyph เสมอ ข้อกำหนดของจุดคือ จุดเริ่มต้นและจุดสิ้นสุดของเส้นรูปร่าง

Poly line record เริ่มต้นโดยจุดสุดท้ายใน record ก่อน (โดยจุดเริ่มต้นสำหรับ record แรก ในเส้นรูปร่าง) แต่ละจุดใน record อยู่บนรูปร่างของ glyph และสามารถถูกต่อโดยวิธีง่ายๆโดยการใช้ Straight line

Spline record เริ่มต้นโดย คสุดท้ายใน record ก่อน (หรือ โดยจุดเริ่มต้น สำหรับ record แรกในเส้นรูปร่าง) สำหรับ Spline record แรก จุดเริ่มต้นสำหรับ record แรกในเส้นรูปร่าง) สำหรับ Spline record แรก จุดเริ่มต้น และจุดสุดท้าย ใน record อยู่บนรูปร่างของ glyph เท่านั้น จุดอื่นๆทั้งหมด น spline record ที่หุขบน glyph และต้องแสดงจุดควบคุม b-spline ของ Spline หรือ Polyline record ในปลายเส้นรูปร่างเสมอ โดยจุดเริ่มต้นของเส้นรูปร่าง เพราะ b-spline จุด 3 จุด (จุดหนึ่งอยู่ห่างออกไปจาก glyph outline ระหว่าง 2 จุดบน outline) จะต้องมีการคำนวณเมื่อ Spline record มี Off -ourte point มากกว่ากว่า 1 จุด

ตัวอย่าง ถ้า Spline record ประกอบด้วย 3 จุด (A,B และ C) และไม่มี record แรก จุด A และ B เป็น อยู่ห่างจาก glyph outline ที่จุด A ซึ่งสามารถใช้ตำแหน่งปัจจุบัน (ซึ่งอยู่บน glyph outline เสมอ) และ จุดบน glyph outline ระหว่าง จุด A และ B ในการหาจุดระหว่าง A และ B แสดงการคำนวณดังนี้

$$M = A + (B-A) / 2$$

จุดกลางระหว่าง off-outline point ใน spline record เป็นจุดที่อยู่บน glyph outline ความเหมาะสม ของรูปแบบ Spline ที่ใช้ในตัวอักษรแบบทวิไทยปีจากสูตร M เป็น จุดกลางเส้นระหว่าง A และ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญูญาติหนาไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าจุดปัจจุบันถูกออกแบบโดย P โดย 2 Quadratic spline ถูกกำหนดโดย Spline record นี้ คือ (P,A,M) และ (M,B,C)

การแสดง TrueType character outline ใน GDI ต้องใช้ Polyline และ Spline record GDI สามารถแสดง Polyline ได้ง่าย แต่ ไม่ได้สนับสนุนรูปแบบของ Spline การใช้ Spline record คอบแปลง ให้รูปในสุดของ Polyline ใกล้เคียงกับ Spline



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ฟังก์ชันและโครงสร้างข้อมูลของ Windows ที่ใช้ในโปรแกรม

4.1 ฟังก์ชันที่ใช้ในโปรแกรม

ฟังก์ชันที่ใช้กับเมสเสจ

ฟังก์ชันในกลุ่มนี้มีหน้าที่ในการอ่านเมสเสจขึ้นมาจากคิวและประมวลผลเมสเสจใน Windows

| ฟังก์ชัน | หน้าที่ |
|----------------------|--|
| CallWindowProc | ส่งผ่านข้อมูลของเมสเสจ ไปยังฟังก์ชัน |
| DispatchMessage | แจกแจงเมสเสจและส่งเมสเสจ ไปยังฟังก์ชันประจำวินโดว ใดๆ |
| GetMessage | รับเมสเสจในช่วงของเมสเสจที่กำหนดไว้ |
| GetMessagePost | ให้ค่าตำแหน่งของเมาส์ในขณะที่มีการรับเมสเสจล่าสุด |
| GetMessageTime | ให้ค่าเวลาในขณะที่มีการรับเมสเสจล่าสุด |
| PeekMessage | ตรวจสอบว่ามีเมสเสจในคิวหรือไม่โดยไม่ดึงออกมา |
| PostAppMessage | ส่งเมสเสจไปยังแอปพลิเคชัน |
| PostMessage | ส่งเมสเสจไปยังคิวของแอปพลิเคชัน |
| PostQuitMessage | ส่ง WM_QUIT ไปยังแอปพลิเคชัน |
| ReplyMessage | ตอบรับเมสเสจ |
| SendMessage | ส่งเมสเสจ ไปยังวินโดวต่างๆ |
| TranslateAccelerator | ประมวลผลคีย์ทันใจของเมนูให้เป็นเมสเสจและส่งเมสเสจให้ แก่วินโดว |
| TranslateMessage | แปลงคีย์ของคีย์ที่กดให้เป็นรหัสตัวอักษร |
| WaitMessage | ส่งการควบคุมไปให้แอปพลิเคชันอื่น |
| WinMain | เป็นจุดที่เริ่มเข้าไปทำงานของ Windows แอปพลิเคชัน |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันเกี่ยวกับการวาด

หน้าที่ของฟังก์ชันในกลุ่มนี้คือ การเตรียมวินโดว์สำหรับการวาดภาพ และทำงานต่างๆที่เกี่ยวข้องกับด้านกราฟิกต่างๆไป แม้ว่าส่วนมากของฟังก์ชันเหล่านี้จะใช้กับจอแสดงผล แต่ว่าก็ยังมียีกหลายฟังก์ชันที่ใช้ได้กับอุปกรณ์เอาท์พุททั่วไป

| ฟังก์ชัน | หน้าที่ |
|----------------|--|
| BeginPaint | เตรียมวินโดว์สำหรับการวาด |
| DrawFocusRect | วาดกรอบเหลี่ยมที่ใช้สำหรับชี้โฟกัส |
| DrawIcon | วาดไอคอน |
| DrawText | แสดงข้อความ |
| EndPaint | สิ้นสุดการวาด |
| FillRect | ระบายสีในกรอบเหลี่ยมด้วยแปรงที่กำหนด |
| FrameRect | วาดรูปกรอบในกรอบเหลี่ยมที่ต้องการ |
| GetDC | ให้ค่าคอนเท็กซ์ดีสเพลย์ของพื้นที่ใช้งาน |
| GetUpdateRgn | หาค่าขอบเขตบริเวณของวินโดว์ที่ควรวาด |
| GetWindowDC | ขอค่าแฮนเดิลของคอนเท็กซ์ดีสเพลย์ของทั้งวินโดว์ |
| GrayString | เขียนข้อความโดยใช้ตัวอักษรจาง |
| InvalidateRect | กำหนดพื้นที่ที่จะวาดใหม่ |
| InvalidateRgn | ทำให้พื้นที่เป็นโมฆะ ต้องวาดใหม่ |
| InvertRect | กลับสีของภาพในพื้นที่ที่กำหนด |
| ReleaseDC | ปล่อยแฮนเดิลของคอนเท็กซ์ดีสเพลย์ |
| UpdateWindow | บอกแอปพลิเคชันว่าบางส่วนของวินโดว์ต้องการวาดใหม่ |
| ValidateRect | เลิกการกำหนดพื้นที่ที่จะวาดใหม่ |
| ValidateRgn | เลิกการกำหนดขอบเขตที่จะวาดใหม่ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มฟังก์ชันกรอบข้อความ

ฟังก์ชันในกลุ่มนี้มีหน้าที่สร้าง ทำลาย แก้ไข หรือทดสอบ กรอบข้อความและคอนโทรลต่างๆที่ประกอบอยู่ในกรอบข้อความนั้น

| ฟังก์ชัน | หน้าที่ |
|---------------------------|---|
| CheckDlgButton | ใส่หรือยกเลิกกากบาทในคอนโทรลเช็ค หรือเปลี่ยนสถานะของ ปุ่มแบบสามสถานะ |
| CheckRadioButton | เลือกปุ่มวิทยุที่ต้องการ |
| CreateDialog | สร้างกรอบข้อความแบบ modeless |
| CreateDialogIndirect | สร้างกรอบข้อความแบบ modeless จากเทมเพลต |
| CreateDialogIndirectParam | สร้างกรอบข้อความแบบ modeless จากเทมเพลต และใส่ข้อมูลเมื่อสร้างเสร็จ |
| CreateDialogParam | สร้างกรอบข้อความแบบ modeless และใส่ข้อมูลเมื่อสร้างเสร็จ |
| DefDlgProc | กำหนดการประมวลผลตามปกติ ให้กับเมสเสจที่ฟังก์ชันประจำกรอบข้อความไม่ได้ประมวลผล |
| DialogBox | สร้างกรอบข้อความแบบ modal |
| DialogBoxIndirect | สร้างกรอบข้อความแบบ modal จากเทมเพลต |
| DialogBoxIndirectParam | สร้างกรอบข้อความแบบ modal จากเทมเพลต และใส่ข้อมูลเมื่อสร้างเสร็จ |
| DialogBoxParam | สร้างกรอบข้อความแบบ modal และใส่ข้อมูลเมื่อสร้างเสร็จ |
| DlgDirList | เติมข้อมูลในกรอบรายชื่อด้วยชื่อไฟล์ตามไดเรกทอรีที่กำหนด |
| DlgDirListComboBox | เติมข้อมูลในคอมโบบ็อกซ์ด้วยชื่อไฟล์ตามไดเรกทอรีที่กำหนด |
| DlgDirSelect | ก๊อปปี้ตัวอักษรจากกรอบรายชื่อสู่ฟิเฟอร์ |
| DlgDirSelectComboBox | ก๊อปปี้ตัวอักษรจากคอมโบบ็อกซ์สู่ฟิเฟอร์ |
| EndDialog | ทำลายกรอบข้อความแบบ modal และปล่อยรีซอร์สต่างๆ ที่เกี่ยวกับกรอบข้อความนั้น |
| GetDialogBaseUnits | หาค่าหน่วยพื้นฐานของกรอบข้อความ |
| GetDlgCtrlID | ให้ค่าอ้างอิง(ID)ของคอนโทรลที่อยู่ในกรอบข้อความ |
| GetDlgItem | ให้ค่าแฮนเดิลของคอนโทรลที่อยู่ในกรอบข้อความ |
| GetDlgItemInt | เปลี่ยนค่าตัวอักษรในคอนโทรลให้เป็นค่าเลขจำนวนเต็ม |
| GetDlgItemText | ก๊อปปี้ตัวอักษรในคอนโทรลมายังฟิเฟอร์ |

| | |
|---------------------|---|
| GetNextDlgGroupItem | ให้ค่าแอสเคลลของคอนโทรลต่อไปที่อยู่ในกลุ่มเดียวกัน (WS_GROUP) |
| GetNextDlgTabItem | ให้ค่าแอสเคลลของคอนโทรลตัวก่อนหรือถัดไปในกรอบข้อความนั้น |
| IsDialogMessage | ดูว่าเมสเสจนั้นเป็นของกรอบข้อความโดยเฉพาะหรือไม่ |
| IsDlgButtonChecked | ทดสอบว่าปุ่มถูกกาบาทหรือไม่ |
| MapDialogReot | แปลงค่าพิกัดของกรอบข้อความเป็นพิกัดของพื้นที่ใช้งาน |
| SendDlgItemMessage | ส่งเมสเสจไปยังคอนโทรลต่างๆในกรอบข้อความ |
| SetDlgItemText | เปลี่ยนข้อความของคอนโทรล |

ฟังก์ชันเกี่ยวกับคอนเท็กซ์อุปกรณ์

หน้าที่ของฟังก์ชันในกลุ่มนี้คือ การสร้าง การทำลาย และการขอคืนคอนเท็กซ์อุปกรณ์

| ฟังก์ชัน | หน้าที่ |
|--------------------|--|
| CreateCompatibleDC | สร้างคอนเท็กซ์อุปกรณ์ในหน่วยความจำให้เหมือนกับคอนเท็กซ์อุปกรณ์ของอุปกรณ์นั้น |
| CreateDC | สร้างคอนเท็กซ์อุปกรณ์ |
| DeleteDC | ทำลายคอนเท็กซ์อุปกรณ์ |
| GetDCOrg | ขอยุคค่าเน็คของคอนเท็กซ์อุปกรณ์ |
| RestoreDC | นำกลับคอนเท็กซ์อุปกรณ์ |
| SaveDC | เก็บสถานะปัจจุบันของคอนเท็กซ์อุปกรณ์ |

นอกจากนี้ยังมีฟังก์ชันของ USER ที่สามารถรับและปล่อยคอนเท็กซ์อุปกรณ์ได้

| ฟังก์ชัน | หน้าที่ |
|-------------|---|
| BeginPaint | เตรียมวินโดว์สำหรับการวาด และรับค่าแอสเคลลของคอนเท็กซ์อุปกรณ์ |
| GetDC | รับค่าแอสเคลลของคอนเท็กซ์อุปกรณ์จากวินโดว์ที่ให้ |
| GetWindowDC | รับคอนเท็กซ์อุปกรณ์ของทั้งวินโดว์ |
| ReleaseDC | ปล่อยคอนเท็กซ์ |

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันเกี่ยวกับจานสี

งานสีของ Windows อนุญาตให้แอปพลิเคชันสามารถใช้สีมากกว่าที่ต้องการ

| ฟังก์ชัน | หน้าที่ |
|-------------------------|---|
| AnimatePalette | ให้ค่าสีใหม่กับจานสีของวินโดว์ และมีการถือป็นสีนั้น ไปยังจานสีของระบบทันที |
| CreatePalette | สร้างจานสีของวินโดว์ |
| GetNearestPaletteIndex | ดึงค่าดัชนีของช่องในจานสีที่มีสีใกล้เคียงกับค่า RGB ที่ต้องการ |
| GetPaletteEntries | ดึงค่าสีจากจานสี |
| GetSystemPaletteEntries | ดึงค่าช่วงของจานสีจากจานสีของระบบ |
| GetSystemPaletteUse | พิจารณาว่าแอปพลิเคชันมีการเข้าใช้จานสีของระบบเต็มที่หรือไม่ |
| RealizePalette | ถือป็นจานสีของวินโดว์ลง ไปยังจานสีของระบบ |
| SelectPalette | เลือกจานสีที่ต้องการ ให้กับคิสเพลย์คอนเท็กซ์ |
| SetPaletteEntries | ให้ค่าสีใหม่ลง ไปในช่องของจานสีของวินโดว์แต่ยังไม่ถือป็นสีลง ไปยังจานสีของระบบจนกว่าจะใช้ RealizePalette |
| SetSystemPaletteUse | ให้แอปพลิเคชันใช้จานสีของระบบอย่างเต็มที่ |
| UpdateColor | เปลี่ยนค่าสีที่แสดงอยู่ที่ละจุด ช่วยให้วินโดว์ที่ไม่แอกทีฟอยู่สามารถแก้สีของตัวเองได้โดยไม่ต้องวาดพื้นที่ใหม่ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันเกี่ยวกับรูปบิตแมพ

บิตแมพเป็นลักษณะการเรียงตัวของบิตที่เมื่อถูกส่งให้อุปกรณ์แล้ว สามารถแปลเป็นสีของจุดต่างๆบนพื้นที่แสดงผลของอุปกรณ์ได้ประโยชน์ของการสร้างภาพด้วยวิธีนี้คือ สามารถเตรียมภาพที่ต้องการไว้ในหน่วยความจำก่อนได้ จากนั้นจึงค่อยส่งมาแสดงทำให้แสดงผลรูปที่ซับซ้อนได้อย่างรวดเร็ว

ความสัมพันธ์ของบิตต่างๆบนรูปภาพบิตแมพในหน่วยความจำกับจุดต่างๆบนอุปกรณ์แสดงผลนั้นจะมีความอิสระต่อกันไม่ตายตัว ขึ้นอยู่กับชนิดของอุปกรณ์เป็นสำคัญ

| ฟังก์ชัน | หน้าที่ |
|-------------------------|---|
| BitBlt | ก๊อปปี้รูปบิตแมพจากต้นทางไปยังปลายทาง |
| CreateBitmap | สร้างรูปบิตแมพ |
| CreateCompatibleBitmap | สร้างรูปบิตแมพโดยขึ้นกับอุปกรณ์แสดงผลในหน่วยความจำ |
| CreateDiscardableBitmap | สร้างรูปบิตแมพโดยขึ้นกับอุปกรณ์แสดงผลในหน่วยความจำที่ถูกละได้ |
| ExtFloodFill | ระบายในหรือทับรูปบิตด้วยสีที่ระบุ |
| FloodFill | ระบายสีในรูปบิต |
| GetBitmapBits | ก๊อปปี้บิตจากรูปบิตแมพไปยังบัฟเฟอร์ |
| GetBitmapDimension | ขอค่าความกว้างยาวของรูปบิตแมพ |
| GetPixel | ขอค่า RGB จากพิกเซลที่ต้องการ |
| LoadBitmap | โหลดรูปบิตแมพจากไฟล์รีซอร์ส |
| PatBlt | สร้างรูปแบบของบิต |
| SetBitmapBits | เซตค่าบิตในรูปบิตแมพ |
| SetBitmapDimension | เซตค่าความกว้างยาวของรูปบิตแมพ |
| SetPixel | ให้ค่า RGB กับพิกเซลใดๆ |
| StretchBlt | ก๊อปปี้รูปบิตแมพจากต้นทางไปยังปลายทาง โดยอาจมีการหดหรือขยาย |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันเกี่ยวกับฟอนต์

ฟังก์ชันเกี่ยวกับฟอนต์นี้มีหน้าที่ในการสร้าง เลือก ลบ หรือดึงข้อมูลเกี่ยวกับฟอนต์นั้นๆ

| ฟังก์ชัน | หน้าที่ |
|----------------------------|--|
| AddFontResource | เพิ่มฟอนต์ใหม่เข้าไปในตารางฟอนต์ระบบ |
| CreateFont | สร้างฟอนต์ทางตรงใหม่ |
| CreateFontIndirect | สร้างฟอนต์ทางตรงใหม่ |
| CreateScalableFontResource | สร้างไฟล์ของรีซอร์สบรรยายข่าวสารเกี่ยวกับโคเรคตอรีของฟอนต์และชื่อโมดูลของฟอนต์ เพื่อนำไปอ้างอิงที่อยู่ของไฟล์ฟอนต์ |
| EnumFontFamilies | หาจำนวนฟอนต์ในครอบครัวของฟอนต์ที่ต้องการบนอุปกรณ์ที่กำหนด |
| EnumFonts | ดูจำนวนฟอนต์ที่มี |
| GetAspectRatioFilter | รับค่าอัตราส่วนที่ปรับไว้ใช้ในปัจจุบัน |
| GetAspectRAtioFilterEx | รับค่าอัตราส่วนที่ปรับไว้ใช้ในปัจจุบัน |
| GetCharABCWidth | ขอค่าความกว้างของตัวอักษรในช่วงที่กำหนดจากฟอนต์ทรูไทย |
| GetCharWidth | ขอค่าความกว้างของตัวอักษรในช่วงที่กำหนดจากฟอนต์ปัจจุบัน |
| GetFontData | รับข้อมูลเมตริกของไฟล์ฟอนต์ทรูไทย |
| GetGlyphOutline | รับเส้นโค้งหรือบิตแมพที่ประกอบกันเป็นรอบนอกของอักษรหนึ่งตัวในฟอนต์ปัจจุบัน |
| GetOutlineTextMetrics | ขอข้อมูลเมตริกของฟอนต์ทรูไทยที่เลือกไว้ |
| GetRasterizerCaps | รับแฟลกที่จะบอกว่าฟอนต์ทรูไทยได้ติดตั้งในระบบหรือไม่ |
| GetTextFace | ถือปี่ชื่อฟอนต์ไปยังบัฟเฟอร์ |
| GetTextMetrics | ถือปี่เมตริก(ข้อมูล)ของฟอนต์นั้นไปยังบัฟเฟอร์ |
| RemoveFontResource | ลบฟอนต์ออกจากระบบ |
| SetMapperFlags | เปลี่ยนแปลงวิธีการในการแมพฟอนต์ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันเกี่ยวกับการจัดการหน่วยความจำ

ฟังก์ชันที่เกี่ยวกับการจัดการหน่วยความจำจะแบ่งออกเป็น 2 กลุ่มตามชนิดของหน่วยความจำคือ ชนิดที่จัดการหน่วยความจำในพื้นที่ส่วนรวม(global) ซึ่งวงเป็นหน่วยความจำส่วนรวมที่ทุกแอปพลิเคชันสามารถขอใช้งานได้ และที่จัดการหน่วยความจำในพื้นที่ส่วนตัวซึ่งจะเป็นส่วนที่อยู่ในค่าเซกเมนต์ของแต่ละแอปพลิเคชัน

| ฟังก์ชัน | หน้าที่ |
|-------------------|--|
| DefineHandleTable | สร้างตารางเก็บแอสเซ็ลของค่าเซกเมนต์ของแอปพลิเคชัน |
| GetFreeSpace | ให้ค่าหน่วยความจำที่เหลือในพื้นที่ส่วนรวม |
| GetWinFlags | ดึงข้อมูลเกี่ยวกับหน่วยความจำของเครื่อง |
| GlobalAlloc | จองหน่วยความจำในส่วน global |
| GlobalCompact | เลื่อนหน่วยความจำส่วนที่เป็นข้อมูลให้มาติดกัน เพื่อให้เกิดที่ว่างขนาดใหญ่ๆ |
| GlobalDiscard | ลบบล็อกของข้อมูลหน่วยความจำในพื้นที่ส่วนรวม เมื่อตัวแปรที่ใช้นับว่ามีใครใช้อยู่บ้างเป็นศูนย์ แต่ไม่ได้เปลี่ยนแปลงแอสเซ็ลของบล็อกนั้น |
| GlobalDosAlloc | จองหน่วยความจำในพื้นที่ส่วนรวม โดยที่แอปพลิเคชันของคอสที่รันใน real mode หรือ protect mode สามารถเรียกใช้งานได้ |
| GlobalDosFree | ฟรีหน่วยความจำที่ถูกจองโดย GlobalDosAlloc |
| GlobalFlags | ให้ค่าแฟล็กและจำนวนบล็อกของหน่วยความจำในพื้นที่ส่วนรวมที่ถูกล็อก |
| GlobalFree | ลบข้อมูลหน่วยความจำพื้นที่ส่วนรวมในบล็อกที่ต้องการ และยกเลิกแอสเซ็ลของบล็อกนั้น |
| GlobalHandle | ให้ค่าแอสเซ็ลของหน่วยความจำพื้นที่ส่วนรวมของเซกเมนต์แอดเดรสที่ต้องการ |
| GlobalLock | ให้ค่าแอดเดรสที่ชี้ไปยังบล็อกของหน่วยความจำของแอสเซ็ลนั้น และตัวแปรที่เก็บจำนวนครั้งการล็อกก็จะมีค่าเพิ่มขึ้นหนึ่ง |
| GlobalRealloc | จองบล็อกของหน่วยความจำอีกครั้ง |
| GlobalSize | ให้ค่าขนาด(เป็น ไบต์)ของบล็อกของหน่วยความจำ |
| GlobalUnlock | ลบค่าแอดเดรสที่ได้มาจาก GlobalLock ถ้าอยู่ real mode หรือ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ถ้าบล็อกนั้นเป็นแบบโละทิ้งได้(discardable)ตัวแปรที่เก็บ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|-----------------|--|
| | จำนวนการล็อกก็จะลดลงหนึ่ง |
| GlobalUnwire | ลบค่าจำนวนครั้งการล็อกที่ถูกเพิ่มโดย GlobalWire และจะเลิกล็อกบล็อกนั้นถ้าค่านี้เป็นศูนย์ |
| GlobalWire | ย้ายข้อมูลในหน่วยความจำลง low memory และเพิ่มค่าล็อกอีกหนึ่ง |
| LimitEMSPages | จำกัดจำนวนเพจของหน่วยความจำแบบ expand ที่ Windows จะให้แอฟพลิเคชันเข้ามาใช้ |
| LocalAlloc | จองเนื้อที่หน่วยความจำในพื้นที่ส่วนตัว |
| LocalCompact | เลื่อนหน่วยความจำส่วนที่เป็นข้อมูลให้มาติดกันเพื่อให้เกิดที่ว่าง |
| LocalDiscard | ลบบล็อกของข้อมูลในหน่วยความจำส่วน local เมื่อตัวแปรที่ใช้ันับว่ามีใครใช้อยู่บ้างเป็นศูนย์ แต่ไม่ได้เปลี่ยนแปลงแอสเคลของบล็อกนั้น |
| LocalFlags | ให้ค่าชนิดของหน่วยความจำพื้นที่ส่วนตัว |
| LocalFree | ลบข้อมูลหน่วยความจำพื้นที่ส่วนตัว ในบล็อกที่ต้องการ และยกเลิกแอสเคลของบล็อกนั้น |
| LocalHandle | ให้ค่าแอสเคลของหน่วยความจำพื้นที่ส่วนตัวของแอสเคลที่ต้องการ |
| LocalLook | ล็อกหน่วยความจำพื้นที่ส่วนตัว โดยการเพิ่มค่าในตัวแปรที่เก็บจำนวนครั้งการล็อก |
| LocalRealloc | จองหน่วยความจำอีกครั้ง |
| LocalShrink | ลดขนาดหน่วยความจำพื้นที่ส่วนตัว |
| LocalSize | ให้ค่าขนาด(เป็น ไบต์)ของหน่วยความจำพื้นที่ส่วนตัว |
| LocalUnlock | เลิกการล็อกบล็อกหน่วยความจำพื้นที่ส่วนตัว |
| LocalData | ล็อกค่าตัวแปรเมมตปัจจุบัน |
| LocalSegment | ล็อกค่าตัวแปรเมมตใดๆ |
| SetSwapAreaSize | เพิ่มขนาดของหน่วยความจำที่แอฟพลิเคชันสงวนไว้สำหรับไค้คเซกเมมต |
| UnlockData | เลิกล็อกค่าตัวแปรเมมตที่ใช้งานอยู่ปัจจุบัน |
| UnlockSegment | เลิกล็อกค่าตัวแปรเมมตใดๆ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โครงสร้างข้อมูล (DATA STRUCTUR) ที่ใช้ในการเขียนโปรแกรม

BITMAPINFOHEADER

ประกอบด้วยรูปแบบของขนาด และสีของรูปบิตแมพที่ไม่ขึ้นกับอุปกรณ์

| สมาชิก | หน้าที่ |
|---------------------|--|
| DWORD biSize | ระบุขนาด(เป็น ไบต์)ที่ต้องการของโครงสร้างข้อมูล |
| DWORD biWidth | ความกว้างของรูปบิตแมพเป็นพิกเซล |
| DWORD biHeight | ความสูงของรูปบิตแมพเป็นพิกเซล |
| WORD biPlanes | จำนวนของรูป ต้องมีค่าเป็น 1 เสมอ |
| WORD biBitCount | จำนวนของบิตต่อพิกเซล ต้องมีค่าใดค่าหนึ่งดังนี้ |
| ค่า | หมายถึง |
| 1 | โมโนโครม |
| 4 | ไม่เกิน 16 สี |
| 8 | ไม่เกิน 64 พันสี |
| 2 | ไม่เกิน 16 ล้านสี |
| DWORD biCompression | ชนิดของการบีบข้อมูล |
| ค่า | หมายถึง |
| BI_RGB | ไม่มีการบีบข้อมูล |
| BI_RLE8 | บีบข้อมูลแบบ Run-Length Encode ขนาด 8 บิตต่อพิกเซล |
| BI_RLE4 | บีบข้อมูลแบบ Run-Length Encode ขนาด 4 บิตต่อพิกเซล |

DWORD biSizeImage

ขนาด(เป็น ไบต์)ของรูปบิตแมพ

DWORD biXPelsPerMeter

ความละเอียดทางแกนนอนในหน่วยพิกเซลต่อเมตรสำหรับอุปกรณ์

DWORD biYPelsPerMeter

ความละเอียดทางแกนตั้ง ในหน่วยพิกเซลต่อเมตรสำหรับอุปกรณ์

DWORD biClrUsed ที่สงวนไว้สำหรับกา จำนวนของสีที่รูปบิตแมพใช้จริง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DWORD biClrImportant จำนวนของสีที่สำคัญสำหรับรูปปิดแมพ

CHOOSEFONT

เป็นโครงสร้างข้อมูลในการกำหนดค่าเริ่มต้นสำหรับตัวอักษร(font) ใน dialog box ที่ใช้เลือกตัวอักษรนั้น

| สมาชิก | หน้าที่ |
|---|---|
| DWORD lStructSize | ขนาด(เป็น ไบต์)ของโครงสร้างข้อมูล |
| HWND hwndOwner | วินโดวซึ่งกำลังใช้ dialog box |
| HDC hDC | ดีไวซ์คอนเท็กซ์ของตัวอักษร |
| LPLOGFONT lpLogFont | ชี้ไปที่โครงสร้างข้อมูลชื่อ LOGFONT |
| int iPointSize | ขนาดของตัวอักษรในหน่วย 1/10 ของ จุด |
| DWORD Flags | กำหนดรูปแบบการแสดงต่างๆ |
| DWORD rgbColors | สีของตัวอักษร |
| DWORD lCustData | ข้อมูลแอฟพลิเคชันของฟังก์ชัน hook |
| BOOL (FAR PASCAL *lpfnHook)(HWND, unsigned, WORD, LONG) | ชี้ไปที่ฟังก์ชัน hook เมตาสถ |
| LPSTR lpTemplateName | ชี้ไปที่ข้อความของเทมเพลตที่ใช้แทนที่เทมเพลตมาตรฐาน |
| HANDLE hInstance | บล็อกข้อมูลของเทมเพลต |
| LPSTR lpzStyle | ชี้ไปยังบัพเฟอร์ที่เก็บสไตล์ของตัวอักษร |
| WORD nFontType | เลือกชนิดของตัวอักษรซึ่งมี 6 แบบคือ <ul style="list-style-type: none"> - BOLD_FONTTYPE - ITALIC_FONTTYPE - PRINTER_FONTTYPE - REGULAR_FONTTYPE - SCREEN_FONTTYPE - SIMULATED_FONTTYPE |
| int nSizeMin | กำหนดขนาดของตัวอักษรที่เล็กที่สุด |
| int nSizeMax | กำหนดขนาดของตัวอักษรที่ใหญ่ที่สุด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สงวนลิขสิทธิ์ | ห้ามทำซ้ำโดยไม่ได้รับอนุญาต | สงวนลิขสิทธิ์ | สงวนลิขสิทธิ์ | สงวนลิขสิทธิ์ | สงวนลิขสิทธิ์ | สงวนลิขสิทธิ์ | สงวนลิขสิทธิ์ |

LOGFONT

เป็นโครงสร้างข้อมูลที่ใช้ในการสร้างตัวอักษร

| สมาชิก | หน้าที่ |
|----------------------|--|
| int lfHeight | กำหนดความสูงทางตรงของตัวอักษร |
| int lfWidth | กำหนดความกว้างเฉลี่ยของตัวอักษร |
| int lfEsopament | มุมเอียงของตัวอักษรที่ทำกับแกนนอนในการแสดงผล |
| int lfOrientation | มุมระหว่างเส้นฐานของตัวอักษรกับแกนนอน |
| int lfWeight | น้ำหนักของตัวอักษร |
| BYTE lfItalic | การแสดงตัวอักษรแบบเอน |
| BYTE lfUnderline | ขีดเส้นใต้ให้กับตัวอักษร |
| BYTE lfStrikeOut | ขีดเส้นกลางตัวอักษร |
| BYTE lfCharSet | ชุดของตัวอักษรมี 5 แบบคือ <ul style="list-style-type: none"> - ANSI_CHARSET - DEFAULT_CHARSET - OEM_CHARSET - SHIFTJIS_CHARSET - SYMBOL_CHARSET |
| BYTE lfOutPrecision | ความละเอียดในการส่งออกมี 8 แบบ <ul style="list-style-type: none"> - OUT_CHARACTER_PRECIS - OUT_DEFAULT_PRECIS - OUT_DEVICE_PRECIS - OUT_RASTER_PRECIS - OUT_STRING_PRECIS - OUT_STROKE_PRECIS - OUT_TT_PRECIS - OUT_TT_ONLY_PRECIS |
| BYTE lfClipPrecision | ความละเอียดในการตัดออกตัวอักษรที่เกินมี 7 แบบ <ul style="list-style-type: none"> - CLIP_CHARACTER_PRECIS - CLIP_DEFAULT_PRECIS - CLIP_EMBEDDED |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CLIP_LH_ANGLES
- CLIP_MASK
- CLIP_STROKE_PRECIS
- CLIP_TT_ALWAYS

BYTE IfQuality

คุณภาพในการส่งออกเป็นเอาร์ทฟุทมี 3 แบบ

- DEFAULT_QUALITY
- DRAFT_QUALITY
- PROOF_QUALITY

BYTE IfPitchAndFamily

ใน 2 บิตต่ำใช้กำหนดการเว้นระยะมี 3 แบบคือ

- DEFAULT_PITCH
- FIXED_PITCH
- VARIABLE_PITCH

และใน 4 บิตสูงใช้กำหนดครอบครัวของตัวอักษรได้แก่

- FF_DECORATIVE
- FF_DONTCARE
- FF_MODERN
- FF_ROMAN
- FF_SCRIPT
- FF_SWISS

BYTE IfFaceName

ข้อความบอกแบบหรือรูปร่างของตัวอักษร

LOGPALETTE

ใช้สำหรับเป็นข้อมูลของงานสีทางตรง

สมาชิก

หน้าที่

WORD palVersion

เวอร์ชันของ Windows

WORD palNemEntries

จำนวนงานสี

palPalEntry[1]

เป็นข้อมูลแบบอาร์เรย์ของโครงสร้างข้อมูลชื่อ

PALETTEENTRY ใช้

ที่สงวนไว้สำหรับใช้กำหนดสีที่สงวนไว้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MSG

เป็นข้อมูลที่เกี่ยวข้องกับคิวของวินโดว์

| สมาชิก | หน้าที่ |
|--------------|--|
| HWND hwnd | รับเมสเสจของวินโดว์ |
| WORD message | หมายเลขเมสเสจ |
| WORD wParam | ข้อมูลของเมสเสจ |
| LONG lParam | ข้อมูลของเมสเสจ |
| DWORD time | เวลาที่เมสเสจถูกส่ง |
| POINT pt | ตำแหน่งของเคอร์เซอร์ในขณะที่เมสเสจถูกส่ง |

NEWTEXTMETRIC

ประกอบด้วยข้อมูลทางกายภาพของตัวอักษร

| สมาชิก | หน้าที่ |
|-----------------------|---|
| int tmHeight | ความสูงของตัวอักษร |
| int tmAscent | ความสูงของส่วน ascent |
| int tmDescent | ความสูงของส่วน descent |
| int tmInternalLeading | ช่องว่างที่อยู่ส่วนบนสุดของตัวอักษร |
| int tmExternalLeading | ช่องว่างระหว่างแถว |
| int tmAveCharWidth | ความกว้างเฉลี่ยของอักษร |
| int tmMaxCharWidth | ความกว้างของตัวอักษรที่กว้างที่สุด |
| int tmWeight | น้ำหนักของตัวอักษร |
| BYTE tmItalic | ความเอียง |
| BYTE tmUnderline | ขีดเส้นใต้ |
| BYTE tmStrikeOut | ขีดเส้นกลางตัวอักษร |
| BYTE tmFirstChar | กำหนดตัวอักษรตัวแรก |
| BYTE tmLastChar | กำหนดตัวอักษรตัวสุดท้าย |
| BYTE tmDefaultChar | ตัวอักษรที่ใช้แทนแทนในกรณีไม่มีตัวอักษรที่ต้องการ |
| BYTE tmBreakChar | ตัวอักษรที่ใช้สำหรับเป็นตัวหยุดคำ |

สงวนไว้สำหรับใช้ประโยชน์ด้านการค้า

อีกทั้งห้ามมิให้เผยแพร่ไปยังผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|------------------------|---|
| BYTE tmPitchAndFamily | ใช้ 2 บิตสำหรับกำหนดระยะของตัวอักษรมีดังนี้ <ul style="list-style-type: none"> - TMPF_PITCH - TMPF_VECTOR - TMPF_TT - TMPF_DEVICE และ 4 บิตทางด้านสูงสำหรับครอบครวของตัวอักษร มีดังนี้ <ul style="list-style-type: none"> - FF_DECORATIVE - FF_DONTCARE - FF_MODERN - FF_ROMAN - FF_SCRIPT - FF_SWISS |
| BYTE tmCharSet | ชุดของอักขระสำหรับรูปแบบตัวอักษรมี 5 รูปแบบคือ <ul style="list-style-type: none"> - ANSI_CHARSET - DEFAULT_CHARSET - OEM_CHARSET - SHIFTJIS_CHARSET - SYMBOL_CHARSET |
| int tmOverhang | ความกว้างที่เพิ่มขึ้นเป็นพิเศษสำหรับประกอบเป็นตัวอักษร |
| int tmDigitizedAspectX | ลักษณะทางแนวนอนของตัวอักษรสำหรับอุปกรณ์ |
| int tmDigitizedAspectY | ลักษณะทางแนวตั้งของตัวอักษรสำหรับอุปกรณ์ |
| DWORD ntmFlags | ธรรมชาติของแบบของตัวอักษร <ul style="list-style-type: none"> - NTM_REGULAR - NTM_BOLD - NTM_ITALIC |
| WORD ntmSizeEM | ขนาดของ em square |
| WORD ntmCellHeight | ความสูงของตัวอักษรในหน่วย notational |
| WORD ntmAvgWidth | ความกว้างเฉลี่ยของแต่ละอักขระในหน่วย notational |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTLINETEXTMETRIC

เป็นโครงสร้างข้อมูลของทรูไทป์โดยเฉพาะ

| สมาชิก | หน้าที่ |
|---------------------------|--|
| UNIT otmSize | ขนาด(เป็น ไบต์)ของโครงสร้างข้อมูล |
| TEXTMETRIC otmTextMetrics | โครงสร้างข้อมูลชื่อ TEXTMETRIC ที่มีข้อมูลของตัวอักษรเพิ่มเติม |
| BYTE otmFiller | ค่าซึ่งเป็นส่วนประกอบให้จำนวนไบต์ของโครงสร้างข้อมูลถูกต้อง |
| PANOSE otmPanoseNumber | ค่า panose ของตัวอักษร |
| UNIT otmsSelection | รูปแบบเริ่มต้นของตัวอักษรมีอยู่ 6 รูปแบบคือ <ul style="list-style-type: none"> - 0 = Italic - 1 = Underscore - 2 = Negative - 3 = Outline - 4 = Strikeout - 5 = Bold |
| UNIT otmsType | แสดงสถานะของตัวอักษรซึ่งได้รับอนุญาตแล้ว |
| UNIT otmsCharSlopeRise | ความชันของตัวอักษรถ้าเป็นหนึ่งจะตั้งตรง |
| UNIT otmsCharSlopeRun | ความเอียงขณะแสดงถ้าเป็นศูนย์คืออยู่ในแนวตั้ง |
| UNIT otmEMSquare | จำนวนทางตรรกของ em square |
| UNIT otmAscent | ค่า ascent ของตัวอักษรที่สูงที่สุด |
| UNIT otmDescent | ค่า descent |
| UNIT otmLineGap | typographic line gap |
| UNIT otmCapEmHeight | ไม่ได้ใช้ใน Windows |
| UNIT otmXHeight | ไม่ได้ใช้ใน Windows |
| RECT otmrFontBox | ขอบเขตของตัวอักษร |
| UNIT otmMacAscent | ค่า ascent สำหรับเครื่องแมคอินทอช |
| UNIT otmMacDescent | ค่า descent สำหรับเครื่องแมคอินทอช |
| UNIT otmMacLineGap | typographic line gap ของเครื่องแมคอินทอช |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 UNIT otmsMinimumPPEM ขนาดเล็กที่สุดของตัวอักษรในหน่วยพิกเซลคือ cm square
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|------------------------------|---|
| POINT otmptSubscriptSize | ขนาดของคั่นนี้กำกับล่าง(subscript) |
| POINT otmptSubscriptOffset | ระยะ(offset)ของคั่นนี้กำกับล่าง(subscript) |
| POINT otmptSuperscriptSize | ขนาดของคั่นนี้กำกับบน(superscript) |
| POINT otmptSuperscriptOffset | ระยะ(offset)ของคั่นนี้กำกับบน(superscript) |
| UNIT otmsStrikeoutSize | ขนาดของเส้นขีดกลางตัวอักษร |
| UNIT otmsStrikeoutPosition | ตำแหน่งของเส้นขีดกลางตัวอักษร |
| UNIT otmsUnderScoreSize | ขนาดของเส้นขีดใต้ตัวอักษร |
| UNIT otmsUnderScorePosition | ตำแหน่งของเส้นขีดใต้ตัวอักษร |
| PSTR otmpFamilyName | พอยเตอร์ของชื่อของครอบครัว(family)ของตัวอักษร |
| PSTR otmpFaceName | พอยเตอร์ของรูปแบบของตัวอักษร |
| PSTR otmpStyleName | พอยเตอร์ของสไตล์ของตัวอักษร |
| PSTR otmpFullName | พอยเตอร์ของชื่อเต็มของชุดตัวอักษร |

PAINTSTRUCT

ประกอบด้วยข้อมูลของแอฟพลิเคชัน

| สมาชิก | หน้าที่ |
|----------------------|--|
| HDC hDC | คิสเพลย์คอนเท็กซ์ |
| BOOL fErase | ไม่เป็นศูนย์ถ้าแบ็คกราวนด์ไม่ถูกวาดใหม่ และเป็นศูนย์ถ้าแบ็คกราวนด์ถูกวาดใหม่ |
| RECT roPaint | ชื่อโครงสร้างข้อมูลที่มีค่ามุมบนด้านซ้ายและล่างด้านขวาของสี่เหลี่ยมที่ใช้วาด |
| BOOL fRestore | สงวนไว้สำหรับ Windows |
| BOOL fIncUpdate | สงวนไว้สำหรับ Windows |
| BYTE rgbReserved[16] | สงวนไว้สำหรับ Windows |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PALETTEENTRY

ข้อมูลสำหรับการใส่งานสีทางตรง

| สมาชิก | หน้าที่ |
|--------------|-----------------------------|
| BYTE peRed | ความเข้มของสีแดงในงานสี |
| BYTE peGreen | ความเข้มของสีเขียวในงานสี |
| BYTE peBlue | ความเข้มของสีน้ำเงินในงานสี |
| BYTE peFlags | NULL หรือ ใส่เงื่อนไขอื่น |

RECT

เก็บค่ามุมบนด้านซ้ายและมุมล่างด้านขวา

| สมาชิก | หน้าที่ |
|------------|-------------------------------|
| int left | จุดทางแกน X ของมุมบนด้านซ้าย |
| int top | จุดทางแกน Y ของมุมบนด้านซ้าย |
| int right | จุดทางแกน X ของมุมล่างด้านขวา |
| int bottom | จุดทางแกน Y ของมุมล่างด้านขวา |

RGBQUAD

ประกอบด้วยความเข้มของสีแดง, สีเขียว และสีน้ำเงิน ซึ่งถูกใช้ในโครงสร้างข้อมูล BITMAPINFO

| สมาชิก | หน้าที่ |
|------------------|----------------------------|
| BYTE rgbBlue | ความเข้มของสีน้ำเงิน |
| BYTE rgbGreen | ความเข้มของสีเขียว |
| BYTE rgbRed | ความเข้มของสีแดง |
| BYTE rgbReserved | ไม่ได้ใช้ให้กำหนดเป็นศูนย์ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WNDCLASS

เป็นโครงสร้างข้อมูลที่เก็บลักษณะเฉพาะของ Windows คลาส

| สมาชิก | หน้าที่ |
|-------------------------------------|--|
| WORD style | สไตล์ของคลาสมีหลายรูปแบบคือ <ul style="list-style-type: none"> - CS_BYTEALIGNCLIENT - CS_BYTEALIGNWINDOW - CS_CLASSDC - CS_DBLCLKS - CS_GLOBALCLASS - CS_HREDRAW - CS_NOCLOSE - CS_OWNDC - CS_PARENTDC - CS_SAVEBITS - CS_VREDRAW |
| LONG (FAR PASCAL *lpfnWndProc)() | พอยเตอร์ชี้ไปยังส่วนของโปรแกรม Windows |
| int obClsExtra | จำนวนไบต์ที่ถูกจองให้กับคลาส |
| int obWndExtra | จำนวนไบต์ที่จองให้กับ Windows instance |
| HANDLE hInstance | โมดูลของคลาส |
| HICON hIcon | ไอคอนของคลาส |
| HCURSOR hCursor | เคอร์เซอร์ของคลาส |
| HBRUSH hbrBackground | สีของแบ็คกราวนด์โดยระบุเป็นแปรงทาสีหรือค่า ต่างๆดังนี้ <ul style="list-style-type: none"> - COLOR_ACTIVEBORDER - COLOR_ACTIVECAPTION - COLOR_APPWORKSPACE - COLOR_BACKGROUND - COLOR_BTNFACE - COLOR_BTNHIGHLIGHT - COLOR_BTNSHADOW |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- COLOR_BTNTEXT
- COLOR_CAPTIONTEXT
- COLOR_GRAYTEXT
- COLOR_HIGHLIGHT
- COLOR_HIGHLIGHTTEXT
- COLOR_INACTIVEBORDER
- COLOR_INACTIVECAPTION
- COLOR_INACTIVECAPTIONTEXT
- COLOR_MENU
- COLOR_MENUTEXT
- COLOR_SCROLLBAR
- COLOR_WINDOW
- COLOR_WINDOWFRAME
- COLOR_WINDOWTEXT

LPSTR lpszMenuName

ชื่อของเมนูของคลาส

LPSTR lpszClassName

ชื่อของ Windows คลาส

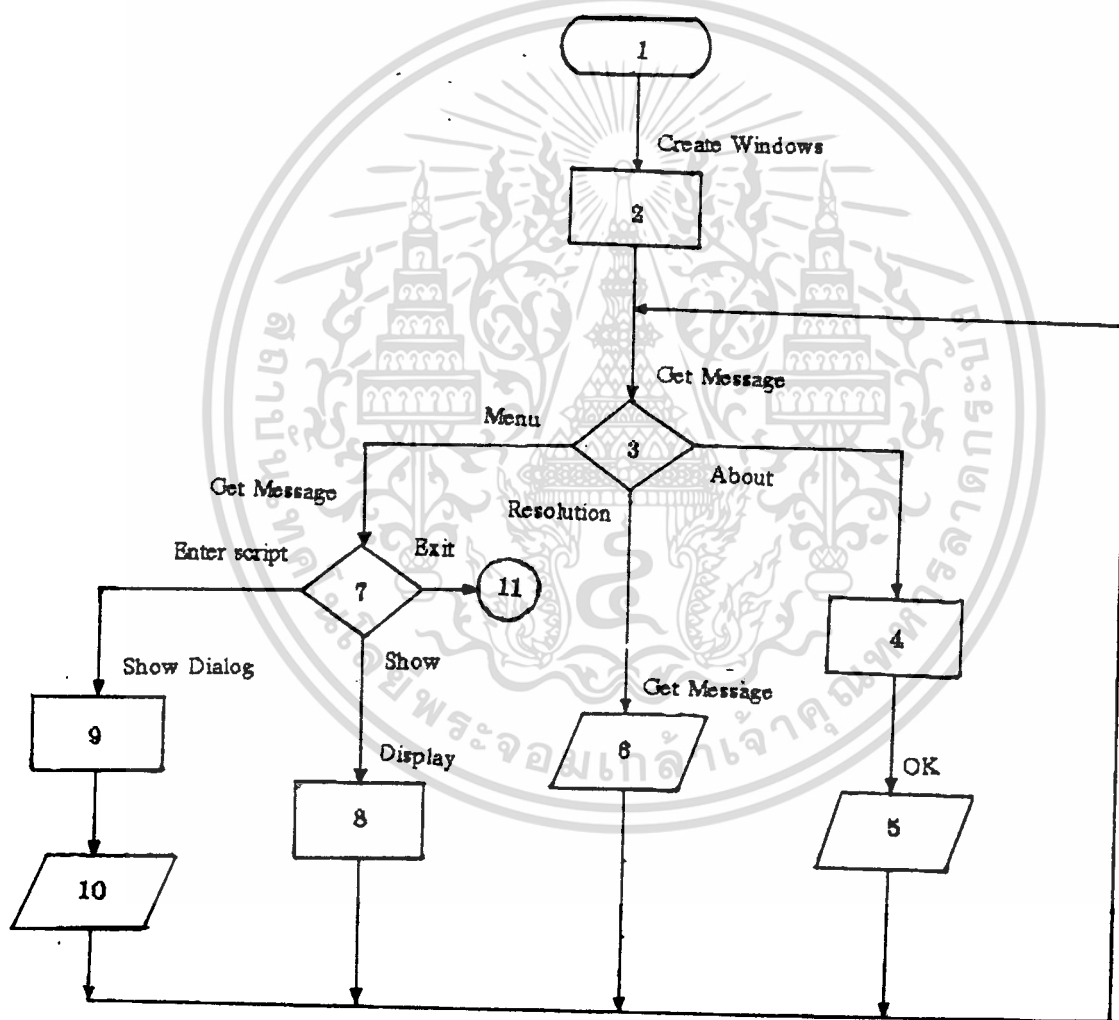
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บล็อกไคอะแกรมและการใช้งานของโปรแกรม

5.1 บล็อกไคอะแกรมของโปรแกรม

ขั้นตอนการทำงานของโปรแกรมนำเสนอภาพและตัวอักษรมีดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปของบล็อก โคอะแกรมอธิบายตามลำดับตัวเลขดังนี้

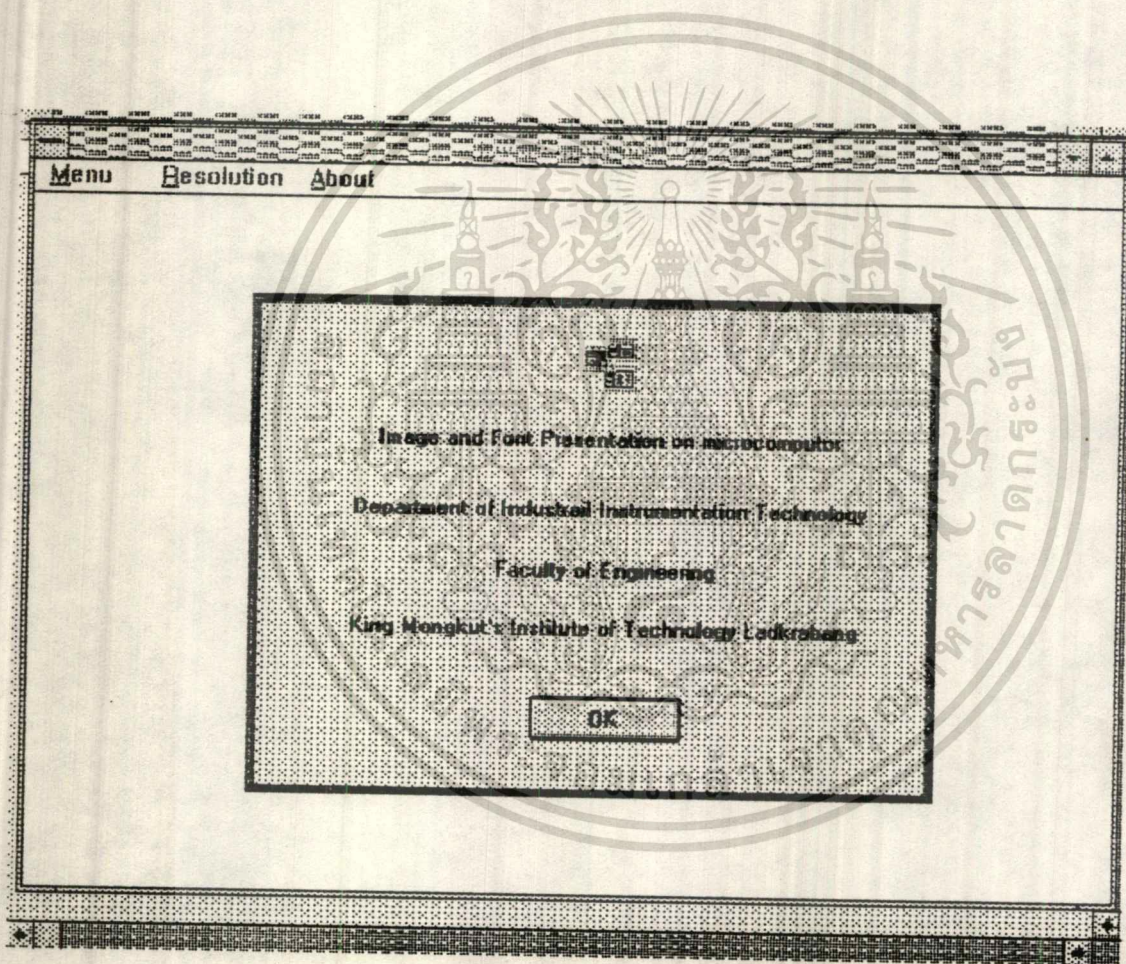
1. เริ่มต้นการทำงาน
2. ทำการสร้างวินโดว์และเมนูหลัก
3. หลังจากสร้างวินโดว์และเมนูหลักแล้ว ก็จะรอรับเมสเสจที่แสดงว่าเมนูใดเมนูหนึ่งได้รับการเลือกใช้งานแล้ว ซึ่งมีเมนูหลัก 3 เมนูคือ
 - About
 - Resolution
 - Menu
4. หากเมนูหลักเลือก About จะทำการแสดง โคอะบล็อกบล็อกสำหรับแสดงข้อความให้ทราบ
5. รอรับเมสเสจที่แสดงว่ามีกรกดคีย์ OK เพื่อลบ โคอะบล็อกบล็อกแล้ว
6. หากเมนูหลักเลือก Resolution จะแสดงเมนูย่อยและรอรับการเลือกเมนูย่อยของเมนู Resolution
7. หากเมนูหลักเลือก Menu จะแสดงเมนูย่อย และทำการรอรับเมสเสจที่แสดงการเลือกเมนูย่อยของ Menu ซึ่งมี 3 เมนูคือ
 - Enter script
 - Show
 - Exit
8. หากเมนูย่อยของ Menu เลือก Show ก็จะทำให้การแสดงผลและตัวอักษรที่ได้กำหนดไว้แล้วจากผู้ใช้งานสิ้นสุดการแสดงผลที่กำหนด
9. หากเมนูย่อยของ Menu เลือก Enter script ก็จะทำให้การสร้าง โคอะบล็อกบล็อกสำหรับรับค่าต่างๆ ในการกำหนดลำดับการแสดงผลและตัวอักษรตามที่ต้องการ
10. รอรับเมสเสจของคีย์ OK ใน โคอะบล็อกบล็อกเพื่อสิ้นสุดการรับค่าของผู้ใช้ และลบ โคอะบล็อกบล็อก
11. หากเมนูย่อยของ Menu เลือก Exit วินโดว์จะถูกทำลายและออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การใช้งานโปรแกรม

โปรแกรมนำเสนอภาพและตัวอักษรจะประกอบด้วยเมนูหลักหลังจากโปรแกรมถูกเรียกขึ้นมาใช้งานแล้วอยู่สามส่วนด้วยกันคือ

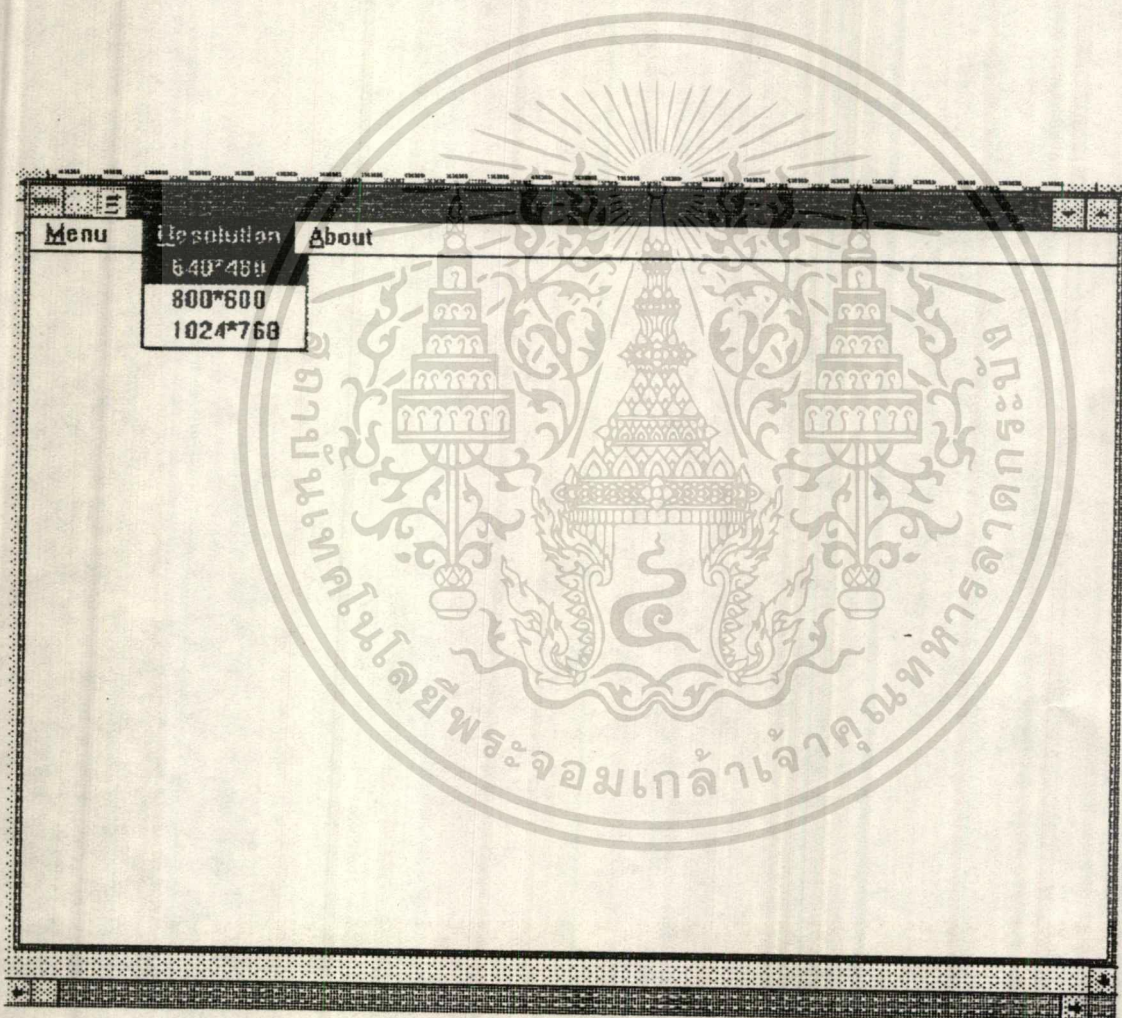
1. About หลังจากเลือกเมนูนี้แล้วจะปรากฏเป็นลักษณะของไดอะล็อกบ็อกซ์ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Resolution เป็นเมนูที่มีไว้สำหรับกำหนดความละเอียดของการแสดงผลของจอภาพ ซึ่งจอภาพในปัจจุบันมีความละเอียดที่แตกต่างกันไป โดยจะมีเมนูย่อยปรากฏหลังจากเลือกเมนู Resolution นี้แล้วให้เลือกสามลักษณะความละเอียดคือ

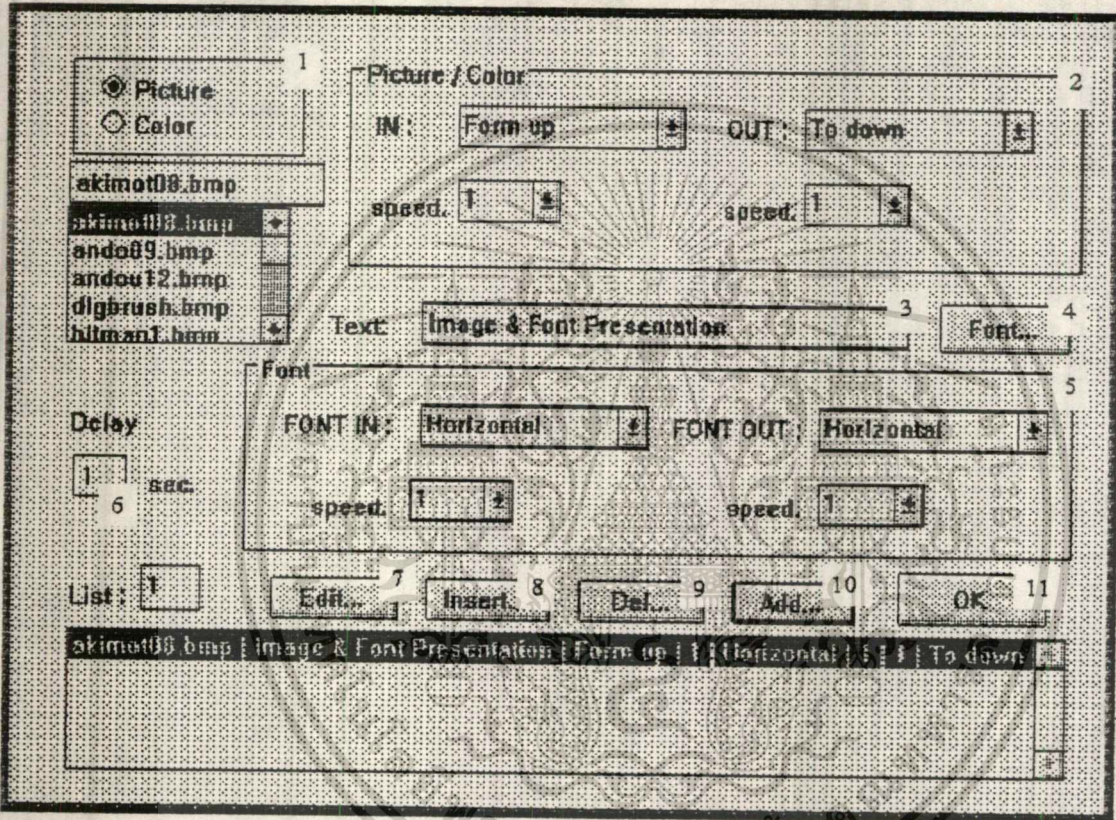
1. 640 * 480
2. 800 * 600
3. 1024 * 726



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Menu เป็นเมนูหลักในการทำงานของโปรแกรมนี้ โดยประกอบด้วยเมนูย่อยดังนี้

1. Enter script เป็นเมนูสำหรับรับข้อมูลจากผู้ใช้โปรแกรมเพื่อกำหนดรูปแบบการแสดงผลภาพและตัวอักษรในรูปแบบต่างๆ หลังจากเลือกเมนูนี้แล้วจะปรากฏไดอะล็อกบ็อกซ์ขึ้น



โดยแต่ละส่วนมีหน้าที่ดังนี้

1. เป็นส่วนของการเลือกภาพที่จะนำมาแสดงโดยสามารถเลือกเป็นภาพ หรือในลักษณะที่เป็นสีพื้นธรรมดาได้
2. เป็นส่วนของการเลือกลักษณะรูปแบบการนำภาพเข้ามาปรากฏบนจอ และการนำภาพออกจากบนหน้าจอภาพโดยที่สามารถกำหนดระดับความเร็วในการแสดงได้
3. ใส่ข้อความที่จะให้ปรากฏในแต่ละครั้ง
4. กำหนดรูปแบบของตัวอักษรที่ปรากฏ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เป็นส่วนของการเลือกลักษณะรูปแบบการนำตัวอักษรเข้ามาปรากฏบนจอ และการนำตัวอักษรออกจากบนหน้าจอภาพโดยที่สามารถกำหนดระดับความเร็วในการแสดงได้

6. หลังจากตัวอักษรปรากฏจนถึงกลางจอภาพจะทำการหน่วงเวลาไว้ตามระยะเวลาที่กำหนดเป็นวินาที

7. เลือกแก้ไขชุดการแสดงชุดใดชุดหนึ่ง

8. เลือกแทรกชุดการแสดงเข้าไประหว่างชุดใดชุดหนึ่ง

9. เลือกลบออกชุดการแสดงชุดใดชุดหนึ่ง

10. เลือกชุดการแสดงชุดใดชุดหนึ่ง

11. สิ้นสุดการเลือกรับข้อมูล และออกจาก โคอะลือบลือก

2. Show เป็นเมนูสำหรับกำหนดให้มีการแสดงภาพและตัวอักษรในรูปแบบต่างๆหลังจากที่ได้เลือกรูปแบบของการแสดงแล้วจากเมนู Enter script หากไม่มีการเลือกรูปแบบของการแสดงใดๆตัวอักษรคำว่า Enter script จะเป็นสีเทาหรือ สีจางๆ ซึ่งไม่สามารถเลือกใช้เมนูนี้ได้

3. Exit ออกจากโปรแกรม



บทที่ 6

สรุปผลและวิจารณ์

โครงการนี้สร้างขึ้นโดยกำหนดวัตถุประสงค์ไว้ดังนี้

1. เป็นการแสดงภาพและตัวอักษรในเวลาเดียวกัน
2. ตัวอักษรที่ใช้สามารถกำหนดและปรับเปลี่ยนได้ตามต้องการ
3. การกำหนดรูปแบบและลำดับการแสดงผลของภาพและตัวอักษร สามารถกำหนดได้ง่ายและสะดวก
4. เป็นโปรแกรมที่ทำงานบน Microsoft Windows

ในโครงการนำเสนอภาพและตัวอักษรบนไมโครคอมพิวเตอร์นี้แบ่งเป็น 2 ส่วน และสรุปผลการทดลอง ดังนี้

1. ในส่วนนี้เป็นการสร้าง Dialog box ขึ้นมาเพื่อใช้สำหรับรับข้อมูลจากผู้ใช้ที่จะกำหนดรูปแบบ และ ลำดับของการแสดงผลภาพและตัวอักษร ซึ่งมีข้อจำกัดอยู่ที่ลำดับของการแสดงกำหนดให้มีลำดับสูงสุดอยู่ที่ 50 ลำดับโดยเกี่ยวเนื่องมาจากการจัดการของหน่วยความจำบน Windows และในการนี้โปรแกรมสามารถ ลบ, แทรก, แก้ไข หรือเพิ่มลำดับการแสดงผลภาพและตัวอักษร ได้อย่างดีตามที่คาดหวังไว้ โดยได้ตรวจสอบจากค่าของ ตัวแปรที่รับข้อมูลเข้ามามีค่าที่ถูกต้องทั้งหมด

2. เป็นส่วนของการแสดงผล โดยแบ่งเป็น 2 ส่วนคือ

2.1 การแสดงผลภาพ แบ่งเป็นการแสดงผลภาพบิตแมพโดยตรง และการแสดงผลภาพบิตแมพที่เป็นสีพื้นซึ่งเกิดจากการสร้างขึ้นเองของโปรแกรม สำหรับใน Windows นั้นการแสดงผลจะเป็นในลักษณะกราฟิกโหมดอยู่แล้ว จึงเป็นการง่ายที่จะนำเสนอภาพ แต่ข้อจำกัดอยู่ที่การแสดงผลแบบกราฟิกโหมดนั้นจะเป็นไปอย่างเชื่องช้า แต่สำหรับเครื่องไมโครคอมพิวเตอร์ที่มีความไวสูงข้อจำกัดนี้จะไม่เป็นปัญหา

2.2 การแสดงตัวอักษรบนภาพ การแสดงตัวอักษรจะต้องเคลื่อนที่อยู่บนภาพ ซึ่งทำได้จากการกำหนด ตำแหน่งตัวอักษรเอาไว้แล้ววางตัวอักษรที่ตำแหน่งนั้น หลังจากนั้นวาดภาพทับไปที่ตัวอักษร แล้วกำหนด ตำแหน่งตัวอักษรใหม่ที่ตำแหน่งต่อไป แล้ววาดภาพทับอีกทำเช่นนี้ไปเรื่อยๆ จากที่ได้กล่าวมาข้างต้นข้อจำกัด ของกราฟิกโหมดในเรื่องความเชื่องช้าจะทำให้การแสดงผลผิดเพี้ยนไปบ้างในบางครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสนอแนะและการปรับปรุง

1. การจัดการหน่วยความจำบน Windows นั้นจะต้องกระทำการโดยผ่านหน่วยความจำที่จัดสรรขึ้นจาก Windows เอง ดังนั้นจึงมีข้อจำกัดอยู่มากในเรื่องการติดต่อกับหน่วยความจำ ซึ่งในการเขียนโปรแกรมต้องระวังอยู่เสมอ

2. การเขียนโปรแกรมที่เกี่ยวกับกราฟิกในลักษณะของโปรแกรมแบบนี้ หากเครื่องไมโครคอมพิวเตอร์ที่ประสิทธิภาพของเครื่องมีการทำงานมีความช้า จะเป็นปัญหาในการแสดงผลอย่างมากในเรื่องความต่อเนื่องของสิ่งที่แสดง ดังนั้นโปรแกรมในลักษณะนี้จึงสนับสนุนสำหรับเครื่องไมโครคอมพิวเตอร์ที่มีการทำงานความไวพอสมควร หรือพัฒนาโปรแกรมให้มีประสิทธิภาพยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// เพิ่มข้อมูล: IFFPRESEN.C
// ประกอบด้วยฟังก์ชัน
//     int PASCAL WinMain(HANDLE, HANDLE, LPSTR, int);
//     void NEAR InitializeStruct(WORD, LPSTR);
//     LPSTR NEAR AllocAndLockMemt(HANDLE *, WORD);
//     long FAR PASCAL MainWndProc(HWND, UINT, WPARAM, LPARAM);
//     BOOL FAR PASCAL About(HWND, UINT, WPARAM, LPARAM);
```

```
#define WIN31
#include "windows.h"
#include "commDlg.h"
#pragma hdrstop
#include "digs.h"
#include "iffpresen.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <conio.h>
#include <memory.h>
#include <time.h>
HANDLE hInst; /* current instance */
char achFileName[32];
char str[64];
char Resoluit;
BOOL bDIBLoaded = FALSE; /* initially no DIB is loaded */
WORD offBits; /* offset to the bits */
HANDLE hDIBInfo = NULL; /* the DIB header */
HBITMAP hDDBitMap = NULL; /* a device dependent copy of the DIB */
HBITMAP hCIBitMap;
HDC hMemDC;
WORD wPalOp = 0; /* default to no palette stuff */
HPALETTE hPalette = NULL; /* palette used for display */
HANDLE hPalHeader = NULL; /* DIB header with indices for color table */
HANDLE hPalInfo;

void PASCAL NEAR PaintDIB(HWND);
DWORD PASCAL lread(int, VOID far *, DWORD);
HPALETTE PASCAL NEAR MakeDIBPalette(LPBITMAPINFOHEADER);
HANDLE PASCAL NEAR MakeIndexHeader(LPBITMAPINFOHEADER);
void ShowRect(HWND, int, int);
void InitDig(HWND);
void AddSpeed(HWND);
void EditDig(HWND, int);
void SetPic(HWND);
void SetCol(HWND);
void Sleep(clock_t wait);
void Delay(int);
void ChangeIn(int);
void ChangeOut(int);
int InitDIB(HWND);
```

```
PAINTSTRUCT ps;
struct {
    char shFile[14];
    char shPsh, shTsh, shPen, shTen, shLst[189];
    char shText[50];
    int shHeight;
    int shWeight;
    BYTE shItalic;
    BYTE shUnderline;
    BYTE shStrikeOut;
    char shFaceName[20];
    DWORD shColor;
    int shLong;
    int shMatly;
    char shImpsh;
    char shTmtsh;
    char shImpert;
    char shTmtent;
    int shPoc;
```

เอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }Getshw(50);
struct {
    int step;
}stepr;
char szFile(20);
char szPsh(11);
char szTsh(11);
char szPen(11);
char szTen(11);
char szTmsh(2);
char szTmpsh(2);
char szTmsh(2);
char szTmsh(2);
char szTmsh(2);
char szTmsh(2);

// ฟังก์ชัน: WinMain(HANDLE, HANDLE, LPSTR, int)
// เป็นฟังก์ชันหลักของโปรแกรม

int PASCAL WinMain( HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{
    MSG msg;
    WNDCLASS wc;
    if ( !hPrevInstance){
        wc.style          = NULL;
        wc.lpfnWndProc    = MainWndProc;
        wc.cbClsExtra     = 0;
        wc.cbWndExtra     = 0;
        wc.hInstance     = hInstance;
        wc.hIcon          = LoadIcon(hInstance, MAKEINTRESOURCE(2));
        wc.hCursor        = LoadCursor(NULL, IDC_ARROW);
        wc.hbrBackground = GetStockObject(WHITE_BRUSH);//COLOR_WINDOW+1;
        wc.lpszMenuName   = "CommonDlgMenu";
        wc.lpszClassName = "CommonWClass";
        if ( ! RegisterClass(&wc)) return FALSE;
    } /* end if */

    if (!InitInstance(hInstance, nCmdShow))
        return(FALSE);

    while (GetMessage(&msg, NULL, NULL, NULL))
    { if ( ghFRDlg == NULL || !IsDialogMessage(ghFRDlg, &msg) )
        { TranslateMessage(&msg);
          DispatchMessage(&msg);
        }
    }

    return (msg.wParam);
} /* end WinMain */

```

```

// ฟังก์ชัน: InitializeStruct(WORD, LPSTR)
// ใช้เป็นส่วนหนึ่งสำหรับการเลือกตัวอักษรสำหรับแสดง

```

```

void NEAR InitializeStruct(LPSTR lpStruct)
{
    LPCHOOSEFONT lpFontChunk;
    lpFontChunk = (LPCHOOSEFONT)lpStruct;
    lpFontChunk->StructSize = sizeof(CHOOSEFONT);
    lpFontChunk->hWndOwner = ghWnd;
    //The hDC field will be initialized when we return-this avoids passing the hDC in or getting the DC twice.
    //The LOGFONT field will also be initialized when we get back
    // lpFontChunk->hDC = hDC;
    // lpFontChunk->lLogFont = &l;
    lpFontChunk->Flags = CF_SCREENFONTS | CF_EFFECTS |
        CF_INITTOLOGFONTSTRUCT | CF_APPLY;
    lpFontChunk->rgbColors = RGB(0, 0, 255);
    lpFontChunk->lCustData = 0L;
    lpFontChunk->lpfnHook = (FARPROC)NULL;
    lpFontChunk->lpTemplateName = (LPSTR)NULL;
    lpFontChunk->hInstance = (HANDLE)NULL;
    lpFontChunk->lpszStyle = (LPSTR)NULL;
    lpFontChunk->nFontType = SCREEN_FONTTYPE;
    lpFontChunk->nSizeMin = 0;
}

```

สงวนลิขสิทธิ์ © 2558 โดยศูนย์พัฒนาระบบสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 สงวนลิขสิทธิ์ ห้ามสงวนลิขสิทธิ์ในส่วนนี้ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
 การบริการและที่ปรึกษาอื่นที่มิได้ติดต่อขอเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    lpFontChunk->nSizeMax    = 0;
return;
}

// ฟังก์ชัน: AllocAndLockMem(HANDLE *, WORD)
// ใช้สำหรับจองและล็อกหน่วยความจำ

LPSTR NEAR AllocAndLockMem(HANDLE *hChunk, WORD wSize)
{
    LPSTR lpChunk;
    *hChunk = GlobalAlloc(GMEM_FIXED, wSize);
    if (*hChunk)
    {
        lpChunk = GlobalLock(*hChunk);
        if (!lpChunk)
        {
            GlobalFree(*hChunk);
            ReportError(IDC_LOCKFAIL);
            lpChunk=NULL;
        }
    }
    else
    {
        ReportError(IDC_ALLOCFAIL);
        lpChunk=NULL;
    }
return(lpChunk);
}

// ฟังก์ชัน: MainWndProc(HWND, UINT, WPARAM, LPARAM)

long FAR PASCAL __export MainWndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    FARPROC    lpProcDialogGet;
    DLGPROC    lpProcAbout1;
    RECT       *rect, rc;
    #define DESIREDPOINTSIZ 12
    static HDC  hDC;
    MSG        msg;
    LPBITMAPINFOHEADER lpInfo, lpHeader;
    HPALETTE   hOldPal;
    HBRUSH     hBrush, hOldBrush;
    LOGFONT    lf;
    LPCHOOSEFONT lpFontChunk;
    HFONT      hOldFont, hFont;
    HMENU      hMenu;
    int        a, j, a, int1, int2, int3, strLong, strp1, spd, nr, Height, Width, Rehg1, Rewth;
    DWORD      dwExtent;
    WORD       wTextWidth, wTextHeight;
    WORD       wDIBUse;
    char       strFace[20], strText[50], Buf;
    static DWORD dwFontColor;

    hMenu = GetMenu(hWnd);
    switch (message)
    {
        case WM_INITMENU:
            if (strEpr.step <= 0)
                EnableMenuItem(hMenu, IDP_SHOW, MF_GRAYED);
            else
                EnableMenuItem(hMenu, IDP_SHOW, MF_ENABLED);
            break;
        case WM_CREATE:
            wFRMsg = RegisterWindowMessage((LPSTR)FINDMSGSTRING);
            wHelpMsg = RegisterWindowMessage((LPSTR)HELPMSGSTRING);
            hDIBInfo = GlobalAlloc(GMEM_MOVEABLE,
                (DWORD)(sizeof(BITMAPINFOHEADER) + 256 * sizeof(RGBQUAD)));
            //Are we in monochrome land????
            hDC = GetDC(NULL);
            gbMonochrome = (2 == GetDeviceCaps(hDC, NUMCOLORS)); // Monochrome!
            ReleaseDC(NULL, hDC);
            //Load brush for painting GetOpenFileName dlg box
            if (!gbMonochrome)

```

เอ็ดเวิร์ด ไบรอน อี. พริทเชอร์ จอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HBITMAP hTempBltmap;
hTempBltmap=LoadBltmap(ghInst, MAKEINTRESOURCE(1));
if (hTempBltmap)
{
    ghDibBrush = CreatePatternBrush(hTempBltmap);
    DeleteObject (hTempBltmap);
}
}
break;
case WM_ERASEBKGD:
if (ghBkgndBrush)
{
    GetClientRect(ghWnd, &rc);
    FillRect((HDC)wParam, &rc, ghBkgndBrush);
    break;
}
else
return(DefWindowProc(hWnd, message, wParam, lParam));
break;
case WM_COMMAND:
switch (wParam)
{

```

```

    case IDP_RES1:
        CheckMenuItem(hMenu, IDP_RES1, MF_CHECKED);
        CheckMenuItem(hMenu, IDP_RES2, MF_UNCHECKED);
        CheckMenuItem(hMenu, IDP_RES3, MF_UNCHECKED);
        Resolun = 1;
        break;
    case IDP_RES2:
        CheckMenuItem(hMenu, IDP_RES1, MF_UNCHECKED);
        CheckMenuItem(hMenu, IDP_RES2, MF_CHECKED);
        CheckMenuItem(hMenu, IDP_RES3, MF_UNCHECKED);
        Resolun = 2;
        break;
    case IDP_RES3:
        CheckMenuItem(hMenu, IDP_RES1, MF_UNCHECKED);
        CheckMenuItem(hMenu, IDP_RES2, MF_UNCHECKED);
        CheckMenuItem(hMenu, IDP_RES3, MF_CHECKED);
        Resolun = 3;
        break;
    case IDP_SHOW:
        if(Resolun == 1){Rewth = 640;Rehgt = 480;}
        else if(Resolun == 2){Rewth = 800;Rehgt = 600;}
        else if(Resolun == 3){Rewth = 1024;Rehgt = 768;}
        else {Rewth = 640;Rehgt = 480;}
        ShowRect(hWnd, Rewth, Rehgt);
        hDC = GetDC(hWnd);
        a=1;

```

bgshw:

```

PeekMessage(&msg, NULL, 0, 0, PM_REMOVE);
TranslateMessage(&msg);
if(msg.wParam == VK_ESCAPE)
goto litrup;
d = a-1;
strcpy(achFileName, Getshw[d].shFile);
strcpy(srText, Getshw[d].shText);
j = d; d = j;
srilong = Getshw[d].shilong;
srpl = Getshw[d].shlmdy;
if(.ifHeight = Getshw[d].shHeight ;
if(.ifWeight = Getshw[d].shWeight ;
if(.ifItalic = Getshw[d].shitalic ;
if(.ifUnderline = Getshw[d].shUnderline ;
if(.ifStrikeOut = Getshw[d].shStrikeOut ;
strcpy(srFace, Getshw[d].shFaceName);
dwFontColor = Getshw[d].shColor;
if(Getshw[d].shPoc == 0)
{

```

```

hMemDC = CreateCompatibleDC(hDC);
hDBltmap = CreateCompatibleBltmap(hDC, Rewth, Rehgt);
hOABltmap = SelectObject(hMemDC, hDBltmap);
if(achFileName[2] == 'd') hBrush = CreateSolidBrush(RGB(255,0,0));
if(achFileName[2] == 'e') hBrush = CreateSolidBrush(RGB(0,255,0));

```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุดแปลกงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(achFileName[2] == 'u') hBrush = CreateSolidBrush(RGB(0,0,255));
if(achFileName[2] == 'f') hBrush = CreateSolidBrush(RGB(255,255,255));
if(achFileName[2] == 'a') hBrush = CreateSolidBrush(RGB(190,190,190));
if(achFileName[2] == 'l') hBrush = CreateSolidBrush(RGB(255,255,0));
if(achFileName[2] == 'o') hBrush = CreateSolidBrush(RGB(255,0,255));
PatBlt(hMemDC, 0, 0, Rewth, Rehgt, BLACKNESS);
hOldBrush = SelectObject(hMemDC, hBrush);
PatBlt(hMemDC, 0, 0, Rewth, Rehgt, 0x00FA00B9);
Height = Rehgt;Width = Rewth;
}
if(Getshw[cj.shPoc == 1)
{
InitDIB(hWnd);
lpInfo = (LPBITMAPINFOHEADER) GlobalLock(hDIBInfo);
hPalette = MakeDIBPalette(lpInfo);
hOldPal = SelectPalette(hDC, hPalette, FALSE);
RealizePalette(hDC);
if (wPalOp == ((lpInfo->biBitCount != 24))
{
hPalHeader = MakeIndexHeader(lpInfo);
lpHeader = (LPBITMAPINFOHEADER) GlobalLock(hPalHeader);
wDIBUse = DIB_PAL_COLORS;
}
hMemDC = CreateCompatibleDC(hDC);
hDDBitmap = CreateCompatibleBitmap(hDC, (WORD)lpInfo->biWidth, (WORD)lpInfo->biHeight);
SetDIBits(hDC, hDDBitmap, 0, (WORD)lpInfo->biHeight, (LPSTR)lpInfo + offBits,
(LPBITMAPINFO)lpHeader, wDIBUse);
hOldBitmap = SelectObject(hMemDC, hDDBitmap);
Width = (WORD)lpInfo->biWidth;Height = (WORD)lpInfo->biHeight; }
SetRect(rect, 0, 0, Rewth, Rehgt);
OffsetRect(rect,0,0);
FillRect(hDC, rect, GetStockObject(BLACK_BRUSH));
Buf = Getshw[cj.shPsh;
szTmpsh[0] = Getshw[cj.shTmpsh;
spd = atoi(szTmpsh);

if(Buf == '1'){
for(int1 = 0;int1 <= Rehgt;int1 += spd)
BitBlt(hDC, Rewth/2-(Width)/2, (Rehgt/2-(Height)/2)+int1, Width, 5, hMemDC, 0, int1, SRCCOPY);
}
if(Buf == '2'){
for(int1 = 0;int1 <= Rewth;int1 += spd)
BitBlt(hDC, (Rewth/2-(Width)/2)+int1, Rehgt/2-(Height)/2, 5, Height, hMemDC, int1, 0, SRCCOPY);
}

if(Buf == '3'){
for(int1 = 0;int1 <= Rehgt/2+(Height)/2;int1 += spd) {
int2 = 0;
if(int1 > Height)
int2 = int1 - Height;
BitBlt(hDC, Rewth/2-(Width)/2, int2, Width, int1 - int2, hMemDC, 0, Height - int1 + int2, SRCCOPY);
if(int1 > Height)
BitBlt(hDC, Rewth/2-(Width)/2, 0,Width, int2, hMemDC, 0, Height, BLACKNESS);
}
}

if(Buf == '4'){
for(int1 = 0;int1 <= Rewth/2+(Width)/2;int1 += spd) {
int2 = 0;
if(int1 > Width)
int2 = int1 - Width;
BitBlt(hDC, int2, Rehgt/2-(Height)/2, int1 - int2, Height, hMemDC, Width - int1 + int2, 0, SRCCOPY);
if(int1 > Width)
BitBlt(hDC, 0, Rehgt/2-(Height)/2, int1 - int2, Height, hMemDC, Width, 0, BLACKNESS);
}
}

if(Buf == '5'){
for(int1 = 0,int2 = 0;int1 <= 320 ;int1 += spd,int2 += spd)
BitBlt(hDC, Rewth/2-int1,240-int2, int1*2, int2*2,hMemDC, ((Width)/2)-int1, ((Height)/2)-int2, SRCCOPY);
}

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

int1 = Rehg/2-wTextHeight/2;
for(int2 = int3;int2 >= (nr*(-1));int2-=spd){
    TextOut(hdc, int2, int1, sText, srlong);
    BitBlt(hdc, int2-3, int1-3, wTextWidth+6, wTextHeight+6,
            hMemDC, (Width)/2+int2-Rewth2-3, (Height)/2-(wTextHeight/2)-3, SRCCOPY);
    BitBlt(hdc, Rewth2+(Width)/2, int1-6, Rewth2-(Width)/2,wTextHeight+12,
            hMemDC, 0, 0, BLACKNESS);
    BitBlt(hdc, 0, int1-6, Rewth2-(Width)/2, wTextHeight+12, hMemDC, 0, 0, BLACKNESS);
}
}

```

```

if(Buf == 2){
    nr = wTextHeight;
    int3 = Rehg/2-nr/2;
    int1 = Rewth2-wTextWidth/2;
    for(int2 = int3;int2 > (nr*(-1));int2-=spd){
        TextOut(hdc, int1, int2, sText, srlong);
        BitBlt(hdc, int1-3, int2-3,wTextWidth+6, wTextHeight+6,
                hMemDC, (Width)/2-wTextWidth/2-3, (Height)/2+int2-Rehg/2-3, SRCCOPY);
        BitBlt(hdc, Rewth2-wTextWidth/2-3, 0, wTextWidth+6,Rehg/2-(Height)/2,
                hMemDC, 0, 0, BLACKNESS);
        BitBlt(hdc, Rewth2-wTextWidth/2-3, Rehg/2+(Height)/2, wTextWidth+6,Rehg/2-(Height)/2,
                hMemDC, 0, 0, BLACKNESS);
        BitBlt(hdc, 0, int2-3, wTextHeight+6, hMemDC, 0, 0, BLACKNESS);
        BitBlt(hdc, Rewth2+(Width)/2, int2-3, Rewth2-(Width)/2,wTextHeight+6,
                hMemDC, 0, 0, BLACKNESS);
    }
}

```

```

    PeekMessage(&msg, NULL,0,0,PM_REMOVE);
    TranslateMessage(&msg);
    if(msg.wParam == VK_ESCAPE)
        goto Itmup;
Buf = Getshw[cl].shPen;
szTmpen[0] = Getshw[cl].shPen;
spd = clol(szTmpen);

```

```

if(Buf == 1){
    for(int1 = 0;int1 <= Rehg;int1 +=spd)
        BitBlt(hdc, Rewth2-(Width)/2, (Rehg/2-(Height)/2)+int1, Width, 5, hMemDC, 0, int1, BLACKNESS);
}
if(Buf == 2){
    for(int1 = 0;int1 <= Rewth;int1 +=spd)
        BitBlt(hdc, (Rewth2-(Width)/2)+int1, Rehg/2-(Height)/2, 5, Height, hMemDC, int1, 0, BLACKNESS);
}

```

```

if(Buf == 3){
    for(int1 = Rehg/2-(Height)/2;int1 <= Rehg;int1 +=spd){
        BitBlt(hdc, Rewth2-(Width)/2, int1, Width, Height, hMemDC, 0, 0, SRCCOPY);
        BitBlt(hdc, Rewth2-(Width)/2, 0,Width, int1, hMemDC, 0, 0, BLACKNESS);
    }
}

```

```

if(Buf == 4){
    for(int1 = Rewth2-(Width)/2;int1 <= Rewth;int1 +=spd){
        BitBlt(hdc, int1, Rehg/2-(Height)/2, Width, Height, hMemDC, 0, 0, SRCCOPY);
        BitBlt(hdc, 0, Rehg/2-(Height)/2, int1, Height, hMemDC, 0, 0, BLACKNESS);
    }
}

```

```

if(Buf == 5){
    for(int1 = 0,int2 = 0;int1 <= Rewth2 ;int1 +=spd,int2 +=spd){
        BitBlt(hdc, 0,0, Rewth, int2, hMemDC, 0,0, BLACKNESS);
        BitBlt(hdc, Rewth-int1, int2, int1, Rehg-int2, hMemDC, 0, 0, BLACKNESS);
        BitBlt(hdc, 0,Rehg-int2, Rewth, int2, hMemDC, 0, 0, BLACKNESS);
        BitBlt(hdc, 0, int2, int1,Rehg-int2*2, hMemDC, 0, 0, BLACKNESS);
    }
}

```

```

if(Buf == 6){
    for(int1 = (Width)/2;int1 >= 0;int1 -=spd){
        BitBlt(hdc, Rewth2-int1, Rehg/2-(Height)/2, int1, Height, hMemDC, 0, 0, SRCCOPY);
        BitBlt(hdc, Rewth2, Rehg/2-(Height)/2, int1, Height,
                hMemDC, Width - int1, 0, SRCCOPY);
        BitBlt(hdc, Rewth2-(Width)/2, Rehg/2-(Height)/2, (Width)/2-int1, Height,
                hMemDC, 0, 0, BLACKNESS);
    }
}

```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        BitBlt(hDC, Rewth/2+int1, Rehg/2-(Height)/2,(Width)/2-int1 , Height,
            hMemDC, 0, 0, BLACKNESS);
    }
}

if(Bur == 7){
for(int1 = (Height)/2;int1 >= 0;int1--spd) {
    BitBlt(hDC, Rewth/2-(Width)/2, Rehg/2-int1, Width, int1, hMemDC, 0, 0, SRCCOPY);
    BitBlt(hDC, Rewth/2-(Width)/2, Rehg/2, Width, int1, hMemDC, 0, Height - int1 , SRCCOPY);
    BitBlt(hDC, Rewth/2-(Width)/2, Rehg/2-(Height)/2, Width, (Height)/2-int1,
        hMemDC, 0, 0, BLACKNESS);
    BitBlt(hDC, Rewth/2-(Width)/2, Rehg/2+int1, Width, (Height)/2-int1,
        hMemDC, 0, 0,BLACKNESS);
    }
}

SelectObject(hDC, hOldFont);
SelectObject(hMemDC, hOldBitmap);
DeleteObject(hFont);
if(Getshw[d].shPoc == 1)
{
SelectPalette(hDC, hOldPal, FALSE);
GlobalFree(hPalInfo);
GlobalUnlock(hDIBInfo);
GlobalUnlock(hPalHeader);
}
if(Getshw[d].shPoc == 0)
{
SelectObject(hMemDC, hOldBrush);
DeleteObject(hBrush);
}
DeleteDC(hMemDC);
a++;
if(a <= Stepr.step)
goto bgnshw;
ReleaseDC(hWnd,hDC);
SetRect(rect, 0, 0, Rewth, Rehg);
FillRect(hDC, rect, GetStockObject(WHITE_BRUSH));
ShowWindow(hWnd,SW_SHOWMAXIMIZED);
MoveWindow(hWnd, 0, 0, Rewth, Rehg, TRUE);
break;
}

//trup: SelectObject(hDC, hOldFont);
SelectObject(hMemDC, hOldBitmap);
DeleteObject(hFont);
if(Getshw[d].shPoc == 1)
{
SelectPalette(hDC, hOldPal, FALSE);
GlobalFree(hPalInfo);
GlobalUnlock(hDIBInfo);
GlobalUnlock(hPalHeader);
}
if(Getshw[d].shPoc == 0)
{
SelectObject(hMemDC, hOldBrush);
DeleteObject(hBrush);
}
DeleteDC(hMemDC);
ReleaseDC(hWnd,hDC);
SetRect(rect, 0, 0, Rewth, Rehg);
FillRect(hDC, rect, GetStockObject(WHITE_BRUSH));
ShowWindow(hWnd,SW_SHOWMAXIMIZED);
MoveWindow(hWnd, 0, 0, 640, 480, TRUE);
break;
}

case IDM_DLGET:
lpProcDialogGet = MakeProcInstance(DialogGet, ghInst);
DialogBox(ghInst,DIALOGGET,hWnd, lpProcDialogGet);
FreeProcInstance(lpProcDialogGet);
break;

case IDM_ABOUT:
lpProcAbout1 = (DLGPROC)MakeProcInstance((FARPROC)About1, ghInst);
if(lpProcAbout1){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าในกรณีใดๆ หากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูงและขอสงวนสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

```

nr = DialogBox(ghInst,ABOUTDLG,hWnd, lpProcAbout1);
FreeProcInstance((FARPROC)lpProcAbout1);
}
break;

case IDM_EXIT:
PostMessage(hWnd, WM_CLOSE, 0, 0);
break;
default:
return(DefWindowProc(hWnd,message,wParam,lParam));
}
break;

case WM_DESTROY:
if (ghDlgBrush) DeleteObject(ghDlgBrush);
if (ghBgkndBrush) DeleteObject(ghBgkndBrush);
if (ghSelectedFont) DeleteObject(ghSelectedFont);
PostQuitMessage(0);
break;
default:
//Let's keep the logic readable and put it in the switch statement.
//Assign some user defined constants here so a switch can be used.
return(DefWindowProc(hWnd,message,wParam,lParam));
break;
}
return (NULL);
} /* end MainWndProc

// ฟังก์ชัน: DialogGet(HWND, UNINT, WPARAM, LPARAM)
// เป็นฟังก์ชันที่ไว้รับค่าที่ผู้ใช้กำหนดเพื่อแสดงไอคอนในรูปแบบแบบไอคอนอีกนอก

BOOL FAR PASCAL __export DialogGet(HWND hDlg, UNINT message, WPARAM wParam, LPARAM lParam)
{
HDC hDC;
LOGFONT lf;
LPCHOOSEFONT lpFontChunck;
HANDLE hFontChunck;
WORD wSize;
WORD wGetset;
DWORD dwGetset;
DWORD dwindex;
WPARAM windex;
static DWORD dwFontColor;
char szText[50], szList[189], intr[4], szTmaly[3];
int a, b, lLogPixsY, szTmaly;
static int z, Height; sWeight;
static BYTE static;
static BYTE sUnderline;
static BYTE sStrikeOut;
static char sFaceName[20];
static char toggler;
static int lnEdit;

switch (message)
{
case WM_INITDIALOG:
InitDlg(hDlg);
AddSpeed(hDlg);
SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_DIR,
DDL_READWRITE | DDL_DIRECTORY, (LPARAM)((LPSTR) "*.BMP"));
SetFocus(GetDlgItem(hDlg, IDP_LISTFILE));
SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_SETCURSEL, 0, 0);
SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_GETTEXT, 0, (LPARAM)((LPSTR) szFile));
SetDlgItemText(hDlg, IDP_FILE, szFile);
SetDlgItemText(hDlg, IDP_DELAY, "1");
sHeight = -16;
sWeight = 400;
static = 0;
sUnderline = 0;
sStrikeOut = 0;

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่สามารถคัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    strcpy(szFaceName, "Modern");
    dwFontColor = RGB(0,0,255);
windex = 0;
SendDlgItemMessage(hDlg, IDP_LIST, LB_SETCURSEL, windex, 0);
if(Stepr.step == 0)
{
    EnableWindow(GetDlgItem(hDlg, IDP_EDIT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_INSERT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_DEL), FALSE);
}
if(Stepr.step >= 1){
    for(a = 0; a <= (Stepr.step - 1); a++){
        ChangeOut(a;
        _itoc(a+1, lntf, 10);
        _itoc(Getshw[a].shFile, szlmatf, 10);
        strcpy(szlstr, Getshw[a].shFile); strcat(szlstr, " | ");
        strcat(szlstr, Getshw[a].shText); strcat(szlstr, " | ");
        strcat(szlstr, szPsh); strcat(szlstr, " | ");
        strcat(szlstr, szTmsh); strcat(szlstr, " | ");
        strcat(szlstr, szTmsh); strcat(szlstr, " | ");
        strcat(szlstr, szTmatf); strcat(szlstr, " } ");
        strcat(szlstr, szPen); strcat(szlstr, " | ");
        strcat(szlstr, szTmPen); strcat(szlstr, " | ");
        strcat(szlstr, szTen); strcat(szlstr, " | ");
        strcat(szlstr, szTmTen); strcat(szlstr, " | ");
        SendDlgItemMessage(hDlg, IDP_LIST, LB_ADDSTRING, 0, (LPARAM)((LPSTR)szlstr));
windex = a;
SendDlgItemMessage(hDlg, IDP_LIST, LB_SETCURSEL, windex, 0);
SetDlgItemText(hDlg, IDP_COUNT, (LPSTR)lntf);
    }
}
break;
case WM_COMMAND:
switch(wParam)
{
    case IDP_OK:
        EndDialog(hDlg, TRUE);
        return (TRUE);
    case IDP_LISTFILE:
        dwindex = SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_GETCURSEL, 0, 0);
        b = dwindex;
        SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_GETTEXT, b, (LPARAM)((LPSTR)szFile));
        SetDlgItemText(hDlg, IDP_FILE, szFile);
        break;
    case IDP_LIST:
        dwindex = SendDlgItemMessage(hDlg, IDP_LIST, LB_GETCURSEL, 0, 0);
        b = dwindex;
        _itoc(b+1, lntf, 10);
        SetDlgItemText(hDlg, IDP_COUNT, (LPSTR)lntf);
        break;

    case IDP_INSERT:
        if(shHeight == 0) {
            MessageBox(hDlg, "You must select Font!", "Error", MB_ICONSTOP | MB_OK);
            break;
        }
        GetDlgItemText(hDlg, IDP_FILE, szFile, 50);
        if(strlen(szFile) == 0)
        {
            MessageBox(hDlg, "You must select BMP file", "Error", MB_ICONSTOP | MB_OK);
            SetFocus(GetDlgItem(hDlg, IDP_FILE));
            break;
        }
        GetDlgItemText(hDlg, IDP_PSH, szPsh, 11);
        GetDlgItemText(hDlg, IDP_TSH, szTsh, 11);
        GetDlgItemText(hDlg, IDP_PEN, szPen, 11);
        GetDlgItemText(hDlg, IDP_TEN, szTen, 11);
        GetDlgItemText(hDlg, IDP_TEXT, szText, 51);
        GetDlgItemText(hDlg, IDP_TMPSH, szTmsh, 2);
        GetDlgItemText(hDlg, IDP_DELAY, szlmatf, 3);
        GetDlgItemText(hDlg, IDP_TMTSH, szTmsh, 2);
        GetDlgItemText(hDlg, IDP_TMPEN, szTmsh, 2);

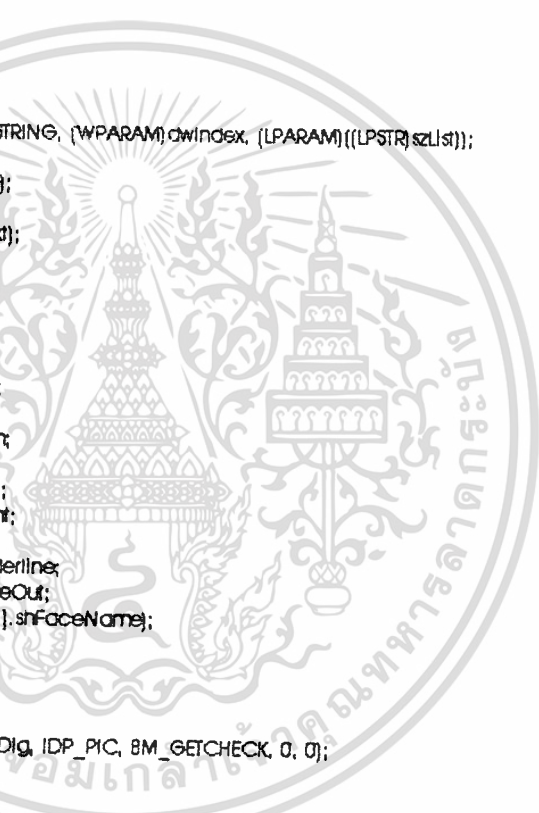
```

```

GetDlgItemText(hDlg, IDP_TMTEN, szTmen, 2);
dwIndex = SendDlgItemMessage(hDlg, IDP_LIST, LB_GETCURREL, 0, 0);
if(dwIndex == LB_ERR){
    MessageBox(hDlg, "You can not Insert!", "Error", MB_ICONSTOP | MB_OK);
    break;
}

b = dwIndex;
if(!strlen(szTmsh) == 0)strcpy(szTmsh, "0");
if(!strlen(szTmsh) == 0)strcpy(szTmsh, "0");
if(!strlen(szTmpen) == 0)strcpy(szTmpen, "0");
if(!strlen(szTmen) == 0)strcpy(szTmen, "0");
if(!strlen(szTmaty) == 0)strcpy(szTmaty, "0");
strcpy(szList, szFile);strcpy(szList, " | ");
strcpy(szList, szText);strcpy(szList, " | ");
strcpy(szList, szPsh);strcpy(szList, " | ");
strcpy(szList, szTmsh);strcpy(szList, " | ");
strcpy(szList, szTsh);strcpy(szList, " | ");
strcpy(szList, szTmsh);strcpy(szList, " | ");
strcpy(szList, szTmaty);strcpy(szList, " | ");
strcpy(szList, szPen);strcpy(szList, " | ");
strcpy(szList, szTpen);strcpy(szList, " | ");
strcpy(szList, szTen);strcpy(szList, " | ");
strcpy(szList, szTmen);strcpy(szList, " | ");
SendDlgItemMessage(hDlg, IDP_LIST, LB_INSERTSTRING, (WPARAM)dwIndex, (LPARAM)((LPSTR)szList);
for(a = 0; a <= b; a++){
    strcpy(Getshw[a].shFile, Getshw[a-1].shFile);
    Getshw[a].shPoc = Getshw[a-1].shPoc;
    strcpy(Getshw[a].shText, Getshw[a-1].shText);
    Getshw[a].shPsh = Getshw[a-1].shPsh;
    Getshw[a].shTsh = Getshw[a-1].shTsh;
    Getshw[a].shPen = Getshw[a-1].shPen;
    Getshw[a].shTen = Getshw[a-1].shTen;
    Getshw[a].shTmaty = Getshw[a-1].shTmaty;
    Getshw[a].shTmsh = Getshw[a-1].shTmsh;
    Getshw[a].shTmsh = Getshw[a-1].shTmsh;
    Getshw[a].shTmpen = Getshw[a-1].shTmpen;
    Getshw[a].shTmen = Getshw[a-1].shTmen;
    Getshw[a].shHeight = Getshw[a-1].shHeight;
    Getshw[a].shWeight = Getshw[a-1].shWeight;
    Getshw[a].shItalic = Getshw[a-1].shItalic;
    Getshw[a].shUnderline = Getshw[a-1].shUnderline;
    Getshw[a].shStrikeOut = Getshw[a-1].shStrikeOut;
    strcpy(Getshw[a].shFaceName, Getshw[a-1].shFaceName);
    Getshw[a].shColor = Getshw[a-1].shColor;
    Getshw[a].shlLong = Getshw[a-1].shlLong;
}
strcpy(Getshw[b].shFile, szFile);
Getshw[b].shPoc = SendDlgItemMessage(hDlg, IDP_PIC, BM_GETCHECK, 0, 0);
strcpy(Getshw[b].shText, szText);
ChangeIn(b);
Getshw[b].shTmsh = szTmsh[0];
Getshw[b].shTmsh = szTmsh[0];
Getshw[b].shTmpen = szTmpen[0];
Getshw[b].shTmen = szTmen[0];
szTmaty = atoi(szTmaty);Getshw[b].shTmaty = szTmaty;
Getshw[b].shHeight = shHeight;
Getshw[b].shWeight = sWeight;
Getshw[b].shItalic = sitalic;
Getshw[b].shUnderline = sUnderline;
Getshw[b].shStrikeOut = sStrikeOut;
strcpy(Getshw[b].shFaceName, sFaceName);
Getshw[b].shColor = dwFontColor;
dwGetset = SendDlgItemMessage(hDlg, IDP_TEXT, EM_GETSEL, 0, 0);
lwGetset = LOWORD(dwGetset);
Getshw[b].shlLong = lwGetset;
Septr.step++;
z++;
break;
case IDP_DELETE:
    dwIndex = SendDlgItemMessage(hDlg, IDP_LIST, LB_GETCURREL, 0, 0);
    if(dwIndex == LB_ERR){
        MessageBox(hDlg, "You can not Delete!", "Error", MB_ICONSTOP | MB_OK);
        break;
    }
    SendDlgItemMessage(hDlg, IDP_LIST, LB_DELETESTRING, (WPARAM)dwIndex, 0);

```



```

    b = dwinde;
    for(a = b; a <= Stepr.step-2; a++){
        lstrcpy(Getshw[a].shFile, Getshw[a+1].shFile);
        lstrcpy(Getshw[a].shText, Getshw[a+1].shText);
        Getshw[a].shPoc = Getshw[a+1].shPoc;
        Getshw[a].shPsh = Getshw[a+1].shPsh;
        Getshw[a].shTsh = Getshw[a+1].shTsh;
        Getshw[a].shPen = Getshw[a+1].shPen;
        Getshw[a].shTen = Getshw[a+1].shTen;
        Getshw[a].shTmaty = Getshw[a+1].shTmaty;
        Getshw[a].shTmsh = Getshw[a+1].shTmsh;
        Getshw[a].shTmtsh = Getshw[a+1].shTmtsh;
        Getshw[a].shTmpen = Getshw[a+1].shTmpen;
        Getshw[a].shTmten = Getshw[a+1].shTmten;
        Getshw[a].shHeight = Getshw[a+1].shHeight;
        Getshw[a].shWeight = Getshw[a+1].shWeight;
        Getshw[a].shItalic = Getshw[a+1].shItalic;
        Getshw[a].shUnderline = Getshw[a+1].shUnderline;
        Getshw[a].shStrikeOut = Getshw[a+1].shStrikeOut;
        lstrcpy(Getshw[a].shFaceName, Getshw[a+1].shFaceName);
        Getshw[a].shColor = Getshw[a+1].shColor;
        Getshw[a].shLong = Getshw[a+1].shLong;
    }
    Stepr.step--;
    z--;
    winde = b;
    SendDlgItemMessage(hDlg, IDP_LIST, LB_SETCURSEL, winde, 0);
    if(b > z-1){
        winde = z-1;
        SendDlgItemMessage(hDlg, IDP_LIST, LB_SETCURSEL, winde, 0);
        _itoa(z, intr, 10);
        SetDlgItemText(hDlg, IDP_COUNT, (LPSTR)intr);
    }
    break;
case IDP_FONT:
    wsize = sizeof(CHOOSFONT);
    if (!!(lpFontChunk = (LPCHOOSEFONT)AllocAndLockMem(&nFontChunk, wsize)))
        break;
    InitializeStruct((LPSTR)lpFontChunk;
    lpFontChunk->hDC = NULL;
    hDC = GetDC(ghWnd);
    lLogPixsY = GetDeviceCaps(hDC, LOGPIXELSY);
    ReleaseDC(ghWnd, hDC);
    if. shHeight = -1 * (lLogPixsY * DESIREDPOINTSIZ / 72);
    if. shWidth = 0;
    if. shEscapement = 0;
    if. shOrientation = 0;
    if. shWeight = 400;
    if. shItalic = 0;
    if. shUnderline = 0;
    if. shStrikeOut = 0;
    if. shCharSet = ANSI_CHARSET;
    if. shOutPrecision = OUT_DEFAULT_PRECIS;
    if. shClipPrecision = CLIP_DEFAULT_PRECIS;
    if. shQuality = DEFAULT_QUALITY;
    if. shPitchAndFamily = DEFAULT_PITCH | FF_DONTCARE;
    lstrcpy(if. shFaceName, "Modern");
    if (toggle == 'e')
    {
        if. shHeight = Getshw[lnEdit].shHeight;
        if. shWeight = Getshw[lnEdit].shWeight;
        if. shItalic = Getshw[lnEdit].shItalic;
        if. shUnderline = Getshw[lnEdit].shUnderline;
        if. shStrikeOut = Getshw[lnEdit].shStrikeOut;
        lstrcpy(if. shFaceName, Getshw[lnEdit].shFaceName);
        lpFontChunk->rgbColors = Getshw[lnEdit].shColor;
    }
    lpFontChunk->lLogFont = &lf;
    EnableWindow(GetDlgItem(hDlg, IDP_EDIT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_INSERT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_DEL), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_ADD), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_OK), FALSE);

```

งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นที่มีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EnableWindow(GetDlgItem(hDlg, IDP_FONT), FALSE);
if { ChooseFont(lpFontChunk) //Now let's create the selected font!
{
    if (gSelectedFont) DeleteObject(gSelectedFont);
    gSelectedFont = CreateFontIndirect((LPLOGFONT)(lpFontChunk->lpLogFont));
    dwFontColor=lpFontChunk->rgbColors;
    InvalidateRect(gHwnd, NULL, TRUE);
    shHeight = if.ifHeight;
    swWeight = if.ifWeight;
    sIfalic = if.ifItalic;
    sUnderscore = if.ifUnderline;
    sStrikeOut = if.ifStrikeOut;
    lstrcpy(sFaceName, if.ifFaceName);
}
EnableWindow(GetDlgItem(hDlg, IDP_EDIT), TRUE);
EnableWindow(GetDlgItem(hDlg, IDP_INSERT), TRUE);
EnableWindow(GetDlgItem(hDlg, IDP_DEL), TRUE);
EnableWindow(GetDlgItem(hDlg, IDP_ADD), TRUE);
EnableWindow(GetDlgItem(hDlg, IDP_OK), TRUE);
EnableWindow(GetDlgItem(hDlg, IDP_FONT), TRUE);

if(toggle == 'e')
{
    EnableWindow(GetDlgItem(hDlg, IDP_EDIT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_INSERT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_DEL), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_OK), FALSE);
}
if(!repr.step == 0)
{
    EnableWindow(GetDlgItem(hDlg, IDP_EDIT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_INSERT), FALSE);
    EnableWindow(GetDlgItem(hDlg, IDP_DEL), FALSE);
}
break;

case IDP_ADD:
if(toggle == 'e')
{
    a = z;
    z = lnEdit;
}
if(shHeight == 0) {
    MessageBox(hDlg, "You must select Font!", "Error", MB_ICONSTOP | MB_OK);
    break;
}
GetDlgItemText(hDlg, IDP_FILE, szFile, 50);
if(strlen(szFile) == 0)
{
    MessageBox(hDlg, "You must select BMP file", "Error", MB_ICONSTOP | MB_OK);
    SetFocus(GetDlgItem(hDlg, IDP_FILE));
    break;
}
GetDlgItemText(hDlg, IDP_PSH, szPsh, 11);
GetDlgItemText(hDlg, IDP_TSH, szTsh, 11);
GetDlgItemText(hDlg, IDP_PEN, szPen, 11);
GetDlgItemText(hDlg, IDP_TEN, szTen, 11);
GetDlgItemText(hDlg, IDP_TEXT, szText, 51);
GetDlgItemText(hDlg, IDP_DELAY, szImdy, 3);
GetDlgItemText(hDlg, IDP_TMPSH, szImpsh, 2);
GetDlgItemText(hDlg, IDP_TMTSH, szImtsh, 2);
GetDlgItemText(hDlg, IDP_TMPEN, szImpen, 2);
GetDlgItemText(hDlg, IDP_TMTEN, szImten, 2);
dwGetset = SendDlgItemMessage(hDlg, IDP_TEXT, EM_GETSEL, 0, 0);
lwGetset = LOWORD(dwGetset);
if(lwGetset == 0) lwGetset = strlen(szText);
if(strlen(szImpsh) == 0) strcpy(szImpsh, "0");
if(strlen(szImtsh) == 0) strcpy(szImtsh, "0");
if(strlen(szImpen) == 0) strcpy(szImpen, "0");
if(strlen(szImten) == 0) strcpy(szImten, "0");
if(strlen(szImdy) == 0) strcpy(szImdy, "0");
GetSet[z].shLong = lwGetset;
    _itoc(z+1, Infr, 10);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        strcpy(szList, szFile); strcat(szList, " | ");
        strcat(szList, szText); strcat(szList, " | ");
        strcat(szList, szPsh); strcat(szList, " | ");
        strcat(szList, szImpsh); strcat(szList, " | ");
        strcat(szList, szTsh); strcat(szList, " | ");
        strcat(szList, szTmsh); strcat(szList, " | ");
        strcat(szList, szTmaly); strcat(szList, " | ");
        strcat(szList, szPen); strcat(szList, " | ");
        strcat(szList, szTpen); strcat(szList, " | ");
        strcat(szList, szTen); strcat(szList, " | ");
        strcat(szList, szTten); strcat(szList, " | ");
    if(toggle == 'e')
    {
        SendDlgItemMessage(hDlg, IDP_LIST, LB_DELETESTRING, z, 0);
        SendDlgItemMessage(hDlg, IDP_LIST, LB_INSERTSTRING, z, (LPARAM)((LPSTR)szList));
    }
    else
        SendDlgItemMessage(hDlg, IDP_LIST, LB_ADDSTRING, 0, (LPARAM)((LPSTR)szList));
    SetDlgItemText(hDlg, IDP_COUNT, (LPSTR)Intr);
    Getshw[z].shPoc = SendDlgItemMessage(hDlg, IDP_PIC, BM_GETCHECK, 0, 0);
    strcpy(Getshw[z].shFile, szFile);
    strcpy(Getshw[z].shText, szText);
    ChangeIn(z);
    Getshw[z].shImpsh = szImpsh[0];
    Getshw[z].shTmsh = szTmsh[0];
    Getshw[z].shTpen = szTpen[0];
    Getshw[z].shTten = szTten[0];
    szTmaly = atoi(szTmaly); Getshw[z].shTmaly = szTmaly;
    Getshw[z].shHeight = shHeight;
    Getshw[z].shWeight = shWeight;
    Getshw[z].shItalic = shItalic;
    Getshw[z].shUnderline = shUnderline;
    Getshw[z].shStrikeOut = shStrikeOut;
    strcpy(Getshw[z].shFaceName, szFaceName);
    Getshw[z].shColor = dwFontColor;
    windex = z;
    SendDlgItemMessage(hDlg, IDP_LIST, LB_SETCURSEL, windex, 0);
    if(toggle == 'e')
    {
        z = a-1;
        z++;
        SetPr.step = z;
        SetFocus(GetDlgItem(hDlg, IDP_FILE));
        EnableWindow(GetDlgItem(hDlg, IDP_EDIT), TRUE);
        EnableWindow(GetDlgItem(hDlg, IDP_INSERT), TRUE);
        EnableWindow(GetDlgItem(hDlg, IDP_DELETE), TRUE);
        EnableWindow(GetDlgItem(hDlg, IDP_ADD), TRUE);
        EnableWindow(GetDlgItem(hDlg, IDP_OK), TRUE);
        EnableWindow(GetDlgItem(hDlg, IDP_FONT), TRUE);
        toggle = 'a';
        break;
    }

case IDP_PIC:
    SetPic(hDlg);
    break;

case IDP_COL:
    SetCol(hDlg);
    break;

case IDP_EDIT:
    dwindex = SendDlgItemMessage(hDlg, IDP_LIST, LB_SETCURSEL, 0, 0);
    b = dwindex;
    EditDlg(hDlg, b);
    SetDlgItemText(hDlg, IDP_TEXT, Getshw[b].shText);
    _itoc(Getshw[b].shTmaly, Intr, 10);
    SetDlgItemText(hDlg, IDP_DELAY, Intr);
    if(Getshw[b].shPoc == 1)
    {
        SendDlgItemMessage(hDlg, IDP_PIC, BM_SETCHECK, TRUE, 0);
        SendDlgItemMessage(hDlg, IDP_COL, BM_SETCHECK, 0, 0);
        SetPic(hDlg);
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    SendDlgItemMessage(hDlg, IDP_COL, BM_SETCHECK, TRUE, 0);
    SendDlgItemMessage(hDlg, IDP_PIC, BM_SETCHECK, 0, 0);
    SetCol(hDlg);
}
EnableWindow(GetDlgItem(hDlg, IDP_EDIT), FALSE);
EnableWindow(GetDlgItem(hDlg, IDP_INSERT), FALSE);
EnableWindow(GetDlgItem(hDlg, IDP_DELETE), FALSE);
EnableWindow(GetDlgItem(hDlg, IDP_OK), FALSE);
InEdit = b;
shHeight = Getshw(InEdit).shHeight;
shWeight = Getshw(InEdit).shWeight;
shItalic = Getshw(InEdit).shItalic;
shUnderline = Getshw(InEdit).shUnderline;
shStrikeOut = Getshw(InEdit).shStrikeOut;
strcpy(shFaceName, Getshw(InEdit).shFaceName);
dwFontColor = Getshw(InEdit).shColor;
Toggle = 'e';
break;
default:
    break;
}
break;
default:
    break;
}
return (FALSE);
}

```

// ฟังก์ชัน: MakeIndexHeader(ipInfo)

// ใช้ BITMAPINFOHEADER สร้างข้อมูลโดยใช้รูปแบบ DIB_PAL_COLORS

```

HANDLE PASCAL NEAR MakeIndexHeader(LPBITMAPINFOHEADER ipInfo)
{
    LPBITMAPINFOHEADER ipPalInfo;
    WORD FAR *IpTable;
    WORD i;
    if (!ipInfo->biClrUsed)
    {
        hPalInfo = GlobalAlloc(GMEM_MOVEABLE, ipInfo->biSize + ipInfo->biClrUsed * sizeof(WORD));
        if (!hPalInfo)
            return(NULL);
        ipPalInfo = (LPBITMAPINFOHEADER)GlobalLock(hPalInfo);
        *ipPalInfo = *ipInfo;
        IpTable = (WORD FAR *)((LPSTR)ipPalInfo + ipPalInfo->biSize);
        for (i = 0; i < (WORD)ipInfo->biClrUsed; i++)
            *IpTable++ = i;
        GlobalUnlock(hPalInfo);
        return(hPalInfo);
    }
    else
        return(NULL);
}

```

// ฟังก์ชัน: MakeDIBPalette(ipInfo)

// ใช้ BITMAPINFOHEADER สร้างงานสีพื้นฐานบนตารางสี

```

HPALETTE PASCAL NEAR MakeDIBPalette(LPBITMAPINFOHEADER ipInfo)
{
    NPLOGPALETTE npPal(36);
    RGBQUAD far *pRGB;
    HPALETTE hLogPal;
    WORD i;
    int lpp, pet;
    static s;
    /* since biClrUsed field was filled during the loading of the DIB,
    ** we know it contains the number of colors in the color table */
    if (!ipInfo->biClrUsed)
    {
        lpp = sizeof(LOGPALETTE);
        pet = 0;
        for (i = 0; i < ipInfo->biClrUsed; i++)
        {
            pRGB = (RGBQUAD far *)GlobalLock(hLogPal);
            pRGB[i].rgbRed = i;
            pRGB[i].rgbGreen = i;
            pRGB[i].rgbBlue = i;
            pRGB[i].rgbReserved = 0;
            pet++;
        }
        GlobalUnlock(hLogPal);
        npPal->palEntries = pet;
        hLogPal = CreatePalette(npPal);
    }
    else
    {
        lpp = sizeof(LOGPALETTE);
        pet = 0;
        for (i = 0; i < ipInfo->biClrUsed; i++)
        {
            pRGB = (RGBQUAD far *)GlobalLock(hLogPal);
            pRGB[i].rgbRed = ipInfo->biColorTable[i].rgbRed;
            pRGB[i].rgbGreen = ipInfo->biColorTable[i].rgbGreen;
            pRGB[i].rgbBlue = ipInfo->biColorTable[i].rgbBlue;
            pRGB[i].rgbReserved = ipInfo->biColorTable[i].rgbReserved;
            pet++;
        }
        GlobalUnlock(hLogPal);
        npPal->palEntries = pet;
        hLogPal = CreatePalette(npPal);
    }
    return(hLogPal);
}

```



การทวงเงินไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pet = sizeof(PALETTEENTRY);
npPal[s] = (NLOGPALETTE)LocalAlloc(LMEM_FIXED, sizeof(LOGPALETTE) +
    (WORD)lpInfo->biClrUsed * sizeof(PALETTEENTRY));
    if (!npPal[s])
        return(FALSE);
    npPal[s]->palVersion = 0x300;
    npPal[s]->palNumEntries = (WORD)lpInfo->biClrUsed;
    /* get pointer to the color table */
    lpRGB = (RGBQUAD FAR *)((LPSTR)lpInfo + lpInfo->biSize);
    /* copy colors from the color table to the LogPalette structure */
    for (i = 0; i < (WORD)lpInfo->biClrUsed; i++, lpRGB++)
    {
        npPal[s]->palPalEntry[i].peRed = lpRGB->rgbRed;
        npPal[s]->palPalEntry[i].peGreen = lpRGB->rgbGreen;
        npPal[s]->palPalEntry[i].peBlue = lpRGB->rgbBlue;
        npPal[s]->palPalEntry[i].peFlags = 0;
    }
    hLogPal = CreatePalette((LPLOGPALETTE)npPal[s]);
    LocalFree((HANDLE)npPal[s]);
s++;
return(hLogPal);
}

/* 24-bit DIB with no color table. return default palette. Another
** option would be to create a 256 color "rainbow" palette to provide
** some good color choices. */
else
    return(GetStockObject(DEFAULT_PALETTE));
}

```

// ฟังก์ชัน: PaintDIB(hWnd)
// ใช้ในการโหลด DIB โดยใช้ข้อมูลของรูปที่ผู้ใช้เลือกเป็นข้อมูล

```

void PASCAL NEAR PaintDIB(HWND hWnd)
{
    HDC hDC;
    LPBITMAPINFOHEADER lpInfo;
    LPBITMAPINFOHEADER lpHeader;
    HPALETTE hOldPal;
    WORD wDIBUse;
    lpInfo = (LPBITMAPINFOHEADER) GlobalLock(hDIBInfo);
    /* if using a palette for drawing and there is actually
    ** a color table, then get a palette and realize it. */
    if (wPalOp && lpInfo->biClrUsed)
    {
        if (!hPalette)
            hPalette = MakeDIBPalette(lpInfo);
        if (!hPalette)
        {
            MessageBox(hWnd, "Can't create palette", "Error", MB_ICONSTOP | MB_OK);
            goto ExitTime;
        }
        hOldPal = SelectPalette(hDC, hPalette, FALSE);
        RealizePalette(hDC);
        /* if using DIB_PAL_COLORS and the bitmap is not 24-bit,
        ** then use a header with a color table of indices. */
        if (wPalOp = IDM_PAIND && (lpInfo->biBitCount != 24))
        {
            if (!hPalHeader)
            {
                hPalHeader = MakeIndexHeader(lpInfo);
            }
            if (!hPalHeader)
            {
                MessageBox(hWnd, "Can't create indexed color table", "Error", MB_ICONSTOP | MB_OK);
                goto ExitTime;
            }
            lpHeader = (LPBITMAPINFOHEADER) GlobalLock(hPalHeader);
            wDIBUse = DIB_PAL_COLORS;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ๖
 ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        lpHeader = lpInfo;
        wDIBUse = DIB_RGB_COLORS;
    }
}
else
{
    lpHeader = lpInfo;
    wDIBUse = DIB_RGB_COLORS;
}

/* NOTE: because driver does not do the StretchDIB
** itself, this is not a fast operation.
** internally it converts to CreateDIBitmap
** followed by a StretchBlt
*/

/* clean up any palette work */
if (wPalOp)
{
    SelectPalette(hDC, hOldPal, FALSE);
    if (wPalOp == IDM_PALETTE)
        GlobalUnlock(hPalHeader);
}

ExitTime:
GlobalUnlock(hDIBInfo);
}

```

// ฟังก์ชัน: ReadDIB(hWnd)

// ใช้อ่าน DIB จากแฟ้มข้อมูล และโหลดแฟ้มข้อมูล

```

int ReadDIB(HWND hWnd)
{
    unsigned          fh;
    LPBITMAPINFOHEADER lpbi;
    OFSTRUCT          of;
    BITMAPFILEHEADER bf;
    WORD              nNumColors;
    WORD              result = FALSE; /* assume failure */
    /* Open the file and get a handle to its BITMAPINFO */
    fh = OpenFile(achFileName, &of, OF_READ);
    if (fh == -1) {
        wprintf(str, "Can't open file %ls", (LPSTR)achFileName);
        MessageBox(hWnd, str, "Error", MB_ICONSTOP | MB_OK);
        return (FALSE);
    }
    lpbi = (LPBITMAPINFOHEADER) GlobalLock(hDIBInfo);
    /* read the BITMAPFILEHEADER */
    if (sizeof (bf) != _read (fh, (LPSTR)&bf, sizeof (bf)))
        goto ErrExit;
    if (bf.bfType != 0x4d42) /* 'BM' */
        goto ErrExit;
    if (sizeof (BITMAPINFOHEADER) != _read (fh, (LPSTR)lpbi, sizeof (BITMAPINFOHEADER)))
        goto ErrExit;
    /*!!!! for now, don't even deal with CORE headers */
    if (lpbi->biSize == sizeof (BITMAPCOREHEADER))
        goto ErrExit;
    if (!(nNumColors = (WORD)lpbi->biClrUsed))
    {
        /* no color table for 24-bit, default size otherwise */
        if (lpbi->biBitCount != 24)
            nNumColors = 1 << lpbi->biBitCount; /* standard size table */
    }
    /*!!!! In some default values if they are zero */
    if (lpbi->biClrUsed == 0)
        lpbi->biClrUsed = (DWORD)nNumColors;
    if (lpbi->biSizeImage == 0)
    {
        lpbi->biSizeImage = (((lpbi->biWidth * (DWORD)lpbi->biBitCount) + 31) & ~31) >> 3;
        *lpbi->biHeight;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

/* get a proper-sized buffer for header, color table and bits */

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hDIBInfo = GlobalReAlloc(hDIBInfo, lpbpl->biSize + nNumColors * sizeof(RGBQUAD) + lpbpl->biSizeImage, 0);
if (!hDIBInfo) /* can't resize buffer for loading */
    goto ErrExit2;
lpbpl = (LPBITMAPINFOHEADER)GlobalLock(hDIBInfo);
/* read the color table */
_read(tfh, (LPSTR)lpbpl + lpbpl->biSize, nNumColors * sizeof(RGBQUAD));
/* offset to the bits from start of DIB header */
offBits = (WORD)lpbpl->biSize + nNumColors * sizeof(RGBQUAD);
if (bf.biOffBits != 0)
{
    _lseek(tfh, bf.biOffBits, SEEK_SET);
}
if (lpbpl->biSizeImage == lread(tfh, (LPSTR)lpbpl + offBits, lpbpl->biSizeImage))
    result = TRUE;
ErrExit:
_close(tfh);
GlobalUnlock(hDIBInfo);
ErrExit2:
return(result);
}

```

```

// โปรแกรมย่อยสำหรับอ่านข้อมูลที่มากกว่า 64 กิโลไบต์
// ฟังก์ชัน: lread(int fh, VOID FAR *pv, DWORD ul)
// อ่านข้อมูลทั้งหมดโดยครั้งละ 32 กิโลไบต์

```

```

DWORD PASCAL lread (int fh, VOID far *pv, DWORD ul)
{
    DWORD    ulT = ul;
    BYTE huge *hp = pv;
    while (ulT > (DWORD)MAXREAD) {
        if (_lread(fh, (LPSTR)hp, (WORD)MAXREAD) != MAXREAD)
            return 0;
        ulT -= MAXREAD;
        hp += MAXREAD;
    }
    if (_lread(fh, (LPSTR)hp, (WORD)ulT) != (WORD)ulT)
        return 0;
    return ulT;
}

```

```

// ฟังก์ชัน: InitDIB(HWND)
// ลบข้อมูล DIB ของเก่า และอ่านข้อมูลของ DIB ใหม่จากแฟ้มข้อมูล

```

```

int InitDIB(HWND hwnd)
{
    LPBITMAPINFOHEADER lpbpl;
    WORD wDIBUser;
    /* If there was an old DIB, free it up */
    if (bDIBLoaded)
    {
        if (hDDBltMap)
        {
            SelectObject(hMemDC, hOldBltMap);
            DeleteDC(hMemDC);
            DeleteObject(hDDBltMap);
            hDDBltMap = NULL;
        }
        bDIBLoaded = FALSE;
        if (hPalette)
        {
            DeleteObject(hPalette);
            hPalette = 0;
        }
        if (hPalHeader)
        {
            GlobalFree(hPalHeader);
            hPalHeader = NULL;
        }
    }
}

```



```

}

if(szTsh[1] == '0') Getshw[qd].shTsh = 1;
if(szTsh[1] == '0') Getshw[qd].shTsh = 2;
if(szPen[0] == '1') Getshw[qd].shPen = 1;
if(szPen[0] == '1') Getshw[qd].shPen = 2;
if(szPen[0] == '0') Getshw[qd].shPen = 3;
if(szPen[0] == '3') Getshw[qd].shPen = 4;
if(szPen[0] == '1') Getshw[qd].shPen = 5;
if(szPen[0] == '0') Getshw[qd].shPen = 6;
if(szPen[0] == '0') Getshw[qd].shPen = 7;
if(szTen[1] == '0') Getshw[qd].shTen = 1;
if(szTen[1] == '0') Getshw[qd].shTen = 2;
}

```

void ChangeOut(int qd)

```

{
    if(Getshw[qd].shPsh == 1) Istrcpy(szPsh,"Form up");
    if(Getshw[qd].shPsh == 2) Istrcpy(szPsh,"Form down");
    if(Getshw[qd].shPsh == 4) Istrcpy(szPsh,"Slide down");
    if(Getshw[qd].shPsh == 4) Istrcpy(szPsh,"Slide side");
    if(Getshw[qd].shPsh == 5) Istrcpy(szPsh,"Form cent.");
    if(Getshw[qd].shPsh == 6) Istrcpy(szPsh,"Line hor.");
    if(Getshw[qd].shPsh == 7) Istrcpy(szPsh,"Line vert.");
    if(Getshw[qd].shTsh == 1) Istrcpy(szTsh,"Horizontal");
    if(Getshw[qd].shTsh == 2) Istrcpy(szTsh,"Vertical");
    if(Getshw[qd].shPen == 1) Istrcpy(szPen,"To down");
    if(Getshw[qd].shPen == 2) Istrcpy(szPen,"To right");
    if(Getshw[qd].shPen == 4) Istrcpy(szPen,"Slide down");
    if(Getshw[qd].shPen == 4) Istrcpy(szPen,"Slide side");
    if(Getshw[qd].shPen == 5) Istrcpy(szPen,"To cent.");
    if(Getshw[qd].shPen == 6) Istrcpy(szPen,"Line hor.");
    if(Getshw[qd].shPen == 7) Istrcpy(szPen,"Line vert.");
    if(Getshw[qd].shTen == 1) Istrcpy(szTen,"Horizontal");
    if(Getshw[qd].shTen == 2) Istrcpy(szTen,"Vertical");
}

```

void EditDig(HWND hDig,int bj)

```

{
    if(Getshw[bj].shPsh == 1) SendDlgItemMessage(hDig, IDP_PSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Form up"));
    if(Getshw[bj].shPsh == 2) SendDlgItemMessage(hDig, IDP_PSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Form left");
    if(Getshw[bj].shPsh == 3) SendDlgItemMessage(hDig, IDP_PSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Slide down");
    if(Getshw[bj].shPsh == 4) SendDlgItemMessage(hDig, IDP_PSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Slide side");
    if(Getshw[bj].shPsh == 5) SendDlgItemMessage(hDig, IDP_PSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Form cent.");
    if(Getshw[bj].shPsh == 6) SendDlgItemMessage(hDig, IDP_PSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Line hor.");
    if(Getshw[bj].shPsh == 7) SendDlgItemMessage(hDig, IDP_PSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Line vert.");
    if(Getshw[bj].shTsh == 1) SendDlgItemMessage(hDig, IDP_TSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Horizontal");
    if(Getshw[bj].shTsh == 2) SendDlgItemMessage(hDig, IDP_TSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Vertical");
    if(Getshw[bj].shPen == 1) SendDlgItemMessage(hDig, IDP_PEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"To down");
    if(Getshw[bj].shPen == 2) SendDlgItemMessage(hDig, IDP_PEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"To right");
    if(Getshw[bj].shPen == 3) SendDlgItemMessage(hDig, IDP_PEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Slide down");
    if(Getshw[bj].shPen == 4) SendDlgItemMessage(hDig, IDP_PEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Slide side");
    if(Getshw[bj].shPen == 5) SendDlgItemMessage(hDig, IDP_PEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"To cent.");
    if(Getshw[bj].shPen == 6) SendDlgItemMessage(hDig, IDP_PEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Line hor.");
    if(Getshw[bj].shPen == 7) SendDlgItemMessage(hDig, IDP_PEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Line vert.");
    if(Getshw[bj].shTen == 1) SendDlgItemMessage(hDig, IDP_TEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Horizontal");
    if(Getshw[bj].shTen == 2) SendDlgItemMessage(hDig, IDP_TEN, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"Vertical");
    if(Getshw[bj].shImpsh == 1) SendDlgItemMessage(hDig, IDP_TMPSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"1");
    if(Getshw[bj].shImpsh == 2) SendDlgItemMessage(hDig, IDP_TMPSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"2");
    if(Getshw[bj].shImpsh == 3) SendDlgItemMessage(hDig, IDP_TMPSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"3");
    if(Getshw[bj].shImpsh == 4) SendDlgItemMessage(hDig, IDP_TMPSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"4");
    if(Getshw[bj].shImpsh == 5) SendDlgItemMessage(hDig, IDP_TMPSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"5");
    if(Getshw[bj].shImpsh == 6) SendDlgItemMessage(hDig, IDP_TMPSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"6");
    if(Getshw[bj].shImpsh == 7) SendDlgItemMessage(hDig, IDP_TMPSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"7");
    if(Getshw[bj].shTmsh == 1) SendDlgItemMessage(hDig, IDP_TMTSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"1");
    if(Getshw[bj].shTmsh == 2) SendDlgItemMessage(hDig, IDP_TMTSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"2");
    if(Getshw[bj].shTmsh == 3) SendDlgItemMessage(hDig, IDP_TMTSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"3");
    if(Getshw[bj].shTmsh == 4) SendDlgItemMessage(hDig, IDP_TMTSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"4");
    if(Getshw[bj].shTmsh == 5) SendDlgItemMessage(hDig, IDP_TMTSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"5");
    if(Getshw[bj].shTmsh == 6) SendDlgItemMessage(hDig, IDP_TMTSH, CB_SELECTSTRING,-1,(LPARAM)((LPSTR)"6");
}

```

โยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุผลเบื้องต้นที่ และต้องอ้างอิงถึงเจ้าของเอกสารที่กรงที่มีการนำไปใช้

```

if((Getshw[b].shTmsh == 7) SendDlgItemMessage(hDlg, IDP_TMTSH, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"7"));
if((Getshw[b].shTmsh == 1) SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"1"));
if((Getshw[b].shTmsh == 2) SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"2"));
if((Getshw[b].shTmsh == 3) SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"3"));
if((Getshw[b].shTmsh == 4) SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"4"));
if((Getshw[b].shTmsh == 5) SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"5"));
if((Getshw[b].shTmsh == 6) SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"6"));
if((Getshw[b].shTmsh == 7) SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"7"));
if((Getshw[b].shTmten == 1) SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"1"));
if((Getshw[b].shTmten == 2) SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"2"));
if((Getshw[b].shTmten == 3) SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"3"));
if((Getshw[b].shTmten == 4) SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"4"));
if((Getshw[b].shTmten == 5) SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"5"));
if((Getshw[b].shTmten == 6) SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"6"));
if((Getshw[b].shTmten == 7) SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"7"));
}

```

void AddSpeed(HWND hDlg

```

{
SendDlgItemMessage(hDlg, IDP_TMPH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"1"));
SendDlgItemMessage(hDlg, IDP_TMPH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"2"));
SendDlgItemMessage(hDlg, IDP_TMPH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"3"));
SendDlgItemMessage(hDlg, IDP_TMPH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"4"));
SendDlgItemMessage(hDlg, IDP_TMPH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"5"));
SendDlgItemMessage(hDlg, IDP_TMPH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"6"));
SendDlgItemMessage(hDlg, IDP_TMPH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"7"));
SendDlgItemMessage(hDlg, IDP_TMPH, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"1"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"1"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"2"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"3"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"4"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"5"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"6"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"7"));
SendDlgItemMessage(hDlg, IDP_TMTSH, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"1"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"1"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"2"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"3"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"4"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"5"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"6"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"7"));
SendDlgItemMessage(hDlg, IDP_TMPEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"1"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"1"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"2"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"3"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"4"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"5"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"6"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"7"));
SendDlgItemMessage(hDlg, IDP_TMTEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"1"));
}

```

void InitDlg(HWND hDlg

```

{
SendDlgItemMessage(hDlg, IDP_PIC, BM_SETCHECK, TRUE, 0);
SendDlgItemMessage(hDlg, IDP_PSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Form up"));
SendDlgItemMessage(hDlg, IDP_PSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Form left"));
SendDlgItemMessage(hDlg, IDP_PSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Slide down"));
SendDlgItemMessage(hDlg, IDP_PSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Slide side"));
SendDlgItemMessage(hDlg, IDP_PSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Form cent.));
SendDlgItemMessage(hDlg, IDP_PSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Line hor.));
SendDlgItemMessage(hDlg, IDP_PSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Line ver.));
SendDlgItemMessage(hDlg, IDP_PEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"To down"));
SendDlgItemMessage(hDlg, IDP_PEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"To right"));
SendDlgItemMessage(hDlg, IDP_PEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"slide down"));
SendDlgItemMessage(hDlg, IDP_PEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"slide side"));
SendDlgItemMessage(hDlg, IDP_PEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"To cent.));
SendDlgItemMessage(hDlg, IDP_PEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Line hor.));
SendDlgItemMessage(hDlg, IDP_PEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Line ver.));
SendDlgItemMessage(hDlg, IDP_TSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Horizontal"));
SendDlgItemMessage(hDlg, IDP_TSH, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Vertical"));
SendDlgItemMessage(hDlg, IDP_TEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Horizontal"));

```

เอกสาร
 ไม่มีการเผยแพร่ทางอื่น ยกเว้นผู้มีเหตุพิเศษขออนุญาต และต้องจ่ายค่าธรรมเนียมของเอกสารทุกครั้งที่มีการนำไปใช้



```

SendDlgItemMessage(hDlg, IDP_TEN, CB_ADDSTRING, 0, (LPARAM)((LPSTR)"Vertical");
SendDlgItemMessage(hDlg, IDP_PSH, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"Form up");
SendDlgItemMessage(hDlg, IDP_TSH, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"Horizontal");
SendDlgItemMessage(hDlg, IDP_PEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"To down");
SendDlgItemMessage(hDlg, IDP_TEN, CB_SELECTSTRING, -1, (LPARAM)((LPSTR)"Horizontal");
}

void SetPic(HWND hDlg)
{
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_RESETCONTENT, 0, 0);
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_DIR, DDL_READWRITE | DDL_DIRECTORY, (LPARAM)((LPSTR)**".BMP");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_GETTEXT, 0, (LPARAM)((LPSTR)szFile);
    SetDlgItemText(hDlg, IDP_FILE, szFile);
}

void SetCol(HWND hDlg)
{
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_RESETCONTENT, 0, 0);
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_ADDSTRING, 0, (LPARAM)((LPSTR)"Red");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_ADDSTRING, 0, (LPARAM)((LPSTR)"Green");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_ADDSTRING, 0, (LPARAM)((LPSTR)"Blue");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_ADDSTRING, 0, (LPARAM)((LPSTR)"Yellow");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_ADDSTRING, 0, (LPARAM)((LPSTR)"Violet");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_ADDSTRING, 0, (LPARAM)((LPSTR)"White");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_ADDSTRING, 0, (LPARAM)((LPSTR)"Gray");
    SendDlgItemMessage(hDlg, IDP_LISTFILE, LB_GETTEXT, 0, (LPARAM)((LPSTR)szFile);
    SetDlgItemText(hDlg, IDP_FILE, szFile);
}

void ShowRect(HWND hWnd, int Rewth, int Rehg)
{
    RECT Rectangle;
    Rectangle.left = 0;
    Rectangle.top = 0;
    Rectangle.right = Rewth;
    Rectangle.bottom = Rehg;
    /* Compute the size of the window rectangle based on the given
    * client rectangle size and the window style, then size the
    * window. Do not deal with possibility of more than one menu line.
    */
    AdjustWindowRect (&Rectangle, WS_POPUP, TRUE);
    SetWindowPos (hWnd, (HWND) NULL, 0, 0, Rewth, Rehg + 1, SWP_NOZORDER);
    GetClientRect(hWnd, &Rectangle);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// เพิ่มข้อมูล IFPRESEN.RC
//Microsoft App Studio generated resource script.
#include "resource.h"
#define APSTUDIO_READONLY_SYMBOLS

// Generated from the TEXTINCLUDE 2 resource.
#define APSTUDIO_HIDDEN_SYMBOLS
#include "windows.h"
#undef APSTUDIO_HIDDEN_SYMBOLS
#undef APSTUDIO_READONLY_SYMBOLS

// Bitmap
ID_BITMAP          BITMAP MOVEABLE PURE "DLGBRUSH.BMP"

// Icon
IC                 ICON DISCARDABLE "IFPRESEN.ICO"

// Menu
COMMONDLGMENU MENU DISCARDABLE
BEGIN
    POPUP "&Menu "
    BEGIN
        MENUITEM "&Enter script...",    IDM_DLGET
        MENUITEM "&Show",                IDP_SHOW
        MENUITEM SEPARATOR
        MENUITEM "E&xit",                IDM_EXIT
    END
    POPUP "&Resolution"
    BEGIN
        MENUITEM "640*480",              IDP_RES1
        MENUITEM "800*600",              IDP_RES2
        MENUITEM "1024*768",             IDP_RES3
    END
    MENUITEM "&About",                  IDM_ABOUT
END

// Dialog
DIALOGGET DIALOG DISCARDABLE 95, 32, 312, 224
STYLE DS_MODALFRAME | WS_POPUP
FONT 8, "Systemf"
BEGIN
    GROUPBOX          "", IDC_STATIC, 16, 9, 65, 31
    CONTROL            "Picture", IDP_PIC, "Button", BS_AUTORADIOBUTTON |
        WS_TABSTOP, 24, 16, 35, 10
    CONTROL            "Color", IDP_COL, "Button", BS_AUTORADIOBUTTON | WS_TABSTOP,
        24, 26, 35, 10
    EDITTEXT           IDP_FILE, 15, 42, 72, 12, ES_AUTOHSCROLL | ES_READONLY
    LISTBOX            IDP_LISTFILE, 15, 55, 62, 38, LBS_NOINTEGRALHEIGHT |
        WS_VSCROLL | WS_TABSTOP
    GROUPBOX           "Picture / Color", IDC_STATIC, 94, 11, 210, 60
    COMBOBOX           IDP_PSH, 125, 25, 65, 50, CBS_DROPDOWNLIST | CBS_AUTOHSCROLL |
        WS_VSCROLL | WS_TABSTOP
    LTEXT              "IN :", IDC_STATIC, 101, 27, 20, 8
    COMBOBOX           IDP_PEN, 225, 25, 65, 50, CBS_DROPDOWNLIST | WS_VSCROLL |
        WS_TABSTOP
    LTEXT              "OUT :", IDC_STATIC, 202, 27, 21, 8
    COMBOBOX           IDP_TMPSH, 125, 46, 30, 50, CBS_DROPDOWNLIST | WS_VSCROLL |
        WS_TABSTOP
    LTEXT              "speed", IDC_STATIC, 100, 50, 22, 8
    COMBOBOX           IDP_TMPEN, 225, 46, 30, 50, CBS_DROPDOWNLIST | WS_VSCROLL |
        WS_TABSTOP
    LTEXT              "speed", IDC_STATIC, 201, 50, 21, 8
    EDITTEXT           IDP_TEXT, 115, 81, 140, 12, ES_AUTOHSCROLL
    LTEXT              "Text:", IDC_STATIC, 89, 84, 23, 10
    PUSHBUTTON         "Font...", IDP_FONT, 264, 80, 37, 14
    GROUPBOX           "Font", IDC_STATIC, 65, 95, 237, 57
    COMBOBOX           IDP_TSH, 115, 110, 65, 50, CBS_DROPDOWNLIST | CBS_AUTOHSCROLL |
        WS_VSCROLL | WS_TABSTOP
    LTEXT              "FONT IN :", IDC_STATIC, 76, 112, 34, 8
    COMBOBOX           IDP_TMTSH, 112, 131, 30, 50, CBS_DROPDOWNLIST | WS_VSCROLL |
        WS_TABSTOP
    LTEXT              "speed", IDC_STATIC, 84, 135, 26, 8

```

เอกสารนี้เป็นเอกสารต้นฉบับสำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COMBOBOX IDP_TEN,230,110,65,50,CBS_DROPDOWNLIST | CBS_AUTOHSCROLL |
WS_VSCROLL | WS_TABSTOP
LTEXT "FONT OUT :",IDC_STATIC,187,112,39,8
COMBOBOX IDP_TMTEN,230,131,30,50,CBS_DROPDOWNLIST | WS_VSCROLL |
WS_TABSTOP
LTEXT "speed",IDC_STATIC,203,134,23,8
EDITTEXT IDP_DELAY,17,125,15,12,ES_AUTOHSCROLL
LTEXT "Delay",IDC_STATIC,16,112,24,8
LTEXT "sec.",IDC_STATIC,38,129,19,8
EDITTEXT IDP_COUNT,36,156,17,13,ES_AUTOHSCROLL | ES_READONLY
LTEXT "Usr :",IDC_STATIC,16,161,18,8
PUSHBUTTON "Edit...",IDP_EDIT,73,159,36,12
PUSHBUTTON "Insert...",IDP_INSERT,117,159,36,12
PUSHBUTTON "Del...",IDP_DEL,161,159,36,12
PUSHBUTTON "Add...",IDP_ADD,205,159,36,12
PUSHBUTTON "OK",IDP_OK,253,156,42,15
LISTBOX IDP_LIST,16,175,284,39,LBS_NOINTEGRALHEIGHT |
LBS_DISABLENOSCROLL | WS_VSCROLL | WS_TABSTOP

```

END

```

ABOUTDLG_DIALOG DISCARDABLE 75, 35, 221, 157
STYLE DS_SYSMODAL | DS_MODALFRAME | WS_POPUP
FONT 8, "MS Sans Serif"

```

BEGIN

```

DEFPUSHBUTTON "OK",IDP_OK1,89,127,50,14
ICON IC,ID_TES,105,9,18,20
LTEXT "Image and Font Presentation on microcomputer",
IDC_STATIC,39,38,161,8
LTEXT "Department of Industrial Instrumentation Technology",
IDC_STATIC,31,61,174,10
LTEXT "Faculty of Engineering",IDC_STATIC,77,81,80,8
LTEXT "King Mongkut's Institute of Technology Ladkrabang",
IDC_STATIC,30,100,170,10

```

END

```
#ifndef APSTUDIO_INVOKED
```

```
// TEXTINCLUDE
```

```
1 TEXTINCLUDE DISCARDABLE
```

```
BEGIN
```

```
"resource.h"
```

```
END
```

```
2 TEXTINCLUDE DISCARDABLE
```

```
BEGIN
```

```
"#define APSTUDIO_HIDDEN_SYMBOLS\r\n"
"#include \"windows.h\"\r\n"
"#undef APSTUDIO_HIDDEN_SYMBOLS\r\n"
"\r\n"

```

```
END
```

```
3 TEXTINCLUDE DISCARDABLE
```

```
BEGIN
```

```
"\r\n"
"\r\n"

```

```
END
```

```
#endif // APSTUDIO_INVOKED
```

```
// Cursor
```

```
IDC_CURSOR1 CURSOR DISCARDABLE "CURSOR1.CUR"
```

```
#ifndef APSTUDIO_INVOKED
```

```
#endif // not APSTUDIO_INVOKED
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// เพิ่มข้อมูล IPRESEN.H
#include "resource.h"
#define SEEK_CUR 1
#define SEEK_END 2
#define SEEK_SET 0
#define MAXREAD 32768 /* Number of bytes to be read during */
/* each read operation. */

int PASCAL WinMain(HANDLE, HANDLE, LPSTR, int);
typedef WORD (CALLBACK* FARHOOK)(HWND,UINT,WPARAM, LPARAM);

#ifndef ININIT
HANDLE ghInst;
HWND ghWnd;
HWND ghFFRDlg=0;
HWND ghPrintingDlg;
HBRUSH ghBkgndBrush=0, ghDlgBrush=0;
HFONT ghSelectedFont=0;
WORD wFRMsg;
WORD wHelpMsg;
BOOL gbMonochrome;
BOOL gbUserAbort;
char gszFilter[256];
char gszBuffer[125];
#endif

#ifdef ININIT
extern HANDLE ghInst;
extern HWND ghWnd;
extern HWND ghFFRDlg;
extern HWND ghPrintingDlg;
extern HBRUSH ghBkgndBrush, ghDlgBrush;
extern HFONT ghSelectedFont;
extern WORD wFRMsg;
extern WORD wHelpMsg;
extern BOOL gbMonochrome;
extern BOOL gbUserAbort;
extern char gszFilter[256];
extern char gszBuffer[125];
#endif
long FAR PASCAL __export MainWndProc(HWND, UINT, WPARAM,LPARAM);
int PASCAL WinMain(HANDLE, HANDLE, LPSTR, int);
BOOL FAR InitInstance(HANDLE, int);
void FAR ReportError(WORD);
BOOL FAR PASCAL __export DialogGet(HWND, UINT, WPARAM,LPARAM);
BOOL FAR PASCAL __export About1(HWND, UINT, WPARAM,LPARAM);

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//   เพิ่มข้อมูล IFPRESEN.DEF

NAME      ifpresen      ; application's module name

DESCRIPTION Image and Font Presentation

EXETYPE   WINDOWS      ; required for all Windows applications

STUB      WINSTUB.EXE ; Generates error message if application
           ; is run without Windows

CODE PRELOAD MOVEABLE DISCARDABLE

DATA PRELOAD MOVEABLE MULTIPLE

HEAPSIZE  2048

SEGMENTS
INIT_TEXT      MOVEABLE PRELOAD DISCARDABLE
IFPRESEN_TEXT  MOVEABLE LOADONCALL DISCARDABLE
ERROR_TEXT     MOVEABLE LOADONCALL DISCARDABLE
_TEXT         PRELOAD

EXPORTS
MainWndProc   @1 ; name of window processing function
DialogGet     @2 ; name of "DialogGet" processing function
About1        @3 ; name of "About1" processing function

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//      เพิ่มข้อมูล INIT.C
//      ประกอบคำอธิบายในการสร้างและแสดงวินโดวแอปพลิเคชัน
//      ฟังก์ชัน: BOOL InitInstance(HANDLE, Int)
```

```
#define WINST
#include "windows.h"
#include "commdlg.h"
#pragma hdrstop
#define ININIT
#include "ifpresen.h"
#undef ININIT
```

```
BOOL FAR InitInstance(HANDLE hInstance, Int nCmdShow)
{
    ghInst = hInstance;
    ghWnd = CreateWindow("CommonWClass",
        "Image and Font",
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT,
        CW_USEDEFAULT, //nCmdShow,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        NULL,
        NULL,
        hInstance,
        NULL);

    if (!ghWnd)
        return(FALSE);
    ShowWindow(ghWnd, SW_SHOWMAXIMIZED); //nCmdShow);
    UpdateWindow(ghWnd);
    return(TRUE);
}
```

```
//      เพิ่มข้อมูล ERROR.C
//      เป็นส่วนแสดงความผิดพลาดของส่วนต่างๆ
//      ฟังก์ชัน: void FAR ReportError(WORD)
```

```
#define WINST
#include "windows.h"
#include "commdlg.h"
#pragma hdrstop
#include "cderr.h"
#include "ifpresen.h"
```

```
void FAR ReportError(WORD wErrorType)
{
    LPSTR lpszErrorMsg;
    switch( wErrorType )
    {
        case IDC_ALLOCFAIL:
            lpszErrorMsg="Error Allocating Memory!";
            break;
        case IDC_LOCKFAIL:
            lpszErrorMsg="Error Locking Memory!";
            break;
        case IDC_LOADSTRINGFAIL:
            lpszErrorMsg="LoadString failed!";
            break;
        default: //let's hope we never get here!
            return;
    }
    MessageBox(ghWnd, (LPSTR)lpszErrorMsg, "Image and Font", MB_OK);
    return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ทรงชัย วีระวิมาศ และอาจารย์อรัญญา ปรีชาไว ที่ให้คำปรึกษาแนวทางในการแก้ปัญหา และให้ความอนุเคราะห์เครื่องไมโครคอมพิวเตอร์ ตลอดจนหนังสือเพื่อใช้ในการทำโครงการงานนี้ และขอขอบคุณเพื่อนๆทุกคนที่ให้การสนับสนุน ช่วยเหลือในการทำโครงการงานจนสำเร็จลุล่วงด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. จิรพัฒน์ จันทร์เจดศักดิ์ และ วีระ นพนิราพาธ , การเขียนโปรแกรมบน Microsoft Windows , กรุงเทพฯ , 2537
2. Microsoft Coporation , Microsoft Windows3.1 Guide to Programming , 1992
3. James W.McCord , Windows3.1 Programming Reference , 1992



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้