

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การแปลงรูปภาพเป็นแบบภาพปักครอสติช

1. นาย ชวลิต ฉันทชัยศิริเวชย์
2. นาย เศชา บุญญะโรดล

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539



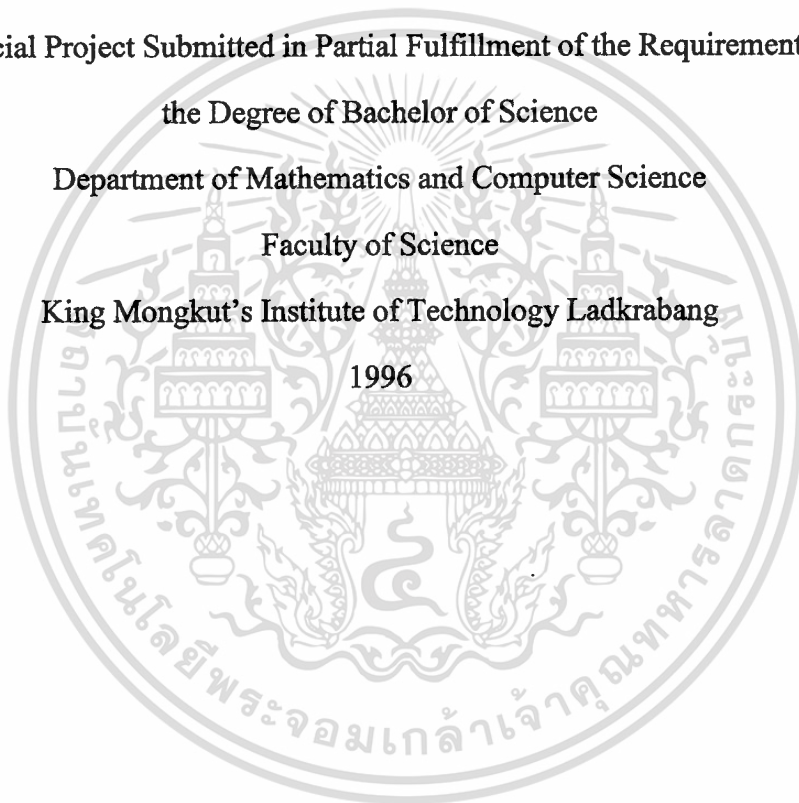
เลขหม.....
เลขทะเบียน..... 33885
วัน, เดือน, ปี 17 ก.ย. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Transformation Picture to Cross-stitch Design

1. Mr. Chavalit Chanchaisrivet
2. Mr. Decha Boonyarodhol

**A Special Project Submitted in Partial Fulfillment of the Requirement for
the Degree of Bachelor of Science
Department of Mathematics and Computer Science
Faculty of Science
King Mongkut's Institute of Technology Ladkrabang**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ การแปลงรูปภาพเป็นแบบภาพปีกรอสติช

โดย นายชวลิต นันชัยศิริเวทย์

นายเดชา บุญญะโรคล

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา คร. นันทิกา เบญจเทพานันท์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยี
พระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำโครงการนี้เป็นส่วนหนึ่งของการ
ศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต



(รศ. รัชคินี ชิตสกุล)

หัวหน้าภาควิชาฯ



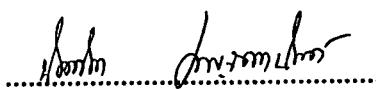
(รศ. อุบลวรรณ เงินวิจิตร)

ประธานกรรมการการสอบปัญหาพิเศษ



(รศ. พ้องพรรณ รัตนธนาวันต์)

กรรมการสอบปัญหาพิเศษ



(คร. นันทิกา เบญจเทพานันท์)

กรรมการและอาจารย์ที่ปรึกษาปัญหาพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ทุกวันนี้ ความเจริญก้าวหน้าทางเทคโนโลยีของวงการคอมพิวเตอร์ ได้เข้ามามีบทบาทในหลายสาขาวิชาชีพ ไม่ว่าจะเป็นในแง่ธุรกิจ, วิศวกรรม, วิทยาศาสตร์, อุตสาหกรรม, เกษตรกรรม หรือแม้แต่ในงานหัตถกรรม ปัญหาพิเศษนี้เป็นการนำเสนอโปรแกรมสำเร็จรูปเพื่อช่วยในการแปลงรูปภาพที่ต้องการออกมาเป็นแบบภาพปักครอสติช โดยใช้เทคนิคทางด้านคอมพิวเตอร์กราฟฟิก เพื่อให้ได้แบบภาพปักที่ถูกต้อง ซึ่งเราไม่สามารถหาซื้อได้ตามท้องตลาดทั่วไป และยังช่วยประหยัดเงินในการซื้อแบบภาพปักครอสติชที่มีราคาแพงได้

ปัญหาพิเศษนี้ได้เขียนโดยใช้ Microsoft Visual C++



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abstract

Nowaday microcomputer plays an important role in many field such as business, engineering, science, industrialization, agriculturing, and handiwork. This project is an application which transforms a picture to cross-stitch design by using computer graphics so that takes the satisfied cross-stitch design and saves a payment for cross-stitch design that is expensive.

The project is written on Microsoft Visual C++



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

CONTENT

หน้าอำนวยการ

บทคัดย่อปัญหาพิเศษภาษาไทย

บทคัดย่อปัญหาพิเศษภาษาอังกฤษ

สารบัญ	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	2
- รูปแบบของบิตแมป	2
- อัลกอริทึมต่าง ๆ ของโปรแกรม	3
- อัลกอริทึมโดยรวมของโปรแกรม	3
- อัลกอริทึมการเปรียบเทียบสีจากรูปภาพกับฐานข้อมูลของไหม	5
บทที่ 3 การวิจัยและการดำเนินการ	7
- การออกแบบ	7
- การพัฒนาและการปรับปรุงโปรแกรม	7
- คู่มือการใช้โปรแกรม	8
- เมนูไฟล์	8
- เมนูมุมมอง	11
- เมนูรูปแบบ	12
- เมนูผ้า	16
- เมนูไหม	18
- เมนูวิธีใช้	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ผลการวิจัย	20
- ความสะดวกในการใช้งานของผู้ใช้	20
- ความสามารถในการเลือกสายผ้า, สัญลักษณ์	20
- ความสามารถในการเพิ่ม, ลบข้อมูลของใหม่	20
- ความสามารถในการดูตัวอย่างก่อนพิมพ์	21
- ผลที่ได้จากการทำงานของโปรแกรม	27
บทที่ 5 สรุปผลและเสนอแนะ	40
- สรุปผล	40
- ข้อเสนอแนะ	40
- การจัดเก็บไฟล์	40
- การแก้ไขแบบภาพปักซึ่งได้รับการแปลงมาจากรูปภาพแล้ว	40
ภาคผนวก	41
- ฐานข้อมูลของใหม่	
- Source Code ของ โปรแกรม XStitch	
- Source Code ของ โปรแกรม MakeFloss	
บรรณานุกรม	

บทที่ 1

บทนำ

ความสำคัญและที่มาของปัญหาพิเศษ

เนื่องจากในปัจจุบันแบบภาพของภาพปักครอสติสนั้นเราไม่สามารถทำได้ จำเป็นต้องซื้อมาด้วยราคาที่ค่อนข้างแพง หรือจะหาแบบตามนิตยสารต่างๆ แต่ก็จะมีแบบที่ดูใจได้ยาก ฉะนั้นถ้าเรามีภาพที่ต้องการแล้วอยากทำเป็นภาพปักครอสติสใส่กรอบเอาไว้โชว์ เราก็สามารถทำได้โดยใช้คอมพิวเตอร์เข้ามาช่วย คือใช้เทคนิคทางด้านคอมพิวเตอร์กราฟฟิก นำรูปภาพที่ต้องการมาแปลงเป็นแบบภาพปักครอสติส

วัตถุประสงค์ของปัญหาพิเศษ

- ให้ได้แบบภาพปักที่ดูใจ
- ประหยัดเงินในการซื้อแบบภาพปักครอสติส

ขอบเขตของปัญหาพิเศษ

การแปลงภาพปักครอสติสนั้นเราจะจำกัดจำนวนสีไว้ที่ 256 สี และเราไม่สามารถทำเส้นตัดภาพหรือเส้นต่างๆ ของภาพได้ ที่ไม่ใช่เส้นที่ปักเป็นรูปกากบาท เราจำเป็นที่จะต้องมีการปักที่ปักครอสติสอยู่แล้วซึ่งอาจจะได้มาจากการสแกนเนอร์ หรือทำขึ้นมาเองการโปรแกรมกราฟฟิกต่างๆ แล้วนำภาพนั้นๆมาแปลงเป็นภาพปักโดยโปรแกรม ก็จะได้แบบภาพปักที่ต้องการและเบอร์สีต่างๆ ของด้ายที่ใช้

ประโยชน์ที่คาดว่าจะได้รับ

ช่วยให้ได้แบบภาพปักที่ดูใจ ลดเวลาในการหาแบบภาพปัก และยังเป็นการประหยัดเงินอีกด้วย

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

รูปแบบของบิตแมป

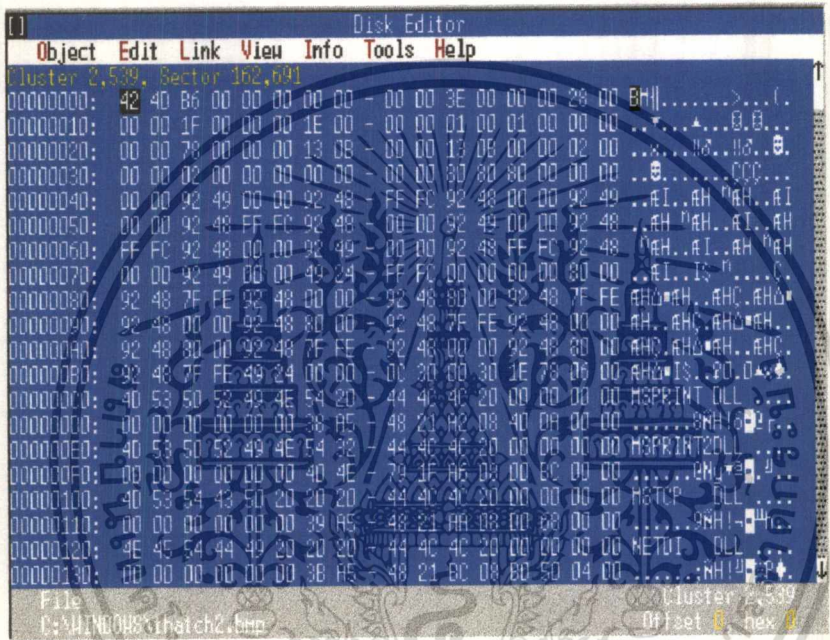
ส่วนแรกอยู่ในคอนตันของไฟล์ เรียกว่า File Header ซึ่งจะต้องมีในไฟล์ภาพทุกชนิด ในแต่ละส่วนจะเป็นส่วนสำคัญที่บอกลักษณะของภาพ เพื่อนำไปคำนวณในการวาดภาพ และการเรียงลำดับของสี (Palette) ในภาพนั้นๆ ถ้าข้อมูลในส่วนนี้เกิดผิดพลาดจะทำให้สีของภาพเปลี่ยนไปจนบางที่ทำให้เราดูภาพไม่รู้เรื่องได้ ข้อมูลเหล่านี้จะนำมาเก็บไว้ในรูปแบบของ struct ซึ่งสามารถแสดงให้ดูได้ดังนี้

struct BMPHEAD	
{ WORD bfType;	ใช้เก็บ code ของ BMP ซึ่งต้องมีค่า = BM เท่านั้น
DWORD bfSize;	เก็บขนาดของไฟล์
WORD bfReserved1;	สงวนไว้ (ไม่ใช้งาน)
WORD bfReserved;	สงวนไว้ (ไม่ใช้งาน)
DWORD bfOffBits;	ขนาดของ Header
DWORD biSize;	ขนาดของ Information
LONG width;	ขนาดความกว้างของภาพ
LONG depth;	ขนาดความสูงของภาพ
WORD biplanes;	จำนวนหน้าของภาพ
WORD biBitCount;	จำนวน bit ที่เก็บสี
DWORD biCompression;	ค่าการบีบอัดไฟล์
DWORD biSizeImage;	ขนาดของรูป
LONG biXPelsPerMeter;	จำนวนพิกเซลต่อเมตรในแนวแกน x
LONG biYPelsPerMeter;	จำนวนพิกเซลต่อเมตรในแนวแกน y
DWORD biClrUsed;	จำนวนของสีที่ใช้ในรูปภาพ
DWORD biClrImportant;	จำนวนของสีที่มีความสำคัญต่อรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่สองอยู่ต่อจากส่วน Header จนถึงท้ายไฟล์ ในส่วนนี้จะเป็นส่วนข้อมูลของภาพจริงๆ ซึ่ง 4 บิต ทางซ้ายของไบต์แรกของส่วนนี้จะเก็บจุดมุมล่างซ้าย ส่วน 4 บิต ทางขวาของไบต์สุดท้ายของไฟล์จะเก็บจุดมุมบนขวาของภาพ ที่เป็นเช่นนี้เพราะ 1 จุดสีจะเก็บด้วยข้อมูลขนาด 4 บิต (เพราะใช้ 16 สี) ฉะนั้น 1 ไบต์จึงเก็บข้อมูลได้ 2 จุด

ตัวอย่างของการเก็บแบบบิตแมปของรูป  (Thatch2.bmp) แสดงได้ดังรูปข้างล่างนี้

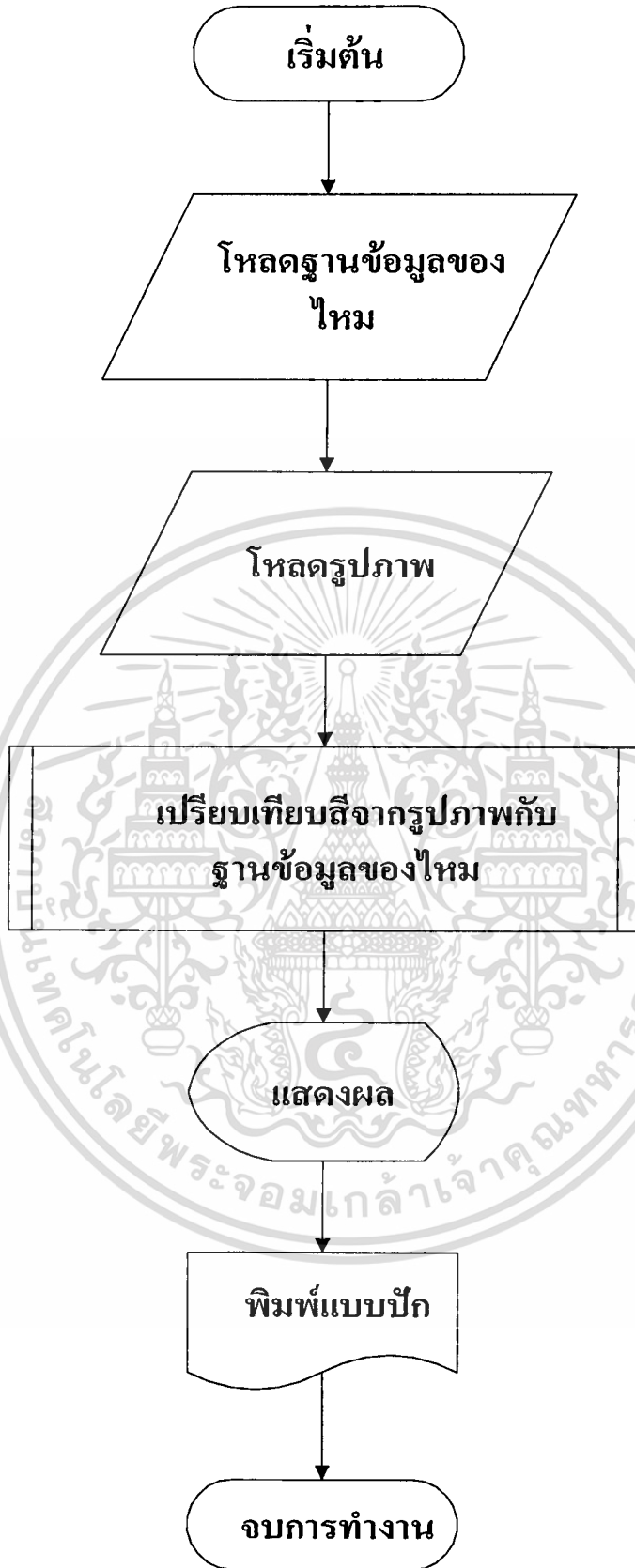


อัลกอริทึมต่างๆ ของโปรแกรม

1. อัลกอริทึมโดยรวมของโปรแกรม

เมื่อเริ่มโปรแกรม โปรแกรมจะทำการโหลดข้อมูลของใหม่ซึ่งประกอบด้วย เบอร์ใหม่, ชื่อ, ค่าของสีแดง, สีเขียว และสีน้ำเงิน ตามลำดับจากฐานข้อมูลซึ่งเก็บไว้ในไฟล์ชื่อ Dmc.mai จากนั้นผู้ใช้จะต้องเลือกรูปที่ต้องการแปลงซึ่งจะต้องเป็นชนิดบิตแมปเท่านั้น ขั้นตอนต่อไป โปรแกรมจะทำการเปรียบเทียบสีของรูปภาพกับสีของใหม่ที่มีอยู่ในฐานข้อมูลแล้วจึงแสดงผลออกมาในรูปแบบภาพปีกครอสติช ซึ่งสามารถพิมพ์ออกมาเป็นเอกสารได้ อัลกอริทึมนี้

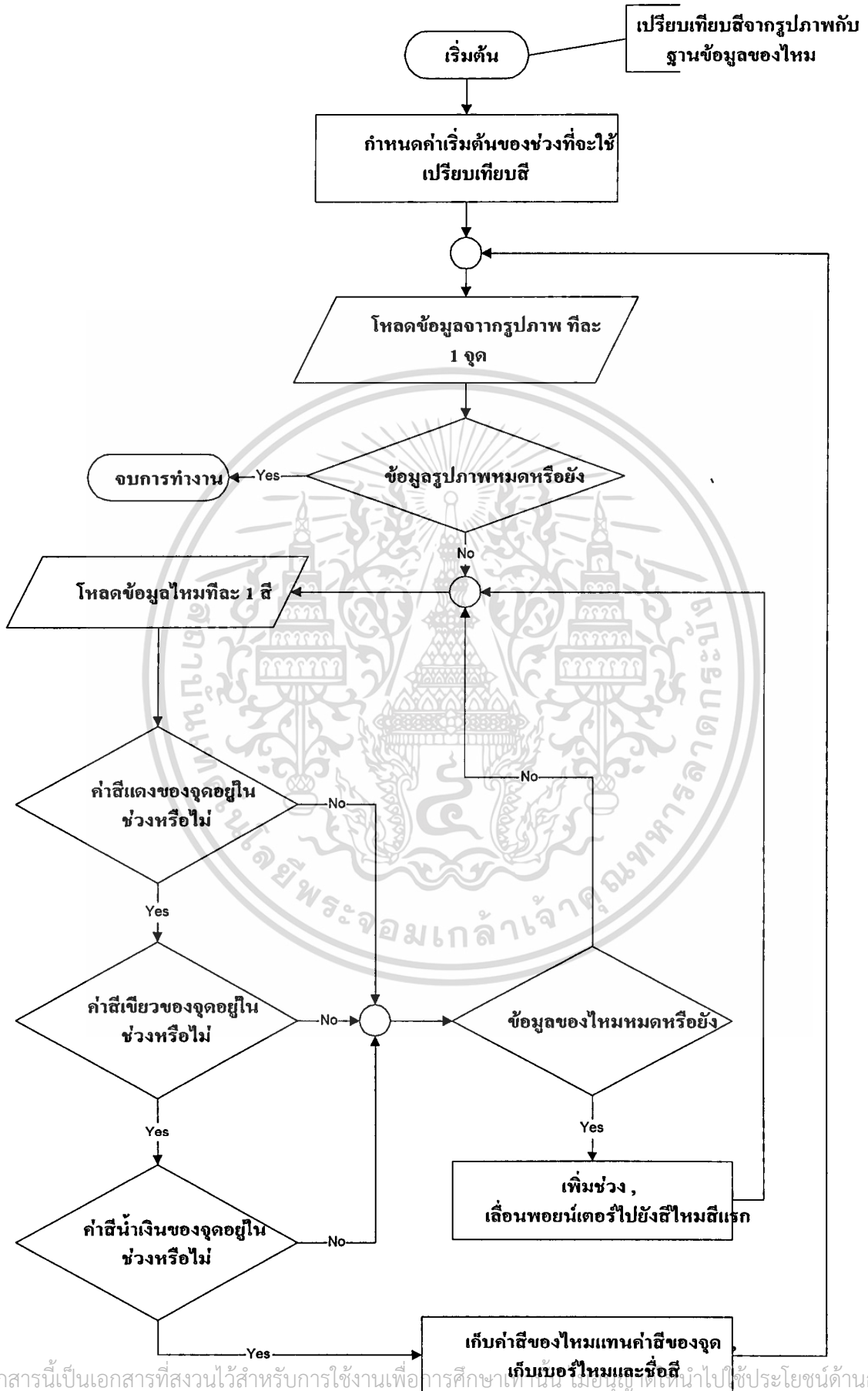
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อัลกอริทึมการเปรียบเทียบสีจากรูปภาพกับฐานข้อมูลของไทย

เราจะทำโดยการ โหลดข้อมูลจากรูปภาพของจุดที่ละจุดแล้วนำมาเปรียบเทียบกับฐานข้อมูลที่อยู่ในหน่วยความจำโดยแยกเปรียบเทียบตามสีแดง สีเขียว และสีน้ำเงิน (R G B) ตามลำดับเพื่อให้ได้สีใหม่ที่ใกล้เคียงกับสีจากรูปจริงมากที่สุด กล่าวคือโปรแกรมจะกำหนดช่วงของค่าสีที่ยอมรับได้ (ในโปรแกรมนี้กำหนดค่าช่วงเริ่มต้นเท่ากับบวกลบ 8) จากนั้นโปรแกรมก็จะเก็บค่าสีแดง, สีเขียว และสีน้ำเงินจากจุดของรูปภาพไว้ในตัวแปร แล้วนำไปเทียบกับสีใหม่ที่มีอยู่ที่ละสีจนกว่าจะพบสีที่ใกล้เคียง แต่ถ้าในกรณีที่ไม่พบสีที่ใกล้เคียงในช่วงบวกลบ 8 โปรแกรมก็จะทำการเพิ่มค่าของช่วงที่ยอมรับได้ให้กว้างขึ้น (ในโปรแกรมนี้กำหนดให้ช่วงเพิ่มขึ้นทีละบวกลบ 4) แล้วทำการเปรียบเทียบใหม่ที่ละสีตั้งแต่เบอร์ใหม่เบอร์แรก ทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะพบสีที่ใกล้เคียงของแต่ละจุดจนครบทุกจุดในรูป เราก็จะได้สีของแบบปกที่เป็นสีใหม่ที่มีอยู่จริงในท้องตลาด ซึ่งสามารถเขียนเป็น Flowchart ได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิใช่ข้อมูลให้ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิจัยและการดำเนินการ

การออกแบบ

ในขั้นแรกของการออกแบบ ได้กำหนดลักษณะและความสามารถของโครงการไว้ดังนี้

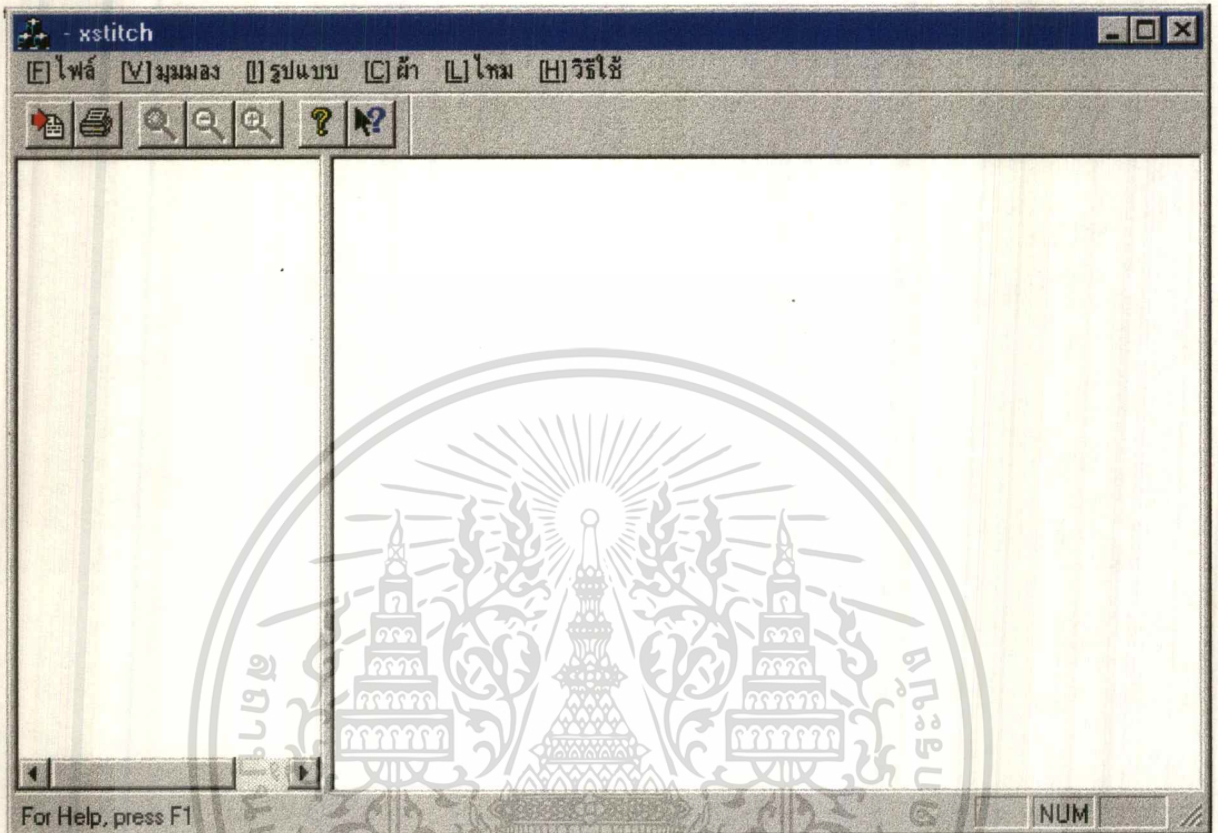
- การนำเสนอผลงานที่เป็นกราฟิกทั้งหมดในลักษณะของ GUI (Graphical User Interface)
- โหมดกราฟฟิกที่ใช้มีข้อจำกัดไม่เกิน 256 สี
- ภาพที่นำมาใช้ใน โปรแกรมเป็นภาพที่เป็นรูปแบบของบิตแมป (.BMP)
- สามารถกำหนดหลายผ้าและสัญลักษณ์ที่ใช้แสดงได้
- สามารถดูตัวอย่างก่อนพิมพ์ได้
- นอกจากจะต้องพิมพ์แบบภาพปริศนาคอสดิซออกมาแล้ว ยังต้องสามารถพิมพ์เบอร์ไหม, สีไหม, สัญลักษณ์ที่ใช้แทน, ชนิดของผ้าและขนาดของผ้าออกมาได้

การพัฒนาและการปรับปรุงโปรแกรม

เมื่อได้ทำการออกแบบส่วนประกอบต่างๆ ทั้งหน้าจอ, หน้าการทำงาน รวมทั้งข้อจำกัดต่างๆ ของโปรแกรมแล้ว จึงเริ่มทำการเขียน โปรแกรมให้เป็นไปตามที่กำหนดไว้ หลังจากนั้นจึงทำการตรวจสอบเพื่อหาข้อบกพร่อง แล้วทำการแก้ไขปรับปรุง และเพิ่มเติมเพื่อความสมบูรณ์และถูกต้องของโปรแกรม

คู่มือการใช้โปรแกรม

เมื่อเริ่มรันโปรแกรม จะมีอินเตอร์เฟซดังนี้



เมนูไฟล์ คำสั่งต่าง ๆ ที่มีในเมนูไฟล์ จะเป็นคำสั่งที่เกี่ยวกับการจัดการไฟล์ เช่น การเปิดไฟล์, การพิมพ์แบบปัก, การดูตัวอย่างก่อนพิมพ์, การตั้งค่าเครื่องพิมพ์ เป็นต้น คำสั่งในเมนูไฟล์ประกอบด้วย

เปิดรูป : ทำหน้าที่โหลดไฟล์ที่มีอยู่จริงบนดิสก์ขึ้นมาในโปรแกรม

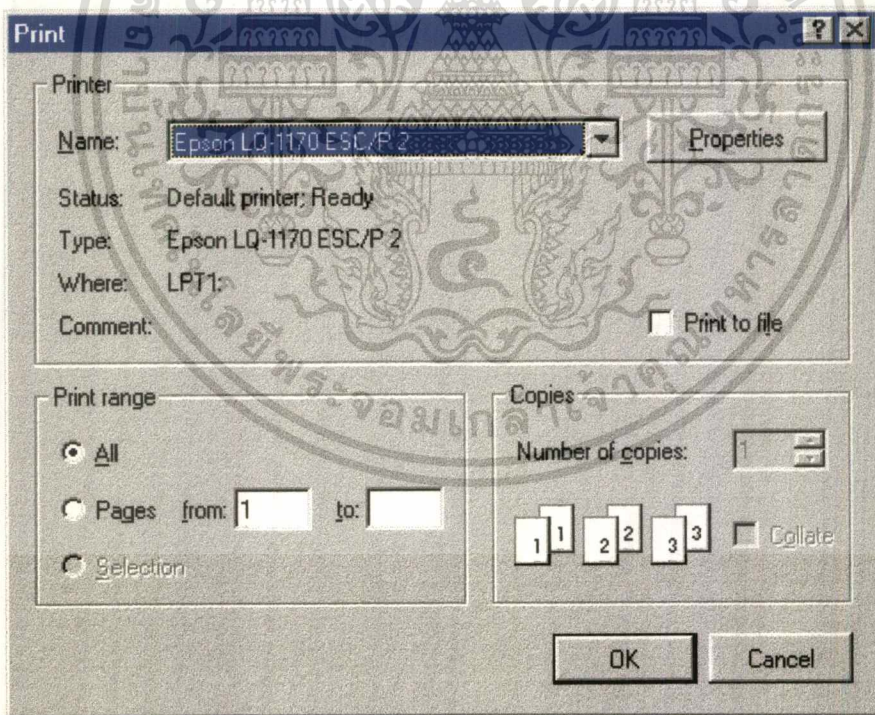
มีไอคอนคือ  เมื่อเลือกคำสั่งเปิดรูปแล้วโปรแกรมจะแสดงไดอะล็อกบ็อกซ์ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



พิมพ์ : คำสั่งพิมพ์แบบภาพปึกครอสติชออกทางเครื่องพิมพ์ มีไอคอน

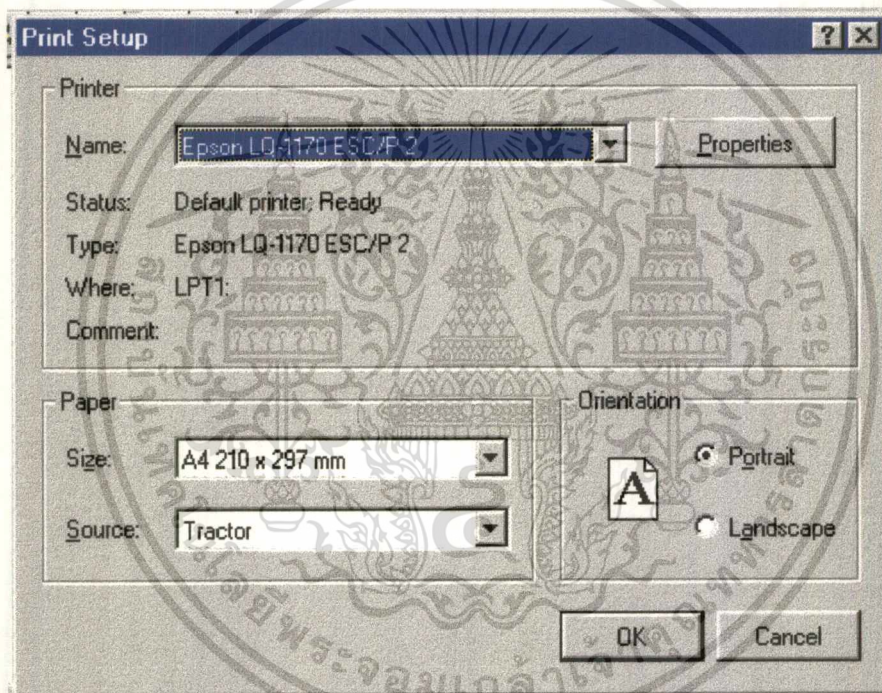
คือ  เมื่อเลือกคำสั่งพิมพ์แล้ว โปรแกรมจะแสดง
ไดอะล็อกบ็อกซ์ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างก่อนพิมพ์ : คำสั่งนี้จะแสดงตัวอย่างแบบภาพปักก่อนที่จะพิมพ์ออกทางเครื่องพิมพ์ เพื่อให้ผู้ใช้สามารถตรวจสอบรายละเอียดของแบบภาพปักก่อนได้

ตั้งค่าเครื่องพิมพ์ : แสดงหน้าต่างสำหรับเลือกค่าต่าง ๆ ของเครื่องพิมพ์ ซึ่งเหมือนกับหน้าต่าง Print Setup ทั่วไป เมื่อเลือกคำสั่งตั้งค่าเครื่องพิมพ์แล้ว โปรแกรมจะแสดงไดอะล็อกบ็อกซ์ดังรูป



Recent File List : จะแสดง path ของไฟล์ 4 ไฟล์ล่าสุดที่โปรแกรมเคยเปิด

จบการทำงาน : ผู้ใช้สามารถสิ้นสุดการทำงานของโปรแกรมได้โดยการใช้คำสั่งนี้

เมนูมุมมอง เมนูนี้ประกอบไปด้วยคำสั่งสำหรับการแสดงผลทางจอภาพ เช่น การย่อภาพ, ขยายภาพ, การแสดงแถบเครื่องมือ (toolbar), แถบสี (colorbar), แถบสถานะ (statusbar) เป็นต้น คำสั่งในเมนูมุมมองประกอบด้วย

แถบเครื่องมือ : เป็นคำสั่งแบบ toggle ทำหน้าที่แสดงแถบเครื่องมือ (แถบเครื่องมือ คือ วินโดว์ที่ประกอบด้วยปุ่มรูปภาพ ฟังก์ชันที่ถูกจัดเรียงในแนวระดับซึ่งอาจจะถูกจัดเป็นกลุ่มๆ ตามที่ออกแบบสำหรับการอินเทอร์เฟซ รูปภาพกราฟ ฟังก์ชันที่ถูกเก็บไว้ในบิตแมปซึ่งเชื่อมต่อกับรีซอร์สไฟล์ ของแอปพลิเคชัน เมื่อปุ่มถูกคลิกก็จะมีคำสั่งส่งเมสเสจ command เช่นเดียวกับเมนูและคีย์บอร์ด) แถบเครื่องมือใน โปรแกรมนี้มีลักษณะดังรูป



แถบสี : เป็นคำสั่งแบบ toggle ทำหน้าที่แสดงแถบสี

แถบสถานะ : เป็นคำสั่งแบบ toggle ทำหน้าที่แสดงแถบสถานะ ซึ่งมีหน้าที่แสดงข้อความในเฟรมภายใต้การควบคุมของ โปรแกรม

ธรรมดา : ทำหน้าที่ให้โปรแกรมแสดงผลออกทางจอภาพ โดยมีขนาด เป็น 16 เท่าของขนาดของรูปภาพที่เปิดขึ้นมา มีไอคอนดัง



ย่อรูป : ทำหน้าที่ให้โปรแกรมแสดงผลออกทางจอภาพโดยย่อขนาด

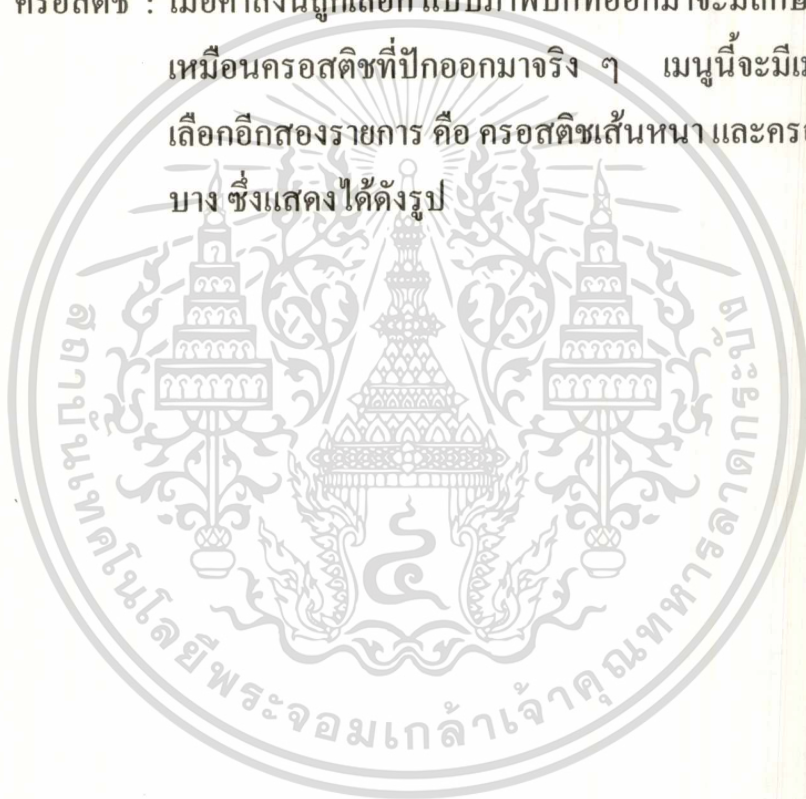
ของรูปภาพที่เปิดลงที่ละ 2 เท่า มีไอคอนดังนี้ 

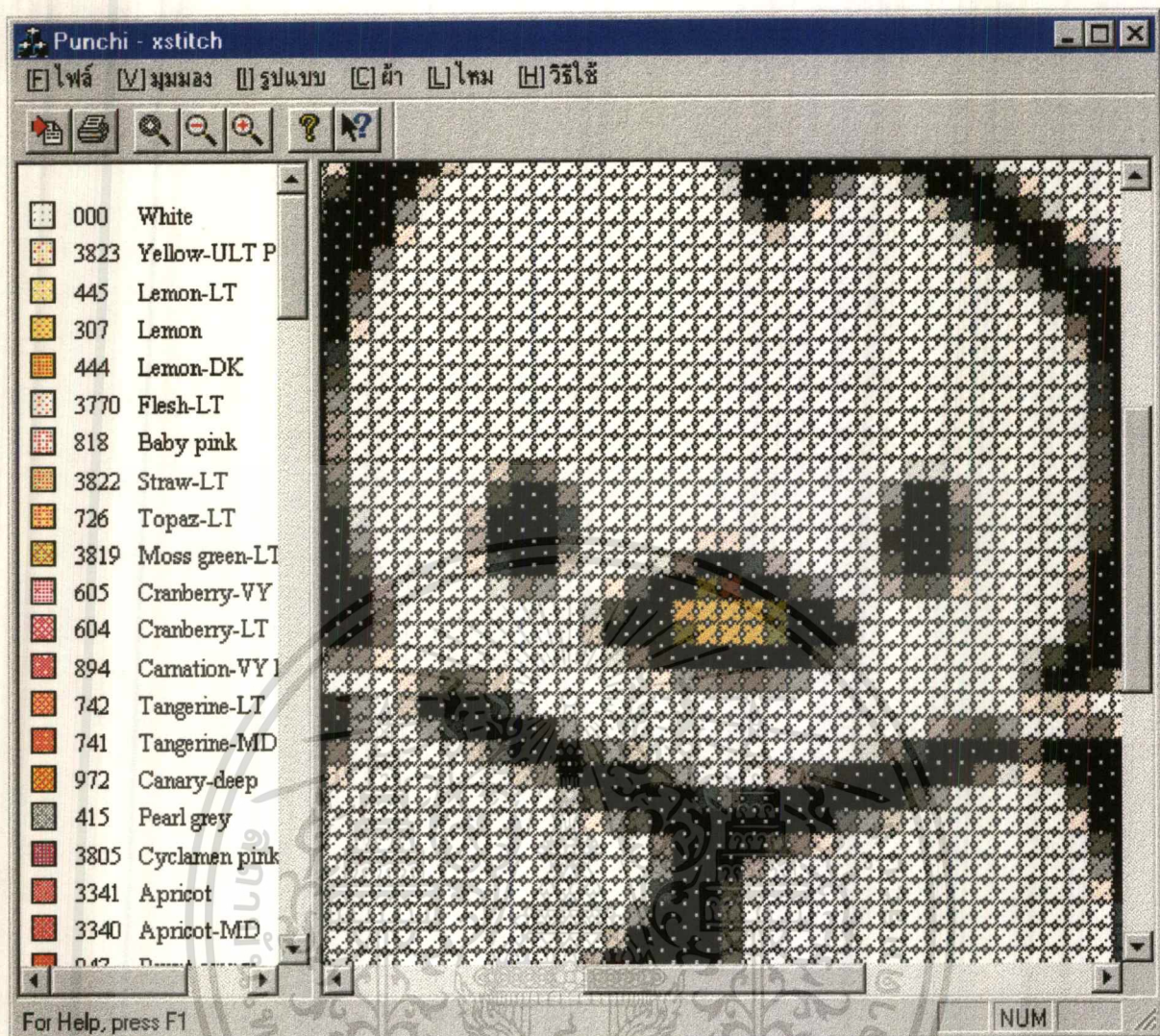
ขยายรูป : ทำหน้าที่ให้โปรแกรมแสดงผลออกทางจอภาพโดยขยาย
ขนาดของรูปภาพที่เปิดขึ้นทีละ 2 เท่า มีไอคอนดังนี้



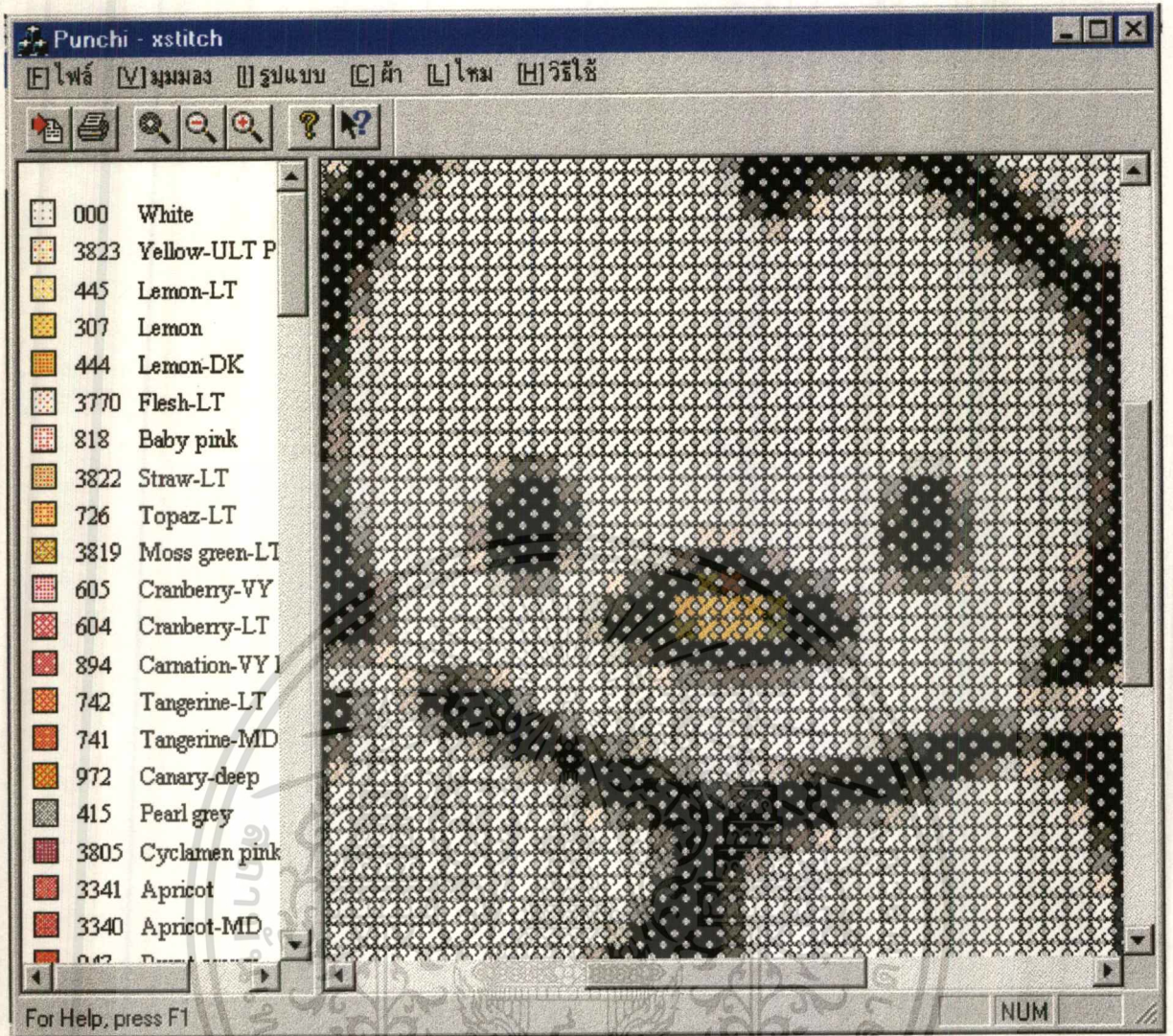
เมนูรูปแบบ เป็นเมนูคำสั่งในการแสดงสัญลักษณ์ของแบบปิก ซึ่งประกอบด้วย
ครอสติช, สัญลักษณ์ และระบายทึบ ดังนี้

ครอสติช : เมื่อคำสั่งนี้ถูกเลือก แบบภาพปิกที่ออกมาจะมีลักษณะ
เหมือนครอสติชที่ปิกออกมาจริง ๆ เมนูนี้จะมีเมนูย่อยให้
เลือกอีกสองรายการ คือ ครอสติชเส้นหนา และครอสติชเส้น
บาง ซึ่งแสดงได้ดังรูป



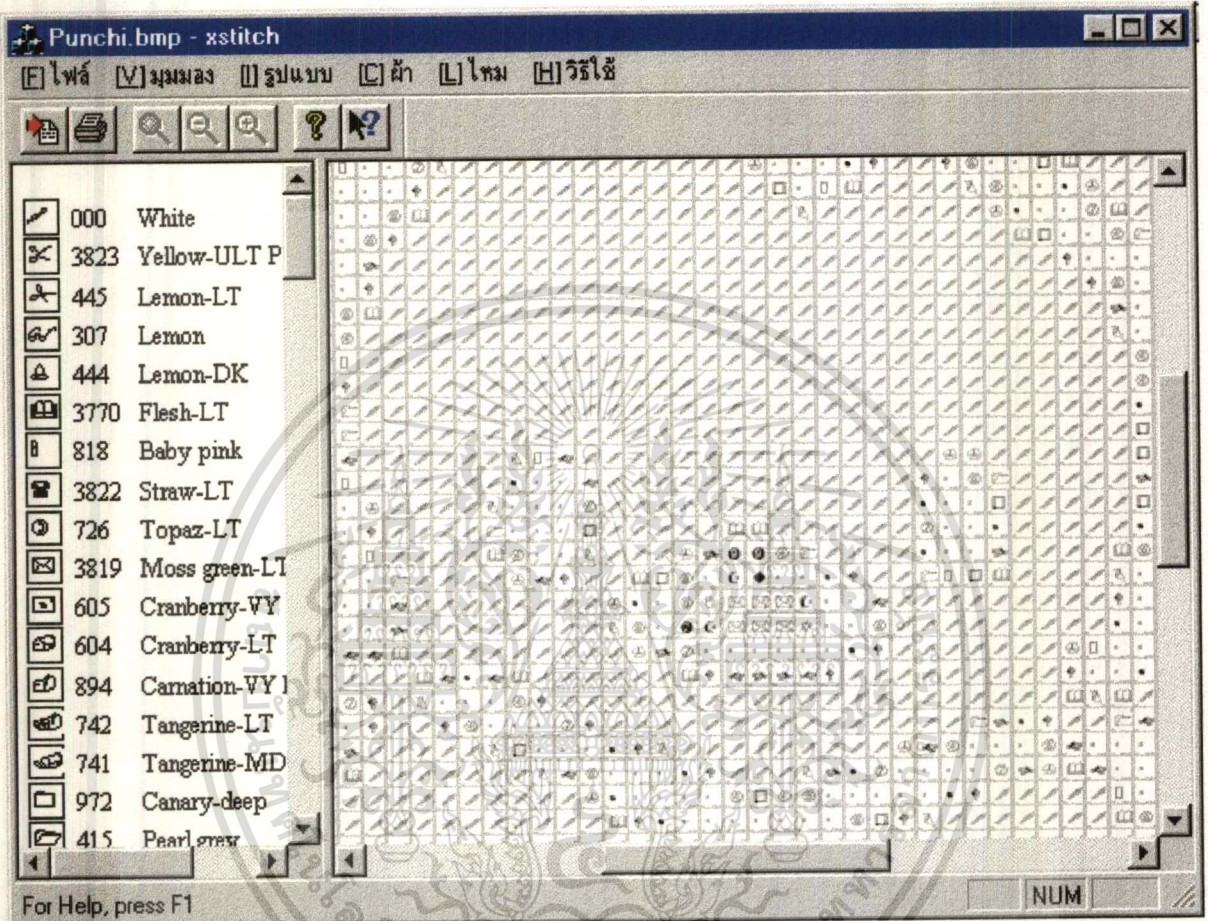


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



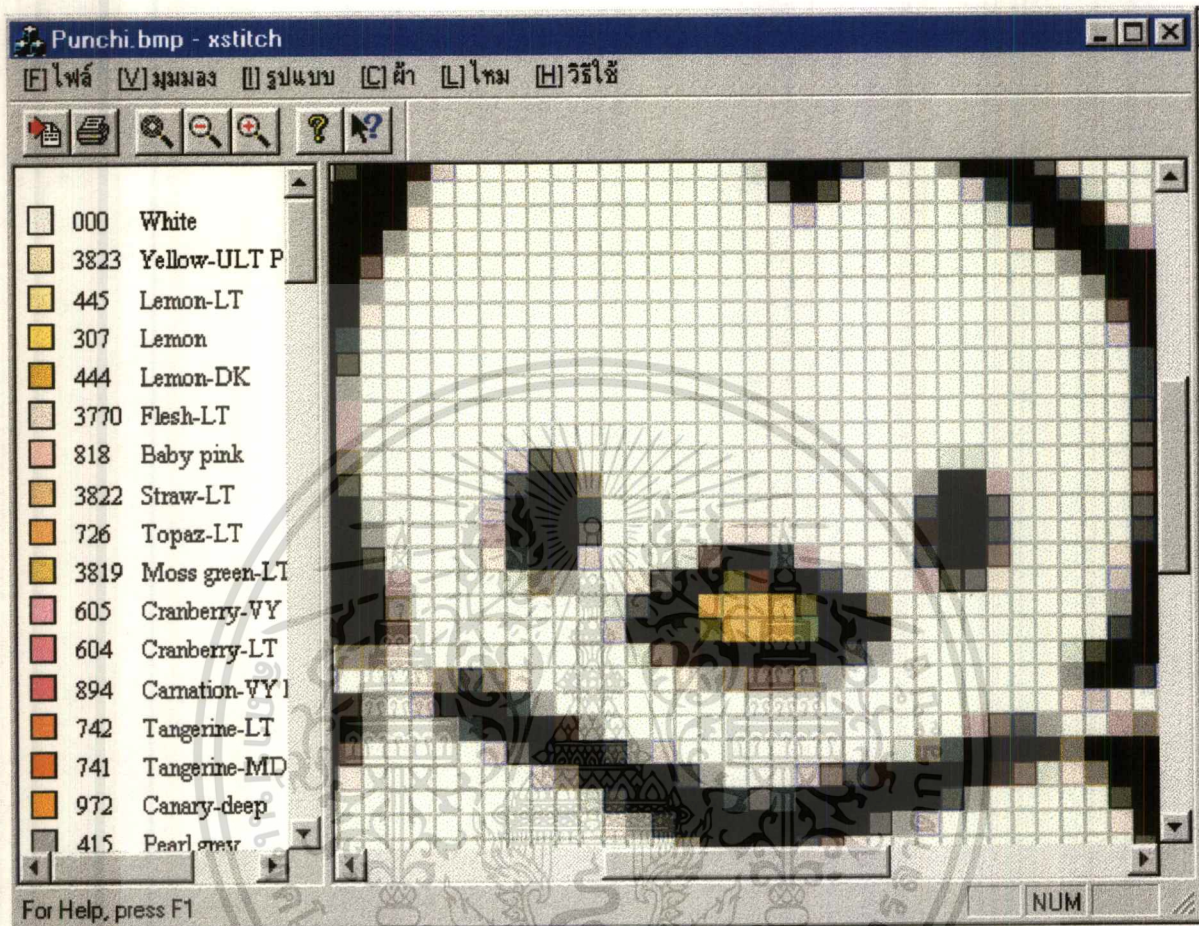
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์ : เมื่อคำสั่งนี้ถูกเลือก แบบภาพปักที่ออกมาจะมีลักษณะ
เป็นสัญลักษณ์ต่าง ๆ ซึ่งสัญลักษณ์หนึ่งสัญลักษณ์จะ
แทนสีของไหมหนึ่งสี ซึ่งแสดงได้ดังรูป



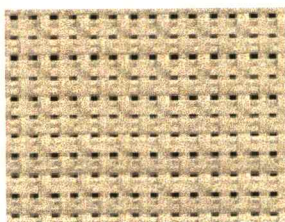
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบายทึบ : เมื่อคำสั่งนี้ถูกเลือก แบบภาพปักที่ออกมาจะมีลักษณะ
เหมือนรูปภาพที่โหลดขึ้นมาจริง ๆ ซึ่งแสดงได้ดังรูป



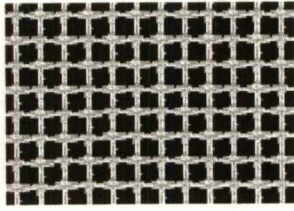
เมนูผ้า เป็นเมนูคำสั่งที่ให้ผู้ใช้งานสามารถเลือกสายผ้าได้ ซึ่งประกอบด้วย

Aida :

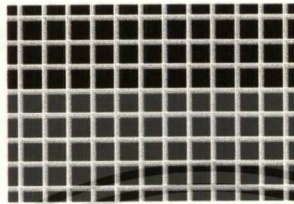


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

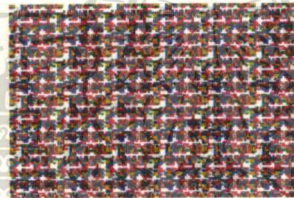
Tapestry :



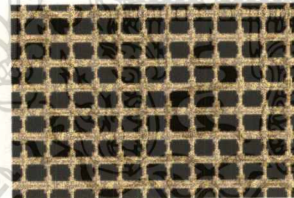
Plastic :



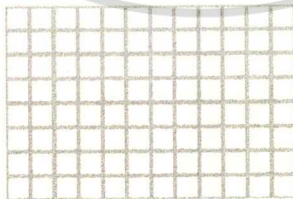
Everweave :



Rug :



เส้น Grid :

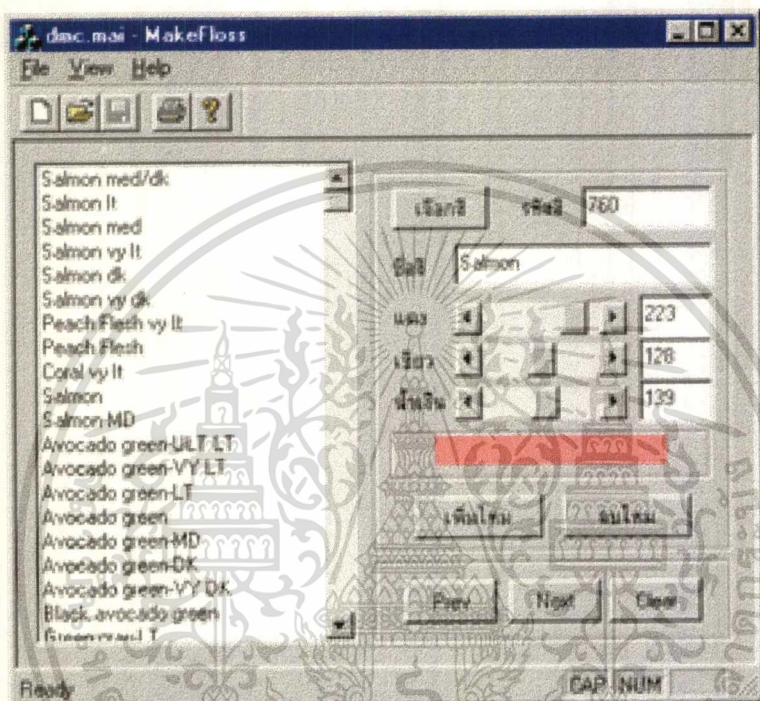


ไม่แสดงพื้นหลัง : เมื่อคำสั่งนี้ถูกเลือก โปรแกรมจะไม่แสดงพื้นหลังให้เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

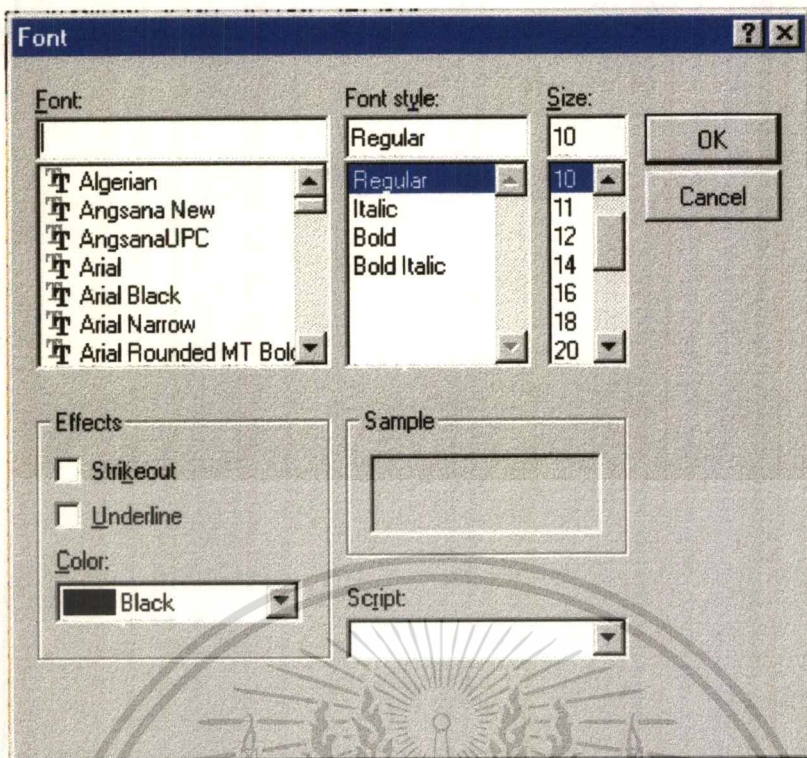
เมนูใหม่ เป็นเมนูคำสั่งที่ให้ผู้ใช้งานสามารถเลือกชนิดของใหม่ได้ ซึ่งประกอบด้วย

ใหม่ DMC : เมื่อเลือกคำสั่งนี้ จะปรากฏไอคอนบอกรหัสดังรูป ซึ่งจะ
เป็นฐานข้อมูลของใหม่ DMC ผู้ใช้จะสามารถเพิ่มเติม
หรือลบข้อมูลของใหม่ได้



รูปแบบสัญลักษณ์ : คำสั่งนี้มีไว้ให้ผู้ใช้งานสามารถเปลี่ยนรูปแบบ
สัญลักษณ์ที่ใช้ในคำสั่งสัญลักษณ์ในเมนูรูป
แบบได้ โดยจะปรากฏไอคอนบอกรหัสดังรูป
ซึ่งจะเป็นฟอนต์ต่าง ๆ ที่มีอยู่ในเครื่องฟอนต์
ปกติที่โปรแกรมตั้งไว้ คือ Wingdings

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

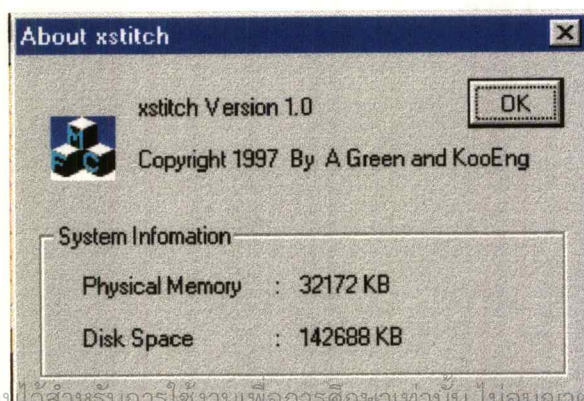


เมนูวิธีใช้ เมนูนี้ประกอบด้วยเมนูหัวข้อวิธีใช้โปรแกรมครอสติช และ เมนูเกี่ยวกับโปรแกรมครอสติช ซึ่งอธิบายได้ดังนี้

หัวข้อวิธีใช้โปรแกรมครอสติช : จะเป็นคู่มือการใช้โปรแกรม

เกี่ยวกับ โปรแกรมครอสติช : จะแสดงไคอะลือกบออกซ์ที่มีข้อมูลเกี่ยวกับเวอร์ชัน และรายงานด้านข้อมูล

ระบบให้กับผู้ใช้ ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัย

ผลการวิจัย

ผลที่ได้จากการพัฒนาโปรแกรมสามารถประเมินผลได้ดังนี้

4.1 ความสะดวกในการใช้งานของผู้ใช้

เนื่องจากโปรแกรมมีส่วนติดต่อกับผู้ใช้ (interface) แบบ GUI และเป็นภาษาไทย จึงทำให้สะดวกต่อการใช้งาน นอกจากนี้ยังมีทูลบาร์และสเตตัสบาร์เพื่อช่วยอธิบายการทำงาน ซึ่งสเตตัสบาร์นี้จะอยู่ส่วนล่างของวินโดว์ และโปรแกรมนี้อีกมีคีย์ค่วนอีกด้วย

4.2 ความสามารถในการเลือกฉายผ้า, ชนิดของสัญลักษณ์

โปรแกรมนี้อีกทำขึ้นนี้ให้ผู้ใช้สามารถเลือกฉายผ้าซึ่งเป็นพื้นหลังของแบบภาพปักครอสติชได้ เพื่อใช้เทียบดูว่ารูปภาพนี้ควรจะนำมาปักบนลายผ้าแบบใด หรือควรจะปักบนแผ่นเฟรม นอกจากนี้ยังสามารถเลือกชนิดของสัญลักษณ์ที่จะนำมาใช้ในการแสดงได้ว่าจะใช้เป็นแบบกากบาท, แบบระบายทึบ หรือ เป็นสัญลักษณ์อื่นๆ

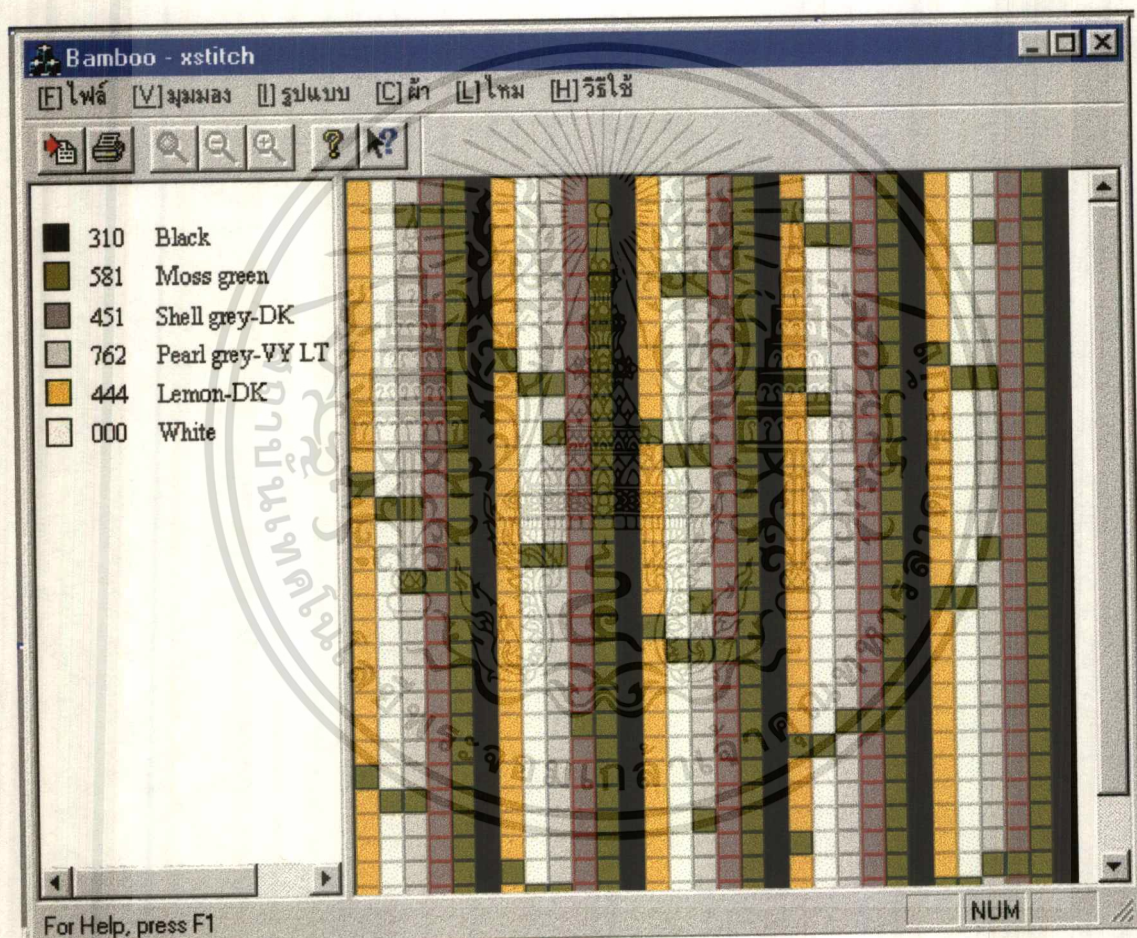
4.3 ความสามารถในการเพิ่ม, ลบ ข้อมูลของใหม่

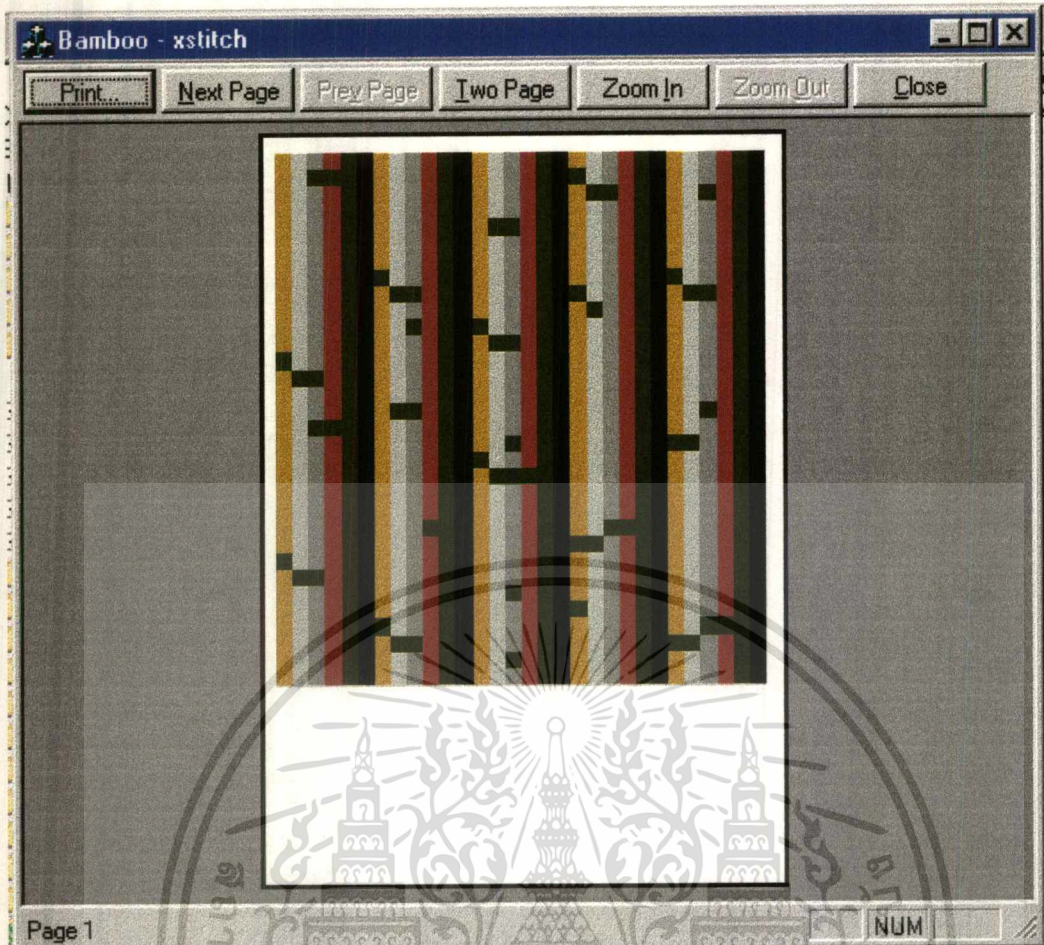
ผู้ใช้สามารถเพิ่มหรือลบข้อมูลของใหม่ได้โดยรันโปรแกรม MakeFloss.exe แล้วเปิดไฟล์ Dmc.mai เพื่อทำการแก้ไขฐานข้อมูลของใหม่ได้

4.4 ความสามารถในการดูตัวอย่างก่อนพิมพ์

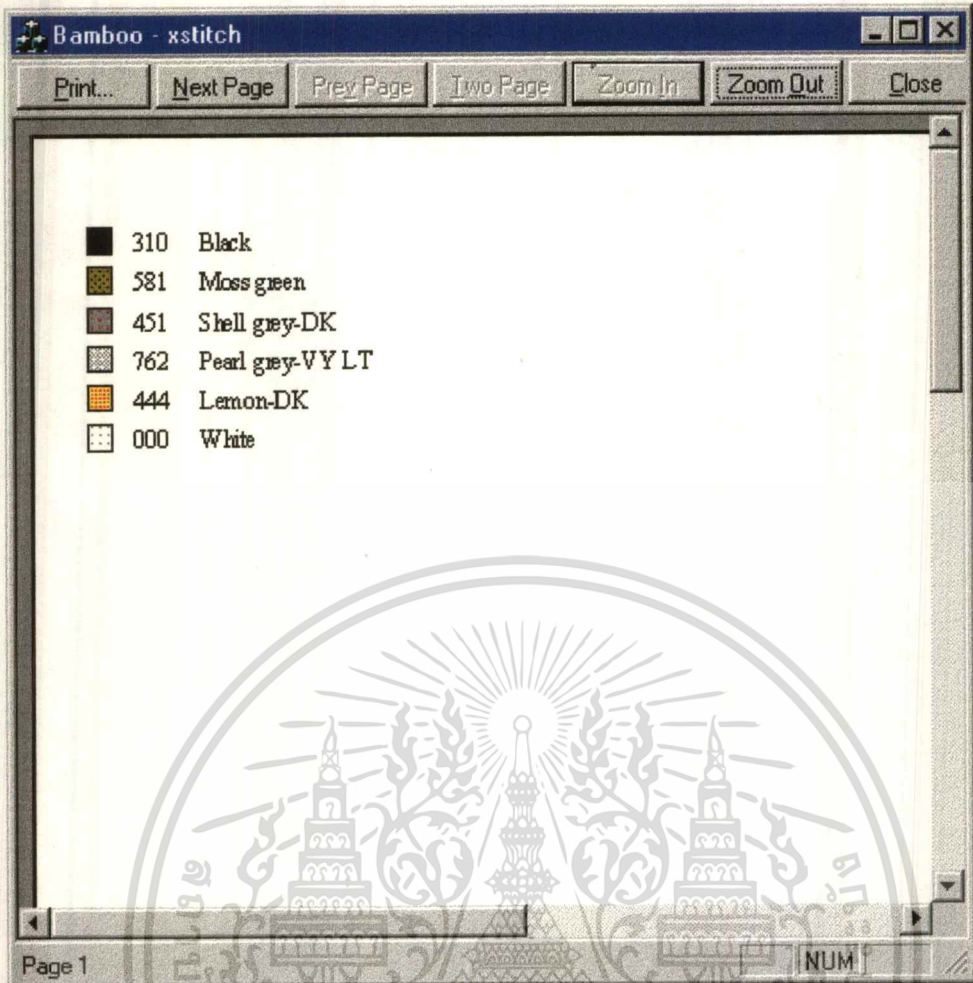
เนื่องจากแบบภาพปักนี้จะต้องทำการพิมพ์ออกมาทางเครื่องพิมพ์เพื่อใช้เป็นแบบในการปัก ดังนั้นจึงจำเป็นต้องมีการดูตัวอย่างก่อนพิมพ์ ทั้งนี้เพื่อให้ผู้ใช้สามารถตรวจสอบความถูกต้องได้ ซึ่งเป็นการอำนวยความสะดวก ทำให้ไม่ต้องเสียเวลาพิมพ์ใหม่ในกรณีที่แบบภาพปักที่พิมพ์ออกมานั้นไม่สวยงาม ไม่เป็นที่น่าพอใจ

ตัวอย่างของรูปที่ได้รับการแปลงแล้ว และตัวอย่างก่อนพิมพ์ แสดงดังรูป

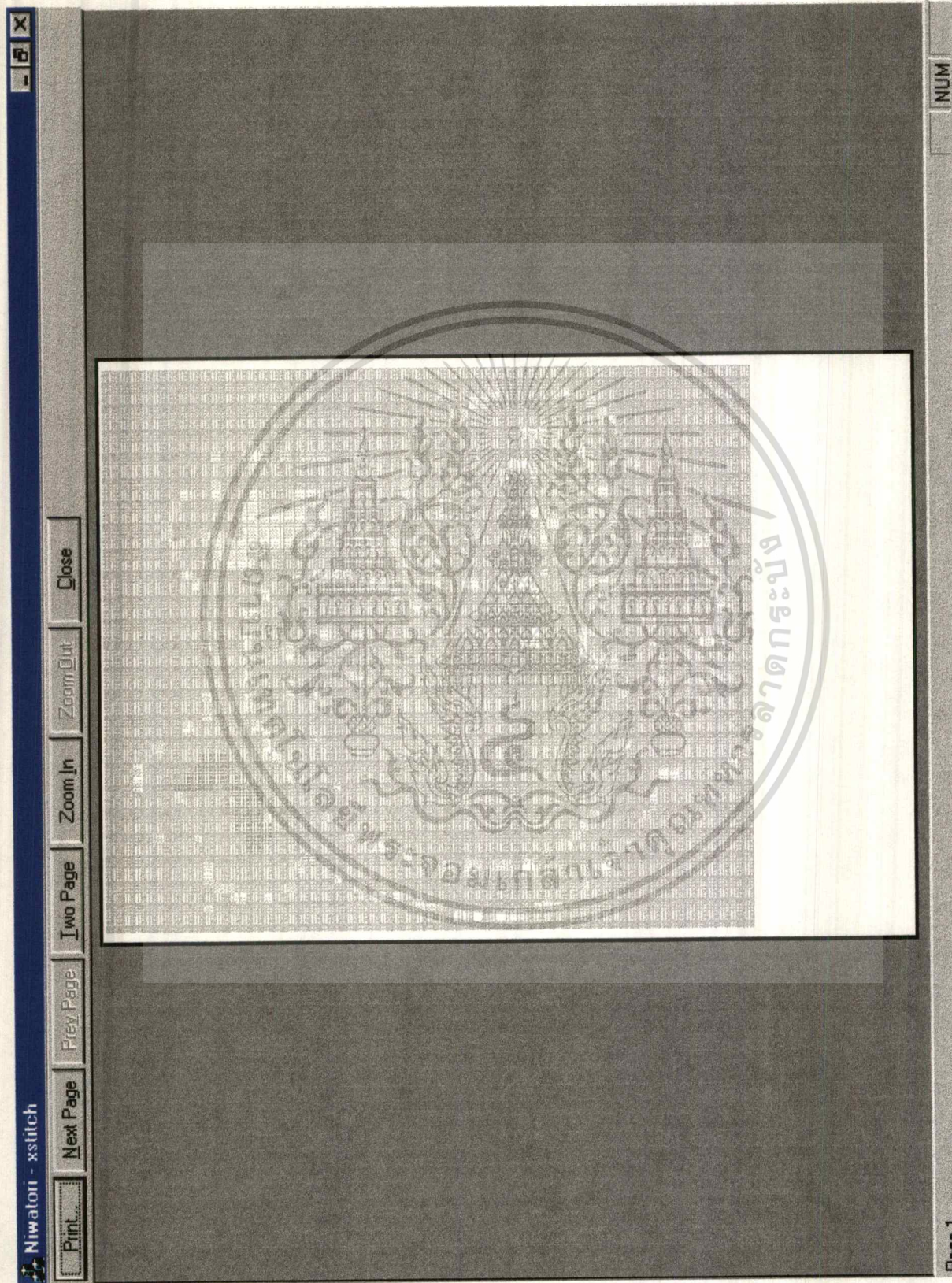




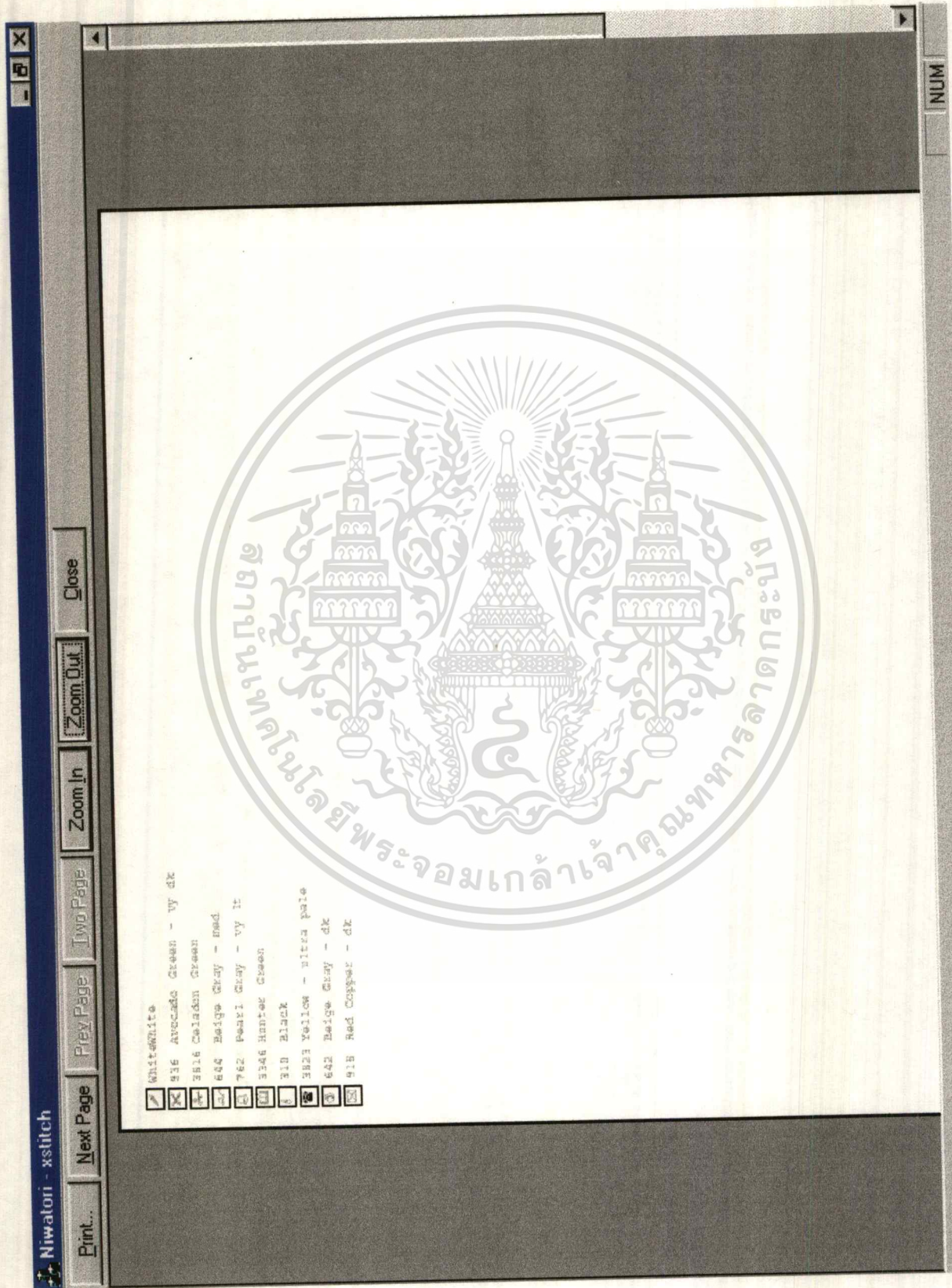
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

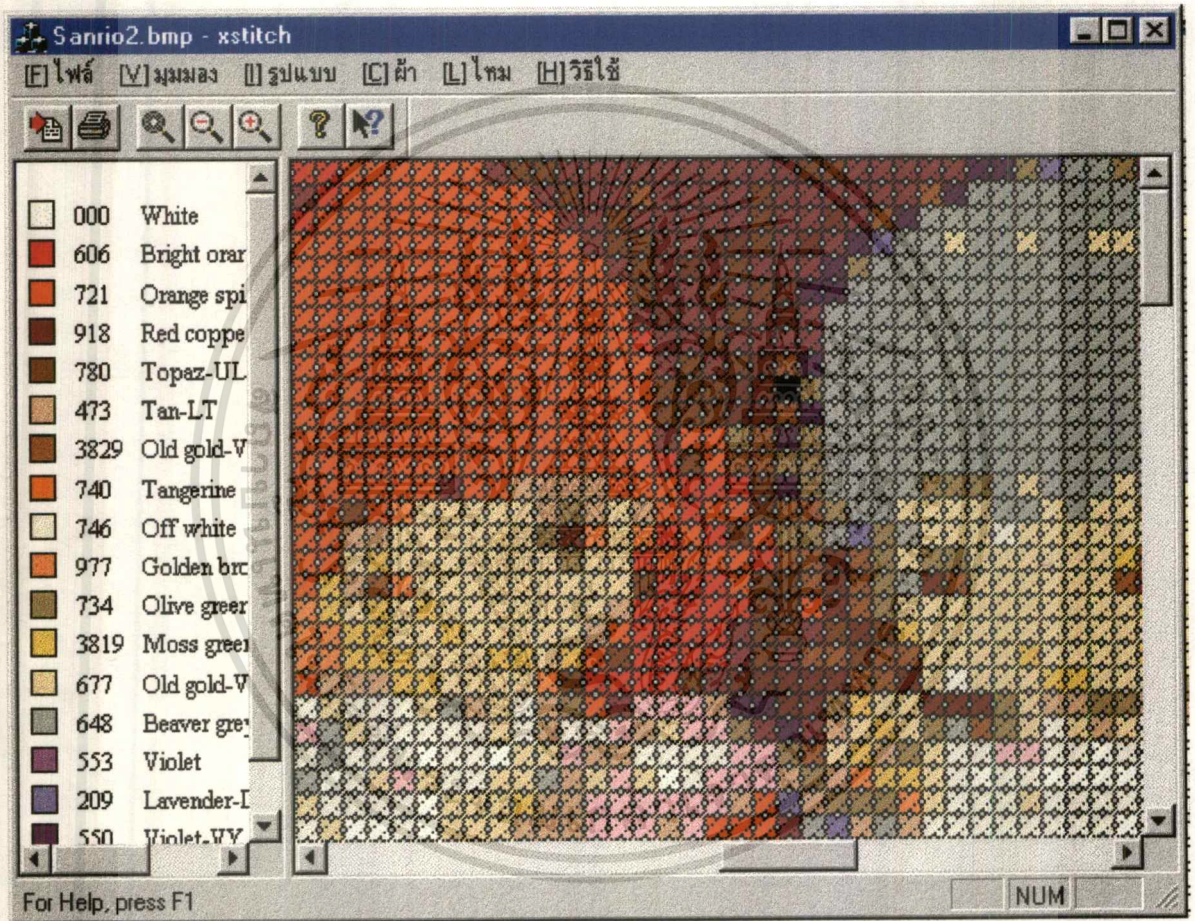


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

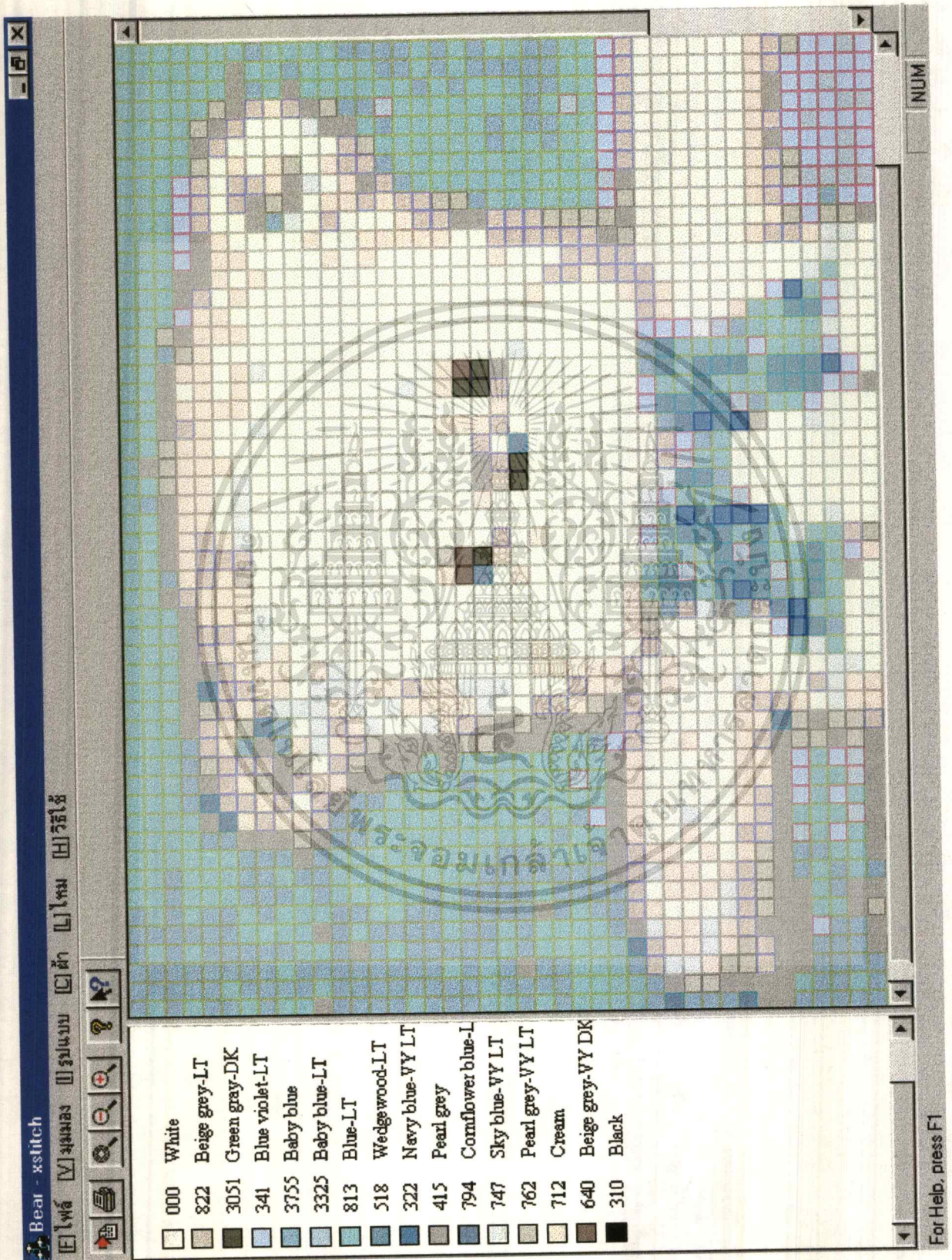
4.5 ผลที่ได้จากการทำงานของโปรแกรม

จะทำให้ได้แบบภาพปักครอสติชที่สวยงาม ถูกใจผู้ใช้ เนื่องจากผู้ใช้สามารถเลือกสายผ้า, ชนิดของสัญลักษณ์ และจำนวนสีของไหมได้เอง และยังเป็นการประหยัดค่าใช้จ่ายในการทำครอสติชได้ด้วย

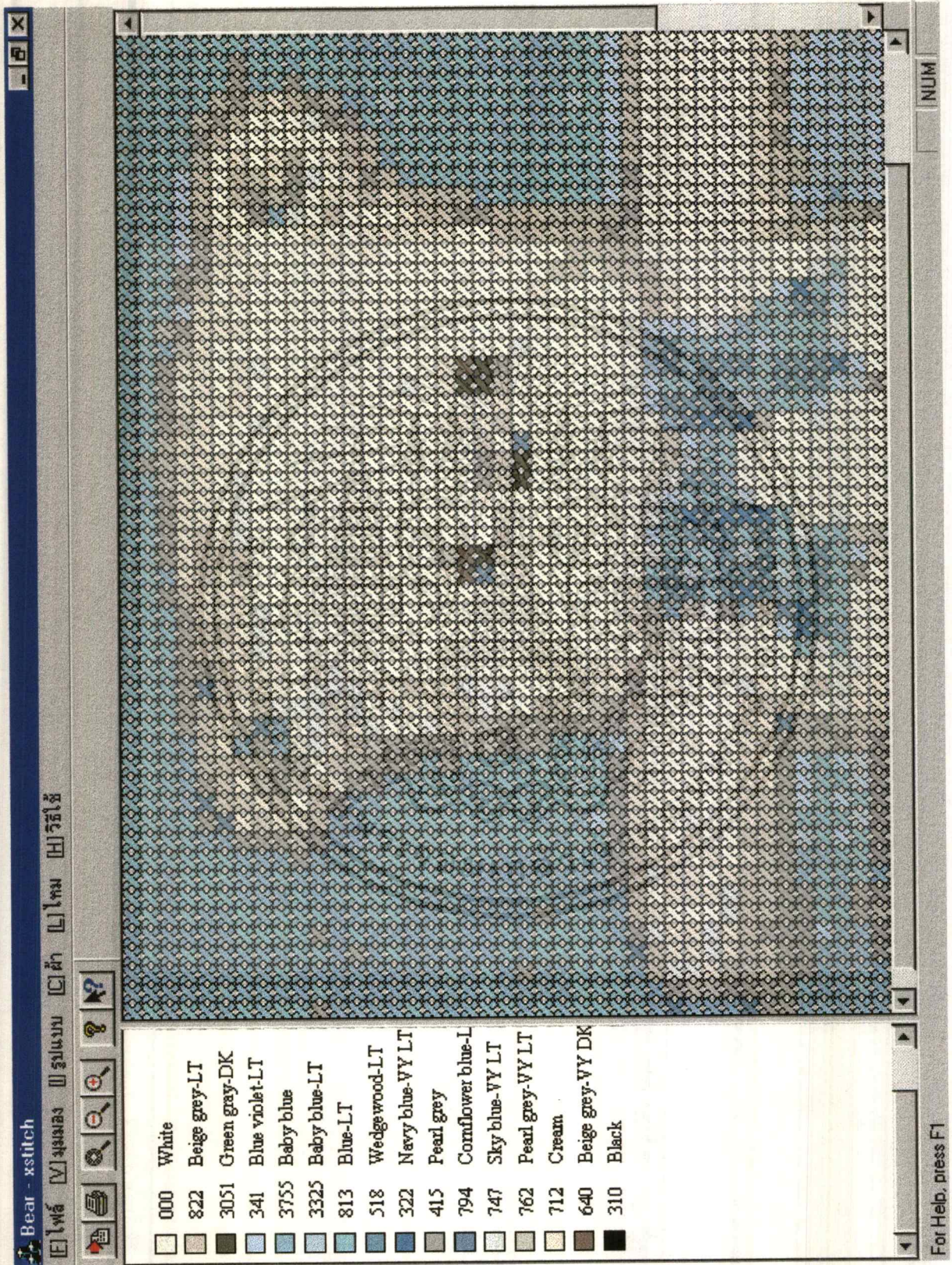
ตัวอย่างของรูปที่ได้รับการแปลงแล้ว แสดงได้ดังรูปต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bear - xstitch

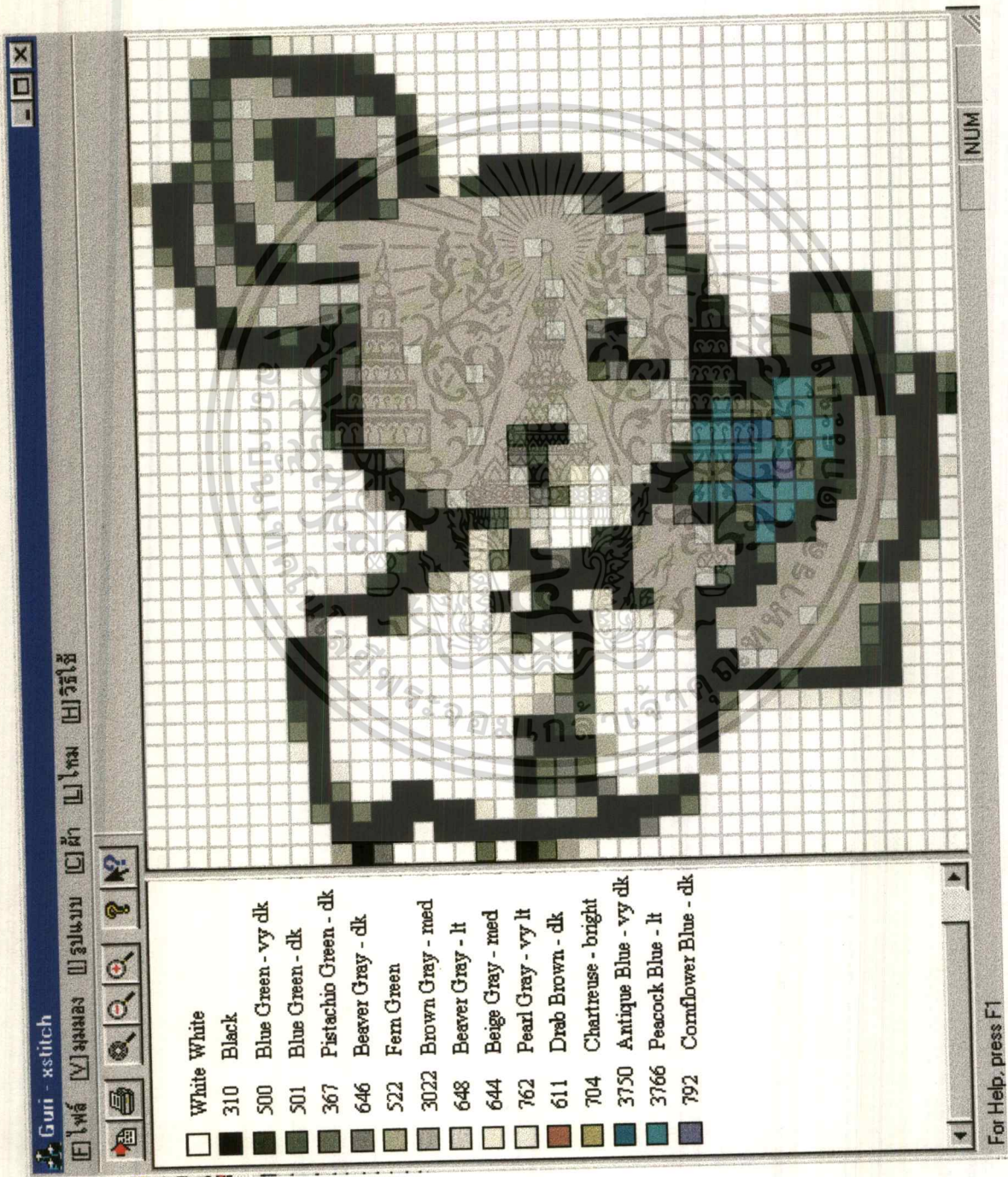
ไฟล์ [V] ลม [W] รูปแบบ [X] ผ้า [Y] โทม [H] ไร่ [Z]

000 White
 822 Beige gray-LT
 3051 Green gray-DK
 341 Blue violet-LT
 3755 Baby blue
 3325 Baby blue-LT
 813 Blue-LT
 518 Wedgewood-LT
 322 Navy blue-VY LT
 415 Pearl grey
 794 Cornflower blue-L
 747 Sky blue-VY LT
 762 Pearl grey-VY LT
 712 Cream
 640 Beige grey-VY DK
 310 Black

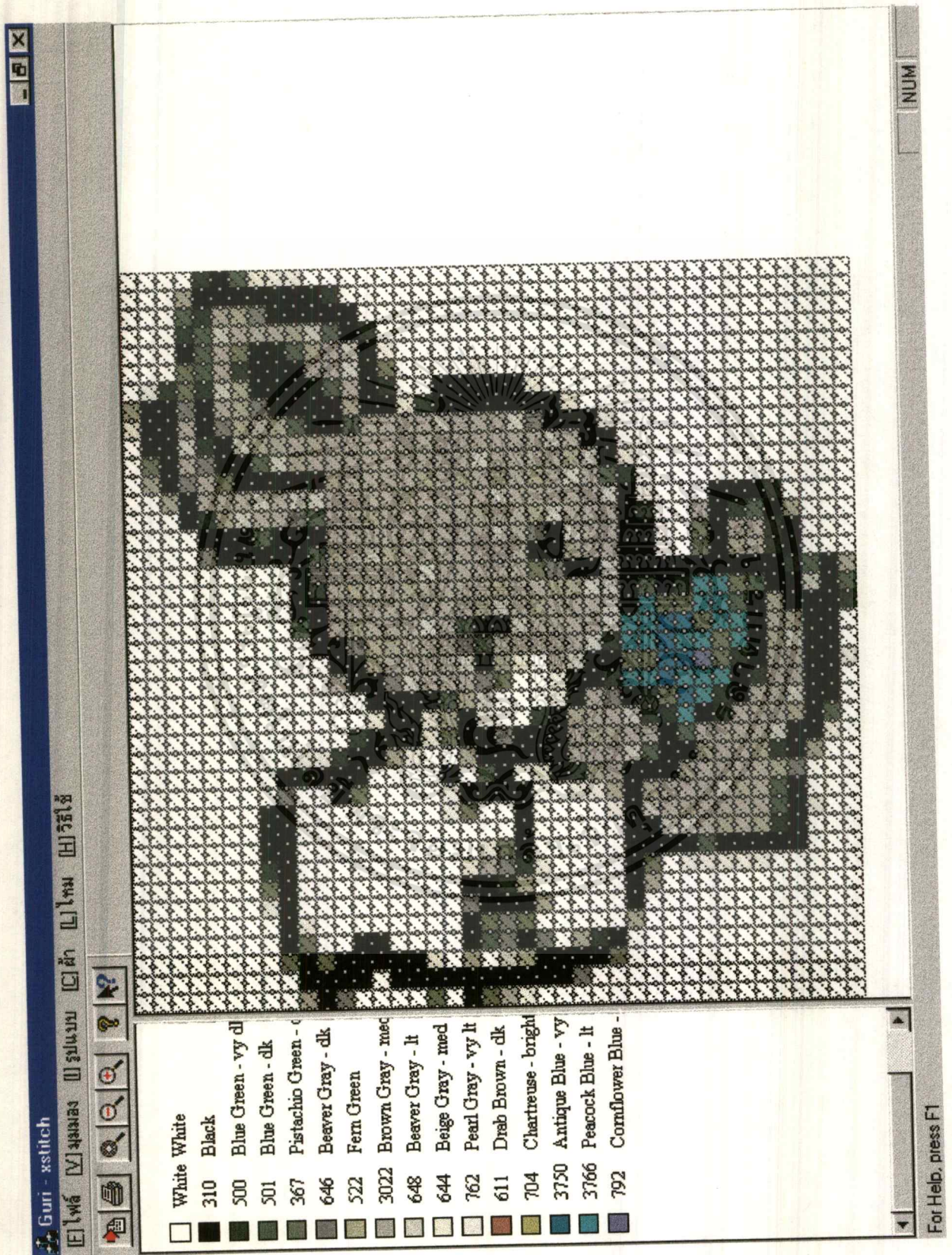
NUM

For Help, press F1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

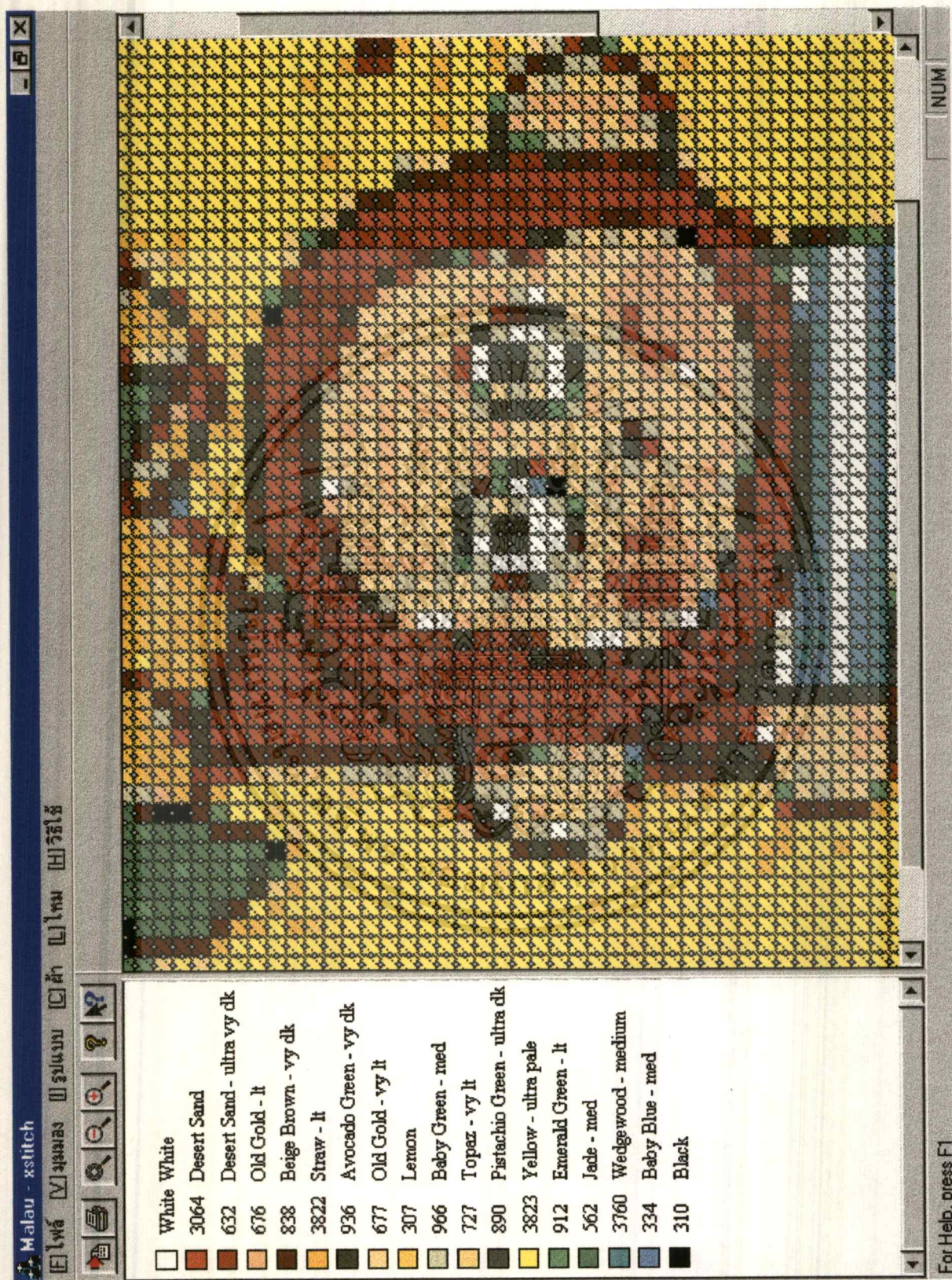
Guni - xstitch

ไฟล์ [V] ลม [M] รูปแบบ [C] ผ้า [L] โทม [H] วัสดุ

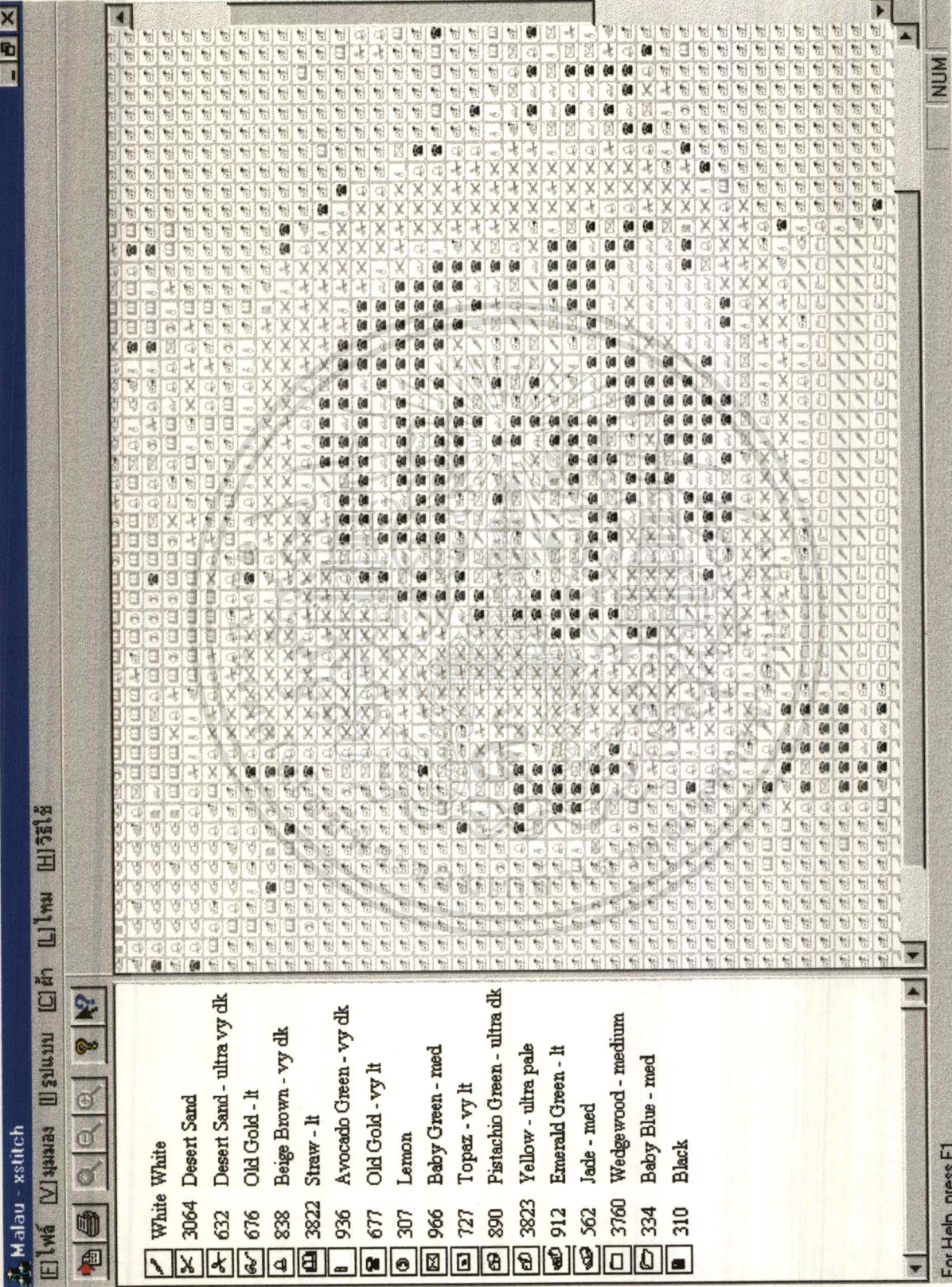
White White
 310 Black
 500 Blue Green - vy dk
 501 Blue Green - dk
 367 Pistachio Green - dk
 646 Beaver Gray - dk
 522 Fern Green
 3022 Brown Gray - med
 648 Beaver Gray - lt
 644 Beige Gray - med
 762 Pearl Gray - vy lt
 611 Drab Brown - dk
 704 Chartreuse - bright
 3750 Antique Blue - vy dk
 3766 Peacock Blue - lt
 792 Cornflower Blue - dk

For Help, press F1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- White
- 3064 Desert Sand
- 632 Desert Sand - ultra vy dk
- 676 Old Gold - lt
- 838 Beige Brown - vy dk
- 3822 Straw - lt
- 936 Avocado Green - vy dk
- 677 Old Gold - vy lt
- 307 Lemon
- 966 Baby Green - med
- 727 Topaz - vy lt
- 890 Pistachio Green - ultra dk
- 3823 Yellow - ultra pale
- 912 Emerald Green - lt
- 562 Jade - med
- 3760 Wedgewood - medium
- 334 Baby Blue - med
- 310 Black

For Help, press F1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pekkle - xstitch

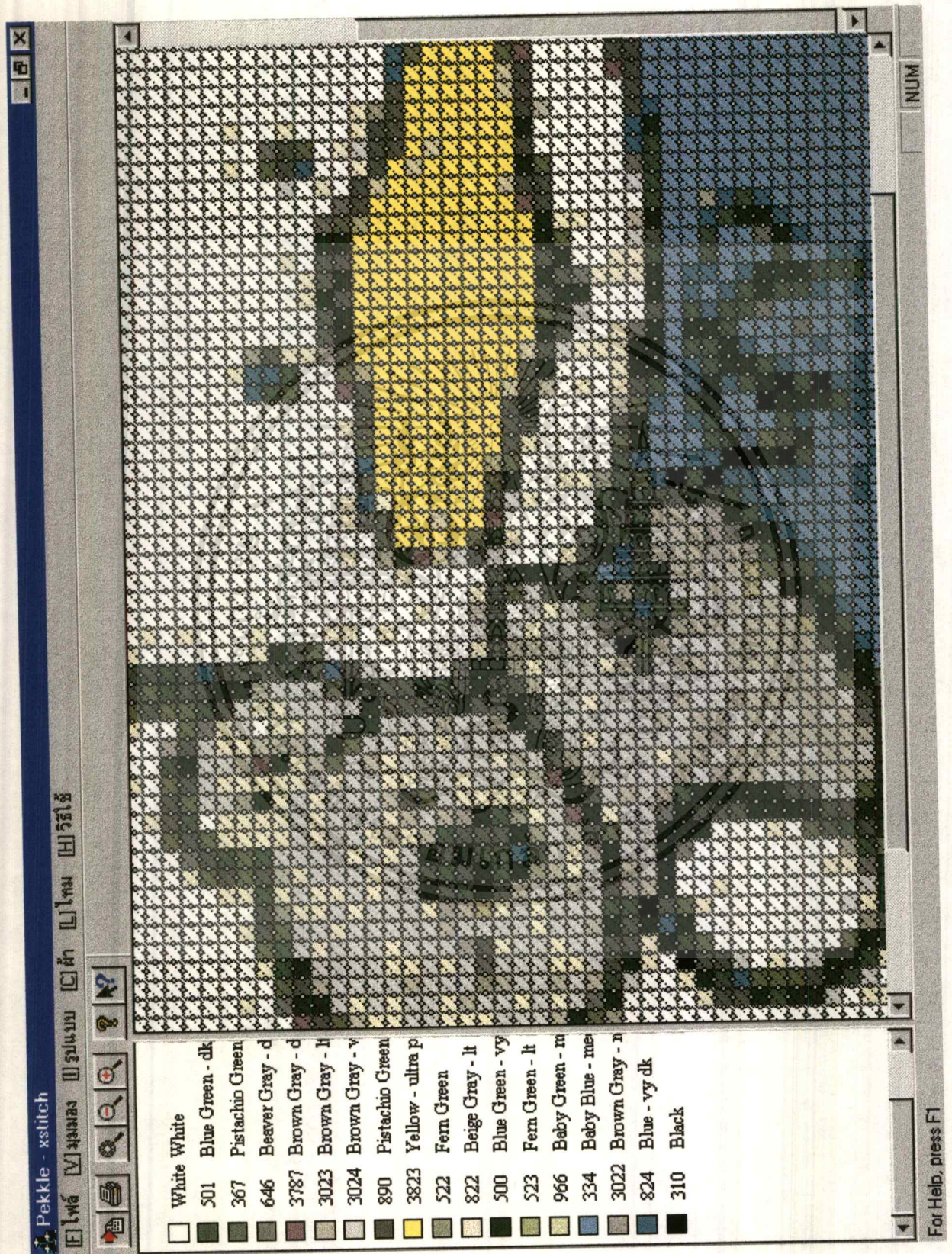
[E] ไฟล์ [V] ลายเย็บ [R] รูปแบบ [C] ผ้า [L] โหมด [H] รหัสใช้

White	501	Blue Green - dk	367	Pistachio Green	646	Beaver Gray - d	3787	Brown Gray - d	3023	Brown Gray - lt	3024	Brown Gray - v	890	Pistachio Green	3823	Yellow - ultra p	522	Fern Green	822	Beige Gray - lt	500	Blue Green - vy	523	Fern Green - lt	966	Baby Green - m	334	Baby Blue - me	3022	Brown Gray - n	824	Blue - vy dk	310	Black
-------	-----	-----------------	-----	-----------------	-----	-----------------	------	----------------	------	-----------------	------	----------------	-----	-----------------	------	------------------	-----	------------	-----	-----------------	-----	-----------------	-----	-----------------	-----	----------------	-----	----------------	------	----------------	-----	--------------	-----	-------

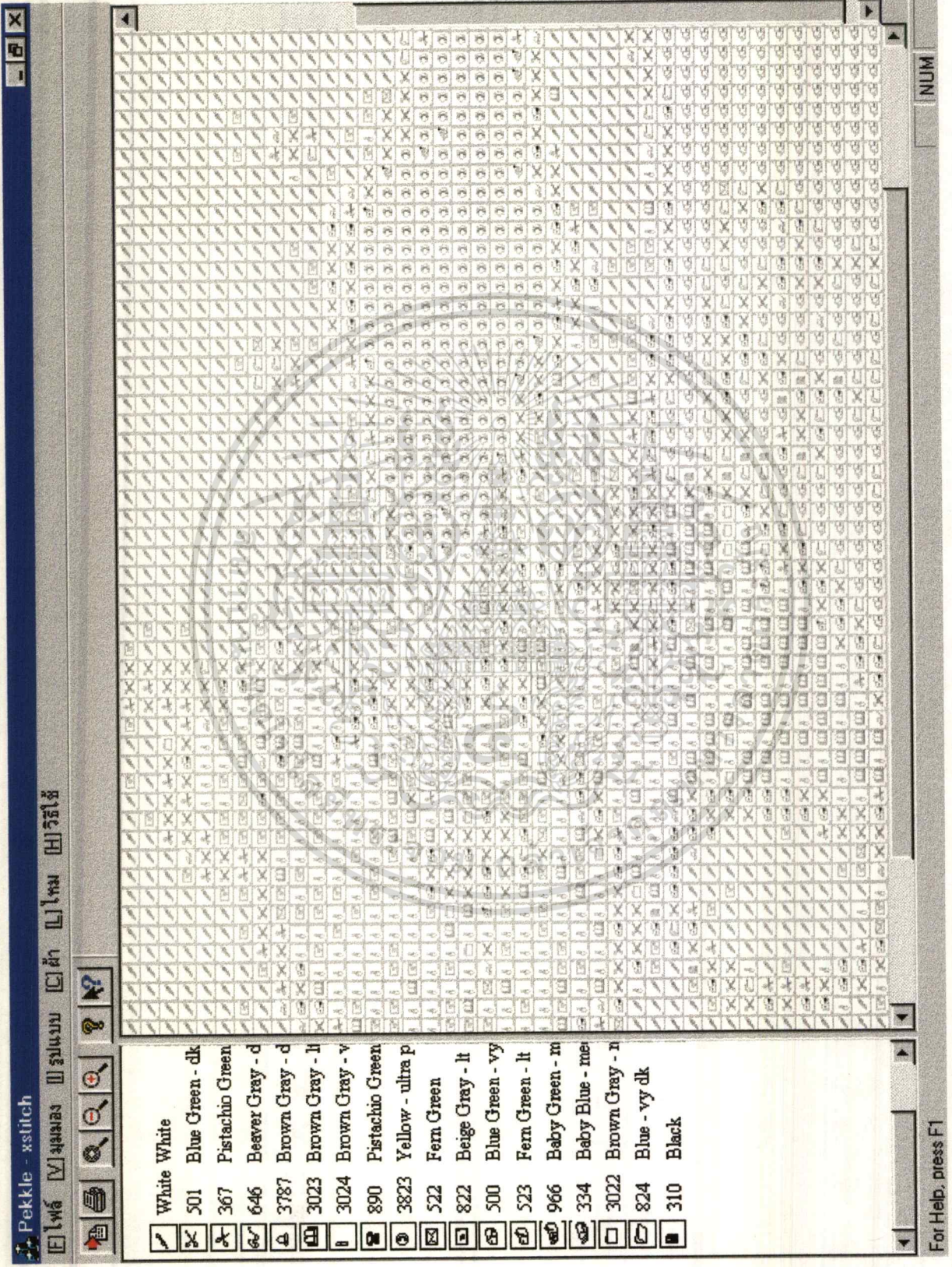
NUM

For Help, press F1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและเสนอแนะ

สรุปผล

ผลที่ได้จากการเขียนโปรแกรม โดยนำเอาเทคนิคทางด้านคอมพิวเตอร์กราฟฟิก มาช่วย ทำให้สามารถสร้างโปรแกรมสำเร็จรูปที่ใช้ในการแปลงรูปภาพที่เก็บในรูปแบบของบิตแมป ให้เป็นแบบภาพปิกครอสติชได้ ซึ่งทำให้ได้แบบภาพปิกที่ถูกต้องผู้ใช้ไม่ว่าจะเป็นในเรื่องของรูปภาพและลายผ้า

ในการพิมพ์แบบภาพปิกออกจากเครื่องพิมพ์ ผู้ใช้สามารถดูตัวอย่างก่อนพิมพ์ได้ เพื่อให้แบบภาพปิกที่จะพิมพ์ออกมา ได้รับการตรวจทานจากผู้ใช้งานถูกต้องและเป็นที่น่าพอใจหรือไม่ ทำให้ไม่สิ้นเปลืองกระดาษและเวลาในกรณีที่แบบภาพปิกไม่เป็นที่น่าพอใจ

ข้อเสนอแนะ

เนื่องจากระยะเวลาในการทำโครงการปัญหาพิเศษนี้ค่อนข้างจำกัด ดังนั้นจึงขอเสนอเพื่อให้ผู้ที่มีความสนใจในปัญหาพิเศษหัวข้อนี้ได้ทำการพัฒนาเพิ่มเติมดังต่อไปนี้

- การจัดเก็บไฟล์

เนื่องจากโปรแกรมนี้อาจไม่มีส่วนของการจัดเก็บไฟล์ กล่าวคือ โปรแกรมจะทำการ แปลงรูปภาพที่โหลดขึ้นมาเป็นแบบภาพปิกครอสติช และจะมีส่วนของการดูตัวอย่าง ก่อนพิมพ์ให้ดู แล้วจึงพิมพ์งานออกจากเครื่องพิมพ์ ผู้ใช้จะไม่สามารถจัดเก็บ (Save) ไฟล์ของแบบภาพปิกซึ่งได้รับการแปลงมาจากรูปภาพแล้วไว้ได้

- การแก้ไขแบบภาพปิกซึ่งได้รับการแปลงมาจากรูปภาพแล้ว

โปรแกรมไม่สามารถให้ผู้ใช้แก้ไขหรือตกแต่งแบบภาพปิกได้ ซึ่งในความเป็นจริงแล้วโปรแกรมควรจะมีส่วนจัดการในเรื่องนี้ด้วย เพื่อทำให้แบบภาพปิกที่ได้เป็นที่น่าพอใจแก่ผู้ใช้งานมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

ฐานข้อมูลของไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Embroidery Floss Colors

DMC offers you a selection of 396 colors, from the palest chartreuse to the deepest evergreen; from the lightest ochre to the darkest brown; from the softest pink to the most fiery red.

Color No.	Color Name	Color No.	Color Name	Color No.	Color Name
	White White	318	Steel Gray - lt	367	Pistachio Green - dk
	Ecru Ecru	319	Pistachio Green - vy dk	368	Pistachio Green - lt
208	Lavender - vy dk	320	Pistachio Green - med	369	Pistachio Green - vy lt
209	Lavender - dk	321	Christmas Red	370	Mustard - med
210	Lavender - med	322	Navy Blue - vy lt	371	Mustard
211	Lavender - lt	326	Rose - vy dk	372	Mustard - lt
221	Shell Pink - vy dk	327	Violet - vy dk	400	Mahogany - dk
223	Shell Pink - lt	333	Blue Violet - vy dk	402	Mahogany - vy lt
224	Shell Pink - vy lt	334	Baby Blue - med	407	Desert Sand - dk
225	Shell Pink - ultra vy lt	335	Rose	413	Pewter Gray - dk
300	Mahogany - vy dk	336	Navy Blue	414	Steel Gray - dk
301	Mahogany - med	340	Blue Violet - med	415	Pearl Gray
304	Christmas Red - med	341	Blue Violet - lt	420	Hazelnut Brown - dk
307	Lemon	347	Salmon - vy dk	422	Hazelnut Brown - lt
309	Rose - dk	349	Coral - dk	433	Brown - med
310	Black	350	Coral - med	434	Brown - lt
311	Navy Blue - med	351	Coral	435	Brown - vy lt
312	Navy Blue - lt	352	Coral - lt	436	Tan
315	Antique Mauve - vy dk	353	Peach	437	Tan - lt
316	Antique Mauve - med	355	Terra Cotta - dk	444	Lemon - dk
317	Pewter Gray	356	Terra Cotta - med	445	Lemon - lt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น กรุณาอย่าเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color No.	Color Name	Color No.	Color Name	Color No.	Color Name
451	Shell Gray - dk	581	Moss Green	703	Chartreuse
452	Shell Gray - med	597	Turquoise	704	Chartreuse - bright
453	Shell Gray - lt	598	Turquoise - lt	712	Cream
469	Avocado Green	600	Cranberry - vy dk	718	Plum
470	Avocado Green - lt	601	Cranberry - dk	720	Orange Spice - dk
471	Avocado Green - vy lt	602	Cranberry - med	721	Orange Spice - med
472	Avocado Green - ultra lt	603	Crabnerry	722	Orange Spice - lt
498	Christmas Red - dk	604	Cranberry - lt	725	Topaz
500	Blue Green - vy dk	605	Cranberry - vy lt	726	Topaz - lt
501	Blue Green - dk	606	Bright Orange - Red	727	Topaz - vy lt
502	Blue Green	608	Bright Orange	729	Old Gold - med
503*	Blue Green - med	610	Drab Brown - vy dk	730	Olive Green - vy dk
504	Blue Green - lt	611	Drab Brown - dk	731	Olive Green - dk
517	Wedgewood - med	612	Drab Brown - med	732	Olive Green
518	Wedgwood - lt	613	Drab Brown - lt	733	Olive Green - med
519	Sky Blue	632	Desert Sand - ultra vy dk	734	Olive Green - lt
520	Fern Green - dk	640	Beige Gray - vy dk	738	Tan - vy lt
522	Fern Green	642	Beige Gray - dk	739	Tan - ultra vy lt
523	Fern Green - lt	644	Beige Gray - med	740	Tangerine
524	Fern Green - vy lt	645	Beaver Gray - vy dk	741	Tangerine - med
535	Ash Gray - vy lt	646	Beaver Gray - dk	742	Tangerine - lt
543	Beige Brown - ultra vy lt	647	Beaver Gray - med	743	Yellow - med
550	Violet - vy dk	648	Beaver Gray - lt	744	Yellow - pale
552	Violet - med	666	Christmas Red - bright	745	Yellow - lt pale
553	Violet	676	Old Gold - lt	746	Off White
554	Violet - lt	677	Old Gold - vy lt	747	Sky Blue - vy lt
561	Jade - vy dk	680	Old Gold - dk	754	Peach - lt
562	Jade - med	699	Christmas Green	758	Terra Cotta - vy lt
563	Jade - lt	700	Christmas Green - bright	760	Salmon
564	Jade - vy lt	701	Christmas Green - lt	761	Salmon - lt
580	Moss Green - dk	702	Kelly Green	762	Pearl Gray - vy lt

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

“ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้”

Color No.	Color Name	Color No.	Color Name	Color No.	Color Name
772	Yellow Green - vy lt	824	Blue - vy dk	907	Parrot Green - lt
775	Baby Blue - vy lt	825	Blue - dk	909	Emerald Green - vy dk
776	Pink - med	826	Blue - med	910	Emerald Green - dk
778	Antique Mauve - vy lt	827	Blue - vy lt	911	Emerald Green - med
780	Topaz - ultra vy dk	828	Blue - ultra vy lt	912	Emerald Green - lt
781	Topaz - vy dk	829	Golden Olive - vy dk	913	Nile Green - med
782	Topaz - dk	830	Golden Olive - dk	915	Plum - dk
783	Topaz - med	831	Golden Olive - med	917	Plum - med
791	Cornflower Blue - vy dk	832	Golden Olive	918	Red Copper - dk
792	Cornflower Blue - dk	833	Golden Olive - lt	919	Red Copper
793	Cornflower Blue - med	834	Golden Olive - vy lt	920	Copper - med
794	Cornflower Blue - lt	838	Beige Brown - vy dk	921	Copper
796	Royal Blue - dk	839	Beige Brown - dk	922	Copper - lt
797	Royal Blue	840	Beige Brown - med	924	Gray Green - vy dk
798	Delft - dk	841	Beige Brown - lt	926	Gray Green - med
799	Delft - med	842	Beige Brown - vy lt	927	Gray Green - lt
800	Delft - pale	844	Beaver Gray - ultra dk	928	Gray Green - vy lt
801	Coffee Brown - dk	869	Hazelnut Brown - vy dk	930	Antique Blue - dk
806	Peacock Blue - dk	890	Pistachio Green - ultra dk	931	Antique Blue - med
807	Peacock Blue	891	Carnation - dk	932	Antique Blue - lt
809	Delft	892	Carnation - med	934	Black Avocado Green
813	Blue - lt	893	Carnation - lt	935	Avocado Green - dk
814	Garnet - dk	894	Carnation - vy lt	936	Avocado Green - vy dk
815	Garnet - med	895	Hunter Green - vy dk	937	Avocado Green - med
816	Garnet	898	Coffee Brown - vy dk	938	Coffee Brown - ultra dk
817	Coral Red - vy dk	899	Rose - med	939	Navy Blue - vy dk
818	Baby Pink	900	Burnt Orange - dk	943	Aquamarine - med
819	Baby Pink - lt	902	Garnet - vy dk	945	Tawny
820	Royal Blue - vy dk	904	Parrot Green - vy dk	946	Burnt Orange - med
822	Beige Grat - lt	905	Parrot Green - dk	947	Burnt Orange
823	Navy Blue - dk	906	Parrot Green - med	948	Peach - vy lt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้งานด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color No.	Color Name	Color No.	Color Name	Color No.	Color Name
950	Desert Sand - lt	3013	Khaki Green - lt	3363	Pine Green - med
951	Tawny - lt	3021	Brown Gray - vy dk	3364	Pine Green
954	Nile Green	3022	Brown Gray - med	3371	Black Brown
955	Nile Green - lt	3023	Brown Gray - lt	3607	Plum - lt
956	Geranium	3024	Brown Gray - vy lt	3608	Plum - vy lt
957	Geranium - pale	3031	Mocha Brown - vy dk	3609	Plum - ultra lt
958	Seagreen - dk	3032	Mocha Brown - med	3685	Mauve - dk
959	Seagreen - med	3033	Mocha Brown - vy lt	3687	Mauve
961	Dusty Rose - dk	3041	Antique Violet - med	3688	Mauve - med
962	Dusty Rose - med	3042	Antique Violet - lt	3689	Mauve - lt
963	Dusty Rose - ultra vy lt	3045	Yellow Beige - dk	3705	Melon - dk
964	Seagreen - ltue - lt	3046	Yellow Beige - med	3706	Melon - med
966	Baby Green - med	3047	Yellow Beige - lt	3708	Melon - lt
970	Pumpkin - lt/P>	3051	Green Gray - dk	3712	Salmon - med
971	Pumpkin	3052	Green Gray - med	3713	Salmon - vy lt
972	Canary - deep	3053	Green Gray	3716	Dusty Rose - vy lt
973	Canary - bright	3064	Desert Sand	3721	Shell Pink - dk
975	Golden Brown - dk	3072	Beaver Gray - vy lt	3722	Shell Pink - med
976	Golden Brown - med	3078	Golden Yellow - vy lt	3726	Antique Mauve -dk
977	Golden Brown - lt	3325	Baby Blue - lt	3727	Antique Mauve - lt
986	Forest Green - vy dk	3326	Rose - lt	3731	Dusty Rose - vy dk
987	Forest Green - dk	3328	Salmon - dk	3733	Dusty Rose
988	Forest Green - med	3340	Apricot - med	3740	Antique Violet - dk
989	Forest Green	3341	Apricot	3743	Antique Violet - vy lt
991	Aquamarine - dk	3345	Hunter Green - dk	3746	Blue Violet - dk
992	Aquamarine - med	3346	Hunter Green	3750	Antique Blue - vy dk
993	Aquamarine - lt	3347	Yellow Green - med	3752	Antique Blue - vy lt
995	Electric Blue - dk	3348	Yellow Green - lt	3753	Antique Blue - ultra vy lt
996	Electric Blue - med	3350	Dusty Rose - ultra dk	3755	Baby Blue
3011	Khaki Green - dk	3354	Dusty Rose - lt	3756	Baby Blue - ultra vy lt
3012	Khaki Green - med	3362	Pine Green - dk	3760	Wedgewood - medium

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปยังบุคคลอื่นโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color No.	Color Name	Color No.	Color Name	Color No.	Color Name
3761	Sky Blue - lt	3799	Pewter Gray - vy dk	3816	Celadon Green
3765	Peacock Blue - vy dk	3801	Christmas Red - lt	3817	Celadon Green - lt
3766	Peacock Blue - lt	3802	Antique Mauve - vy dk	3818	Emerald Green - ultra vy
3768	Gray Green - dk	3803	Mauve - med	3819	Moss Green - lt
3770	Tawny - vy lt	3804	Cyclamen Pink - dk	3820	Straw - dk
3772	Desert Sand - vy dk	3805	Cyclamen Pink	3821	Straw
3773	Desert Sand - med	3806	Cyclamen Pink - lt	3822	Straw - lt
3774	Desert Sand - vy lt	3807	Cornflower Blue	3823	Yellow - ultra pale
3776	Mahogany - lt	3808	Turquoise - ultra vy dk	3824	Apricot - lt
3777	Terra Cotta - vy dk	3809	Turquoise - vy dk	3825	Pumpkin - pale
3778	Terra Cotta - lt	3810	Turquoise - dk	3826	Golden Brown
3779	Terra Cotta - ultra vy lt	3811	Turquoise - vy lt	3827	Golden Brown - pale
3781	Mocha Brown - dk	3812	Seagreen - vy dk	3828	Hazelnut Brown
3782	Mocha Brown - lt	3813	Blue Green - lt	3829	Old Gold - vy dk
3787	Brown Gray - dk	3814	Aquamarine	3830	Terra Cotta
3790	Beige Gray - ultra dk	3815	Celadon Green - dk		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Source Code

โปรแกรม Xstitch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AppWizard has created this xstitch application for you. This application not only demonstrates the basics of using the Microsoft Foundation classes but is also a starting point for writing your application.

This file contains a summary of what you will find in each of the files that make up your xstitch application.

xstitch.h

This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CXstitchApp application class.

xstitch.cpp

This is the main application source file that contains the application class CXstitchApp.

xstitch.rc

This is a listing of all of the Microsoft Windows resources that the program uses. It includes the icons, bitmaps, and cursors that are stored in the RES subdirectory. This file can be directly edited in Microsoft Developer Studio.

res\xstitch.ico

This is an icon file, which is used as the application's icon. This icon is included by the main resource file xstitch.rc.

res\xstitch.rc2

This file contains resources that are not edited by Microsoft Developer Studio. You should place all resources not editable by the resource editor in this file.

xstitch.clw

This file contains information used by ClassWizard to edit existing classes or add new classes. ClassWizard also uses this file to store information needed to create and edit message maps and dialog data maps and to create prototype member functions.

////////////////////////////////////

For the main frame window:

MainFrm.h, MainFrm.cpp

These files contain the frame class CMainFrame, which is derived from CFrameWnd and controls all SDI frame features.

res\Toolbar.bmp

This bitmap file is used to create tiled images for the toolbar. The initial toolbar and status bar are constructed in the CMainFrame class. Edit this toolbar bitmap along with the array in MainFrm.cpp to add more toolbar buttons.

////////////////////////////////////

AppWizard creates one document type and one view:

xstitchDoc.h, xstitchDoc.cpp - the document

These files contain your CXstitchDoc class. Edit these files to add your special document data and to implement file saving and loading (via CXstitchDoc::Serialize).

xstitchView.h, xstitchView.cpp - the view of the document
These files contain your CXstitchView class.
CXstitchView objects are used to view CXstitchDoc objects.

//

Help Support:

MakeHelp.bat

Use this batch file to create your application's Help file, xstitch.hLP.

xstitch.hpj

This file is the Help Project file used by the Help compiler to create your application's Help file.

hlp*.bmp

These are bitmap files required by the standard Help file topics for Microsoft Foundation Class Library standard commands.

hlp*.rtf

This file contains the standard help topics for standard MFC commands and screen objects.

//
Other standard files:

StdAfx.h, StdAfx.cpp

These files are used to build a precompiled header (PCH) file named xstitch.pch and a precompiled types file named StdAfx.obj.

Resource.h

This is the standard header file, which defines new resource IDs. Microsoft Developer Studio reads and updates this file.

//
Other notes:

AppWizard uses "TODO:" to indicate parts of the source code you should add to or customize.

If your application uses MFC in a shared DLL, and your application is in a language other than the operating system's current language, you will need to copy the corresponding localized resources MFC40XXX.DLL from the Microsoft Visual C++ CD-ROM onto the system or system32 directory, and rename it to be MFCLOC.DLL. ("XXX" stands for the language abbreviation. For example, MFC40DEU.DLL contains resources translated to German.) If you don't do this, some of the UI elements of your application will remain in the language of the operating system.

//

```
[OPTIONS]
CONTENTS=main_index
TITLE=XSTITCH Application Help
COMPRESS=true
WARNING=2
BMROOT= ....
ROOT= ....
```

```
[FILES]
afxcore.rtf
afxprint.rtf
```

```
[ALIAS]
HIDR_MAINFRAME = main_index
HIDD_ABOUTBOX = HID_APP_ABOUT
```

```
HID_HT_SIZE = HID_SC_SIZE
HID_HT_HSCROLL = scrollbars
HID_HT_VSCROLL = scrollbars
HID_HT_MINBUTTON = HID_SC_MINIMIZE
HID_HT_MAXBUTTON = HID_SC_MAXIMIZE
AFX_HIDP_INVALID_FILENAME = AFX_HIDP_default
AFX_HIDP_FAILED_TO_OPEN_DOC = AFX_HIDP_default
AFX_HIDP_FAILED_TO_SAVE_DOC = AFX_HIDP_default
AFX_HIDP_ASK_TO_SAVE = AFX_HIDP_default
AFX_HIDP_FAILED_TO_CREATE_DOC = AFX_HIDP_default
AFX_HIDP_FILE_TOO_LARGE = AFX_HIDP_default
AFX_HIDP_FAILED_TO_START_PRINT = AFX_HIDP_default
AFX_HIDP_FAILED_TO_LAUNCH_HELP = AFX_HIDP_default
AFX_HIDP_INTERNAL_FAILURE = AFX_HIDP_default
AFX_HIDP_COMMAND_FAILURE = AFX_HIDP_default
AFX_HIDP_PARSE_INT = AFX_HIDP_default
AFX_HIDP_PARSE_REAL = AFX_HIDP_default
AFX_HIDP_PARSE_INT_RANGE = AFX_HIDP_default
AFX_HIDP_PARSE_REAL_RANGE = AFX_HIDP_default
AFX_HIDP_PARSE_STRING_SIZE = AFX_HIDP_default
AFX_HIDP_FAILED_INVALID_FORMAT = AFX_HIDP_default
AFX_HIDP_FAILED_INVALID_PATH = AFX_HIDP_default
AFX_HIDP_FAILED_DISK_FULL = AFX_HIDP_default
AFX_HIDP_FAILED_ACCESS_READ = AFX_HIDP_default
AFX_HIDP_FAILED_ACCESS_WRITE = AFX_HIDP_default
AFX_HIDP_FAILED_IO_ERROR_READ = AFX_HIDP_default
AFX_HIDP_FAILED_IO_ERROR_WRITE = AFX_HIDP_default
AFX_HIDP_STATIC_OBJECT = AFX_HIDP_default
AFX_HIDP_FAILED_TO_CONNECT = AFX_HIDP_default
AFX_HIDP_SERVER_BUSY = AFX_HIDP_default
AFX_HIDP_BAD_VERB = AFX_HIDP_default
AFX_HIDP_FAILED_MEMORY_ALLOC = AFX_HIDP_default
AFX_HIDP_FAILED_TO_NOTIFY = AFX_HIDP_default
AFX_HIDP_FAILED_TO_LAUNCH = AFX_HIDP_default
AFX_HIDP_ASK_TO_UPDATE = AFX_HIDP_default
AFX_HIDP_FAILED_TO_UPDATE = AFX_HIDP_default
AFX_HIDP_FAILED_TO_REGISTER = AFX_HIDP_default
AFX_HIDP_FAILED_TO_AUTO_REGISTER = AFX_HIDP_default
```

```
[MAP]
#include <E:\MSDEV\MFC\include\afxhelp.hm>
#include <xstitch.hm>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// stdafx.h : include file for standard system include files,  
// or project specific include files that are used frequently, but  
// are changed infrequently  
//  
  
#define VC_EXTRALEAN // Exclude rarely-used stuff from Windows headers  
  
#include <afxwin.h> // MFC core and standard components  
#include <afxext.h> // MFC extensions  
#ifndef _AFX_NO_AFXCMN_SUPPORT  
#include <afxcmn.h> // MFC support for Windows 95 Common Controls  
#endif // _AFX_NO_AFXCMN_SUPPORT
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// stdafx.cpp : source file that includes just the standard includes
// xstitch.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (C) 1992 Microsoft Corporation
// All rights reserved.
//
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and Microsoft
// QuickHelp and/or WinHelp documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.

#ifndef _INC_DIBAPI
#define _INC_DIBAPI

/* Handle to a DIB */
DECLARE_HANDLE(HDIB);

/* DIB constants */
#define PALVERSION 0x300

/* DIB Macros*/

#define IS_WIN30_DIB(lpbi) ((*(LPDWORD)(lpbi)) == sizeof(BITMAPINFOHEADER))
#define RECTWIDTH(lpRect) ((lpRect)->right - (lpRect)->left)
#define RECTHEIGHT(lpRect) ((lpRect)->bottom - (lpRect)->top)

// WIDTHBYTES performs DWORD-aligning of DIB scanlines. The "bits"
// parameter is the bit count for the scanline (biWidth * biBitCount),
// and this macro returns the number of DWORD-aligned bytes needed
// to hold those bits.

#define WIDTHBYTES(bits) (((bits) + 31) / 32 * 4)

/* Function prototypes */
BOOL WINAPI PaintDIB (HDC, LPRECT, HDIB, LPRECT, CPalette* pPal, DWORD dwROP);
BOOL WINAPI CreateDIBPalette(HDIB hDIB, CPalette* cPal, COleArray* m_FlossArray, CString path);
LPSTR WINAPI FindDIBBits (LPSTR lpbi);
DWORD WINAPI DIBWidth (LPSTR lpDIB);
DWORD WINAPI DIBHeight (LPSTR lpDIB);
WORD WINAPI PaletteSize (LPSTR lpbi);
WORD WINAPI DIBNumColors (LPSTR lpbi);
HGLOBAL WINAPI CopyHandle (HGLOBAL h);

BOOL WINAPI SaveDIB (HDIB hDib, CFile& file);
HDIB WINAPI ReadDIBFile(CFile& file);

#endif // !_INC_DIBAPI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// dibapi.cpp
//
// Source file for Device-Independent Bitmap (DIB) API. Provides
// the following functions:
//
// PaintDIB()           - Painting routine for a DIB
// CreateDIBPalette()   - Creates a palette from a DIB
// FindDIBBits()        - Returns a pointer to the DIB bits
// DIBWidth()           - Gets the width of the DIB
// DIBHeight()          - Gets the height of the DIB
// PaletteSize()        - Gets the size required to store the DIB's palette
// DIBNumColors()        - Calculates the number of colors
//                       in the DIB's color table
// CopyHandle()         - Makes a copy of the given global memory block
//
// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (C) 1992 Microsoft Corporation
// All rights reserved.
//
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and Microsoft
// QuickHelp and/or WinHelp documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.

```

```

#include "stdafx.h"
#include "dibapi.h"
#include "FlossData.h"
#include <io.h>
#include <errno.h>

```

```

/*****
 *
 * PaintDIB()
 *
 * Parameters:
 *
 * HDC hDC           - DC to do output to
 *
 * LPRECT lpDcRect   - rectangle on DC to do output to
 *
 * HDIB hDIB         - handle to global memory with a DIB spec
 *                   in it followed by the DIB bits
 *
 * LPRECT lpDIBRect  - rectangle of DIB to output into lpDcRect
 *
 * CPalette* pPal    - pointer to CPalette containing DIB's palette
 *
 * Return Value:
 *
 * BOOL              - TRUE if DIB was drawn, FALSE otherwise
 *
 * Description:
 *   Painting routine for a DIB. Calls StretchDIBits() or
 *   SetDIBitsToDevice() to paint the DIB. The DIB is
 *   output to the specified DC, at the coordinates given
 *   in lpDcRect. The area of the DIB to be output is
 *   given by lpDIBRect.
 *
 *****/

```

```

BOOL WINAPI PaintDIB(HDC hDC,
                    LPRECT lpDcRect,
                    HDIB hDIB,
                    LPRECT lpDIBRect,
                    CPalette* pPal)

```

สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกข้อควรระวังคือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LPRECT lpDIBRect,  
CPalette* pPal,  
DWORD dwRop)
```

```
{  
LPSTR lpDIBHdr; // Pointer to BITMAPINFOHEADER  
LPSTR lpDIBBits; // Pointer to DIB bits  
BOOL bSuccess=FALSE; // Success/fail flag  
HPALETTE hPal=NULL; // Our DIB's palette  
HPALETTE holdPal=NULL; // Previous palette  
  
/* Check for valid DIB handle */  
if (hDIB == NULL)  
return FALSE;  
  
/* Lock down the DIB, and get a pointer to the beginning of the bit  
* buffer  
*/  
lpDIBHdr = (LPSTR) ::GlobalLock((HGLOBAL) hDIB);  
lpDIBBits = ::FindDIBBits(lpDIBHdr);  
  
// Get the DIB's palette, then select it into DC  
if (pPal != NULL)  
{  
hPal = (HPALETTE) pPal->m_hObject;  
  
// Select as background since we have  
// already realized in foreground if needed  
holdPal = ::SelectPalette(hDC, hPal, TRUE);  
}  
  
/* Make sure to use the stretching mode best for color pictures */  
::SetStretchBltMode(hDC, COLORONCOLOR);  
  
/* Determine whether to call StretchDIBits() or SetDIBitsToDevice() */  
if ((RECTWIDTH(lpDCRect) == RECTWIDTH(lpDIBRect)) &&  
(RECTHEIGHT(lpDCRect) == RECTHEIGHT(lpDIBRect)))  
bSuccess = ::SetDIBitsToDevice(hDC, // hDC  
lpDCRect->left, // DestX  
lpDCRect->top, // DestY  
RECTWIDTH(lpDCRect), // nDestWidth  
RECTHEIGHT(lpDCRect), // nDestHeight  
lpDIBRect->left, // SrcX  
(int)DIBHeight(lpDIBHdr) - // SrcY  
lpDIBRect->top -  
RECTHEIGHT(lpDIBRect), // nStartScan  
0, // nNumScans  
(WORD)DIBHeight(lpDIBHdr), // lpBits  
lpDIBBits, // lpBitsInfo  
(LPBITMAPINFO)lpDIBHdr, // wUsage  
DIB_RGB_COLORS);  
else  
bSuccess = ::StretchDIBits(hDC, // hDC  
lpDCRect->left, // DestX  
lpDCRect->top, // DestY  
RECTWIDTH(lpDCRect), // nDestWidth  
RECTHEIGHT(lpDCRect), // nDestHeight  
lpDIBRect->left, // SrcX  
lpDIBRect->top, // SrcY  
RECTWIDTH(lpDIBRect), // wSrcWidth  
RECTHEIGHT(lpDIBRect), // wSrcHeight  
lpDIBBits, // lpBits  
(LPBITMAPINFO)lpDIBHdr, // lpBitsInfo  
DIB_RGB_COLORS, // wUsage  
dwRop);  
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น กรุณาอย่าเผยแพร่ให้บุคคลอื่น
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแต่งเอกสาร และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง
กรุณาอย่าเผยแพร่เอกสารนี้ไปภายนอก

```

::GlobalUnlock((HGLOBAL) hDIB);

/* Reselect old palette */
if (hOldPal != NULL)
{
    ::SelectPalette(hDC, hOldPal, TRUE);
}

return bSuccess;
}

/*****;*****/
*
* CreatedDIBPalette()
*
* Parameter:
*
* HDIB hDIB          - specifies the DIB
*
* Return Value:
*
* HPALETTE          - specifies the palette
*
* Description:
*
* This function creates a palette from a DIB by allocating memory for the
* logical palette, reading and storing the colors from the DIB's color table
* into the logical palette, creating a palette from this logical palette,
* and then returning the palette's handle. This allows the DIB to be
* displayed using the best possible colors (important for DIBs with 256 or
* more colors).
*
*****/

BOOL WINAPI CreatedDIBPalette(HDIB hDIB, CPalette* pPal,
                             CObArray* m_FlossArray, CString path)
{
    LPLOGPALETTE lpPal;          // pointer to a logical palette
    HANDLE hLogPal;             // handle to a logical palette
    HPALETTE hPal = NULL;       // handle to a palette
    int i;                      // loop index
    WORD wNumColors;           // number of colors in color table
    LPSTR lpbi;                 // pointer to packed-DIB
    LPBITMAPINFO lpbmi;         // pointer to BITMAPINFO structure (Win3.0)
    LPBITMAPCOREINFO lpbmc;     // pointer to BITMAPCOREINFO structure (old)
    BOOL bWinStyleDIB;         // flag which signifies whether this is a Win3.0 DIB
    BOOL bResult = FALSE;

    /* if handle to DIB is invalid, return FALSE */

    if (hDIB == NULL)
        return FALSE;

    lpbi = (LPSTR) ::GlobalLock((HGLOBAL) hDIB);

    /* get pointer to BITMAPINFO (Win 3.0) */
    lpbmi = (LPBITMAPINFO)lpbi;

    /* get pointer to BITMAPCOREINFO (old 1.x) */
    lpbmc = (LPBITMAPCOREINFO)lpbi;
}

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีโทษตามกฎหมายที่เกี่ยวข้องอย่างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
wNumColors = ::DIBNumColors(lpbi);
```

```
if (wNumColors != 0)
```

```
{
```

```
    /* allocate memory block for logical palette */
```

```
    hLogPal = ::GlobalAlloc(GHND, sizeof(LOGPALETTE)
```

```
        + sizeof(PALETTEENTRY)
```

```
        * wNumColors);
```

```
    /* if not enough memory, clean up and return NULL */
```

```
    if (hLogPal == 0)
```

```
    {
```

```
        ::GlobalUnlock((HGLOBAL) hDIB);
```

```
        return FALSE;
```

```
    }
```

```
    lpPal = (LPLOGPALETTE) ::GlobalLock((HGLOBAL) hLogPal);
```

```
    /* set version and number of palette entries */
```

```
    lpPal->palVersion = PALVERSION;
```

```
    lpPal->palNumEntries = (WORD)wNumColors;
```

```
    /* is this a Win 3.0 DIB? */
```

```
    bWinStyleDIB = IS_WIN30_DIB(lpbi);
```

```
////////////////////////////////////
```

```
// my code
```

```
    CFile f;
```

```
    char buf[512];
```

```
    CFileException fe;
```

```
    if (!f.Open(path, CFile::modeRead | CFile::shareDenyWrite |
```

```
        CFile::shareDenyRead, &fe))
```

```
    {
```

```
        #ifdef _DEBUG
```

```
            if( fe.m_cause == CFileException::fileNotFound )
```

```
                afxDump << "ERROR: File not found\n";
```

```
        #endif
```

```
        return FALSE;
```

```
    }
```

```
    CArchive ar( &f, CArchive::load, 512, buf );
```

```
    COBList m_FlossList;
```

```
    POSITION m_position = NULL;
```

```
    LONG    m_lBlue;
```

```
    LONG    m_lGreen;
```

```
    LONG    m_lRed;
```

```
    CString m_number;
```

```
    CString m_name;
```

```
    BOOL    map, dup;
```

```
    int     red, green, blue;
```

```
    int     grow = 0;
```

```
    m_FlossList.Serialize(ar);
```

```
    m_position = m_FlossList.GetHeadPosition();
```

```
    if(m_position)
```

```
    {
```

```
        CFlossData* pFlossData = (CFlossData*) m_FlossList.GetAt(m_position);
```

```
        m_name = pFlossData->m_name;
```

```
        m_number = pFlossData->m_number;
```

```
        m_lBlue = pFlossData->m_lBlue;
```

```
        m_lGreen = pFlossData->m_lGreen;
```

```
        m_lRed = pFlossData->m_lRed;
```

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆก็ตามหากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
// end my code
/////////////////////////////////////////////////////////////////
    for (i = 0; i < (int)wNumColors; i++)
    {
        if (bWinStyleDIB)
        {
/////////////////////////////////////////////////////////////////
// my code
            red = lpbmi->bmiColors[i].rgbRed;
            green = lpbmi->bmiColors[i].rgbGreen;
            blue = lpbmi->bmiColors[i].rgbBlue;
            TRACE("%d\t%d\t%d = ", red , green, blue);

            map = FALSE;
            int period = 16;
            do
            {
                if(!m_FlossList.IsEmpty())
                {
                    m_position = m_FlossList.GetHeadPosition();
                    if(m_position)
                    {
                        CFlossData* pFlossData = (CFlossData*) m_FlossList.Ge
tAt(m_position);
                        //
                        //
                        //
                        //
                        TRACE("name = %s number = %s blue = %d green = %d red
", m_name, m_number, m_lBlue, m_lGreen, m_lRed
);
                    }
                }
            }
            do
            {
                if((red > m_lRed - period) && (red < m_lRed + period))
                    if((green > m_lGreen - period) && (green < m_lGreen +
period))
                        if((blue > m_lBlue - period) && (blue < m_lBlue +
period))
                            {
                                dup = FALSE;
                                lpbmi->bmiColors[i].rgbRed = (unsigned char)
                                lpbmi->bmiColors[i].rgbGreen = (unsigned char)
                                lpbmi->bmiColors[i].rgbBlue = (unsigned char)
                                map = TRUE;
                                for(int i=0; i<=m_FlossArray->GetUpperBound()
; i++)
                                {
                                    CFlossData* pFloss = (CFlossData*) m_Flos
sArray->GetAt(i);
                                    if((m_lRed==pFloss->m_lRed) &&
                                        (m_lGreen==pFloss->m_lGreen) &&
                                        (m_lBlue==pFloss->m_lBlue))
                                        dup = TRUE;
                                }
                            }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sList.GetAt(m_position);
                                CFlossData* pFloss = (CFlossData*) m_Flos
                                m_FlossArray->SetAtGrow(grow++, pFloss);
                                }
                                TRACE("OK blue = %d \tgreen = %d \tred = %d\n", m
                                _lBlue, m_lGreen, m_lRed);
                                }
                                if(m_position != NULL)
                                {
                                    m_FlossList.GetNext(m_position);
                                    if(m_position)
                                    {
                                        CFlossData* pFlossData = (CFlossData*) m_FlossLis
                                        t.GetAt(m_position);
                                        //
                                        //
                                        m_name = pFlossData->m_name;
                                        m_number = pFlossData->m_number;
                                        m_lBlue = pFlossData->m_lBlue;
                                        m_lGreen = pFlossData->m_lGreen;
                                        m_lRed = pFlossData->m_lRed;
                                        TRACE("name = %s number = %s blue = %d green = %d
                                        , m_name, m_number, m_lBlue, m_lGreen, m_
                                        lRed);
                                    }
                                }
                                }
                                while(!(map) && (m_position != NULL));
                                period += 16;
                                }while(!(map));

// end my code
//////////////////////////////////////
lpPal->palPalEntry[i].peRed = lpbmi->bmiColors[i].rgbRed;
lpPal->palPalEntry[i].peGreen = lpbmi->bmiColors[i].rgbGreen;
lpPal->palPalEntry[i].peBlue = lpbmi->bmiColors[i].rgbBlue;
lpPal->palPalEntry[i].peFlags = 0;
TRACE("%d\t%d\t%d\n", lpbmi->bmiColors[i].rgbRed,
lpbmi->bmiColors[i].rgbGreen, lpbmi->bmiColors[i].rgbBlue);
}
else
{
lpPal->palPalEntry[i].peRed = lpbmc->bmciColors[i].rgbtRed;
lpPal->palPalEntry[i].peGreen = lpbmc->bmciColors[i].rgbtGreen;
lpPal->palPalEntry[i].peBlue = lpbmc->bmciColors[i].rgbtBlue;
lpPal->palPalEntry[i].peFlags = 0;
}
}

/* create the palette and get handle to it */
bResult = pPal->CreatePalette(lpPal);
::GlobalUnlock((HGLOBAL) hLogPal);
::GlobalFree((HGLOBAL) hLogPal);
}

::GlobalUnlock((HGLOBAL) hDIB);

return bResult;
}

```

```

/*****
*
* FindDIBBits() ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
*
* Parameter: ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* LPSTR lpbi - pointer to packed-DIB memory block

*

* Return Value:

*

* LPSTR - pointer to the DIB bits

*

* Description:

*

* This function calculates the address of the DIB's bits and returns a pointer to the DIB bits.

*

*****/

```
LPSTR WINAPI FindDIBBits(LPSTR lpbi)
```

```
{  
    return (lpbi + *(LPDWORD)lpbi + ::PaletteSize(lpbi));  
}
```

*****/

* DIBWidth()

*

* Parameter:

*

* LPSTR lpbi - pointer to packed-DIB memory block

*

* Return Value:

*

* DWORD - width of the DIB

*

* Description:

*

* This function gets the width of the DIB from the BITMAPINFOHEADER

* width field if it is a Windows 3.0-style DIB or from the BITMAPCOREHEADER

* width field if it is an other-style DIB.

*

*****/

```
DWORD WINAPI DIBWidth(LPSTR lpDIB)
```

```
{  
    LPBITMAPINFOHEADER lpbmi; // pointer to a Win 3.0-style DIB  
    LPBITMAPCOREHEADER lpbmc; // pointer to an other-style DIB
```

```
/* point to the header (whether Win 3.0 and old) */
```

```
lpbmi = (LPBITMAPINFOHEADER)lpDIB;  
lpbmc = (LPBITMAPCOREHEADER)lpDIB;
```

```
/* return the DIB width if it is a Win 3.0 DIB */
```

```
if (IS_WIN30_DIB(lpDIB))
```

```
    return lpbmi->biWidth;
```

```
else /* it is an other-style DIB, so return its width */
```

```
    return (DWORD)lpbmc->bcWidth;
```

```
}
```

*****/

*เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

* DIBHeight()

* ไม่วาทกรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* Parameter:
*
* LPSTR lpbi      - pointer to packed-DIB memory block
*
* Return Value:
*
* DWORD          - height of the DIB
*
* Description:
*
* This function gets the height of the DIB from the BITMAPINFOHEADER
* height field if it is a Windows 3.0-style DIB or from the BITMAPCOREHEADER
* height field if it is an other-style DIB.
*
*****/

```

```

DWORD WINAPI DIBHeight(LPSTR lpDIB)

```

```

{
    LPBITMAPINFOHEADER lpbmi; // pointer to a Win 3.0-style DIB
    LPBITMAPCOREHEADER lpbmc; // pointer to an other-style DIB

    /* point to the header (whether old or Win 3.0) */

    lpbmi = (LPBITMAPINFOHEADER)lpDIB;
    lpbmc = (LPBITMAPCOREHEADER)lpDIB;

    /* return the DIB height if it is a Win 3.0 DIB */
    if (IS_WIN30_DIB(lpDIB))
        return lpbmi->biHeight;
    else /* it is an other-style DIB, so return its height */
        return (DWORD)lpbmc->bcHeight;
}

```

```

/*****
*
* PaletteSize()
*
* Parameter:
*
* LPSTR lpbi      - pointer to packed-DIB memory block
*
* Return Value:
*
* WORD           - size of the color palette of the DIB
*
* Description:
*
* This function gets the size required to store the DIB's palette by
* multiplying the number of colors by the size of an RGBQUAD (for a
* Windows 3.0-style DIB) or by the size of an RGBTRIPLE (for an other-
* style DIB).
*
*****/

```

```

WORD WINAPI PaletteSize(LPSTR lpbi)

```

```

{
    /* calculate the size required by the palette */
    if (IS_WIN30_DIB (lpbi))
        return (WORD) ( (::DIBNumColors (lpbi) * sizeof (RGBQUAD));
    else
        return (WORD) ( (::DIBNumColors (lpbi) * sizeof (RGBTRIPLE));
}

```

```

/*****
*
* DIBNumColors()
*
* Parameter:
*
* LPSTR lpbi      - pointer to packed-DIB memory block
*
* Return Value:
*
* WORD            - number of colors in the color table
*
* Description:
*
* This function calculates the number of colors in the DIB's color table
* by finding the bits per pixel for the DIB (whether Win3.0 or other-style
* DIB). If bits per pixel is 1: colors=2, if 4: colors=16, if 8: colors=256,
* if 24, no colors in color table.
*
*****/

```

```

WORD WINAPI DIBNumColors(LPSTR lpbi)
{
    WORD wBitCount; // DIB bit count

    /* If this is a Windows-style DIB, the number of colors in the
     * color table can be less than the number of bits per pixel
     * allows for (i.e. lpbi->biClrUsed can be set to some value).
     * If this is the case, return the appropriate value.
     */

    if (IS_WIN30_DIB(lpbi))
    {
        DWORD dwClrUsed;

        dwClrUsed = ((LPBITMAPINFOHEADER) lpbi)->biClrUsed;
        if (dwClrUsed != 0)
            return (WORD)dwClrUsed;
    }

    /* Calculate the number of colors in the color table based on
     * the number of bits per pixel for the DIB.
     */
    if (IS_WIN30_DIB(lpbi))
        wBitCount = ((LPBITMAPINFOHEADER) lpbi)->biBitCount;
    else
        wBitCount = ((LPBITMAPCOREHEADER) lpbi)->bcBitCount;

    /* return number of colors based on bits per pixel */
    switch (wBitCount)
    {
        case 1:
            return 2;

        case 4:
            return 16;

        case 8:
            return 256;
    }
}

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
default:
    return 0;
```

```
}
}
```

```
////////////////////////////////////
//// Clipboard support
```

```
//-----
//
// Function: CopyHandle (from SDK DibView sample clipbrd.c)
//
// Purpose: Makes a copy of the given global memory block. Returns
// a handle to the new memory block (NULL on error).
//
// Routine stolen verbatim out of ShowDIB.
//
// Parms: h == Handle to global memory to duplicate.
//
// Returns: Handle to new global memory block.
//-----
```

```
HGLOBAL WINAPI CopyHandle (HGLOBAL h)
{
    if (h == NULL)
        return NULL;

    DWORD dwLen = ::GlobalSize((HGLOBAL) h);
    HGLOBAL hCopy = ::GlobalAlloc(GHND, dwLen);

    if (hCopy != NULL)
    {
        void* lpCopy = ::GlobalLock((HGLOBAL) hCopy);
        void* lp = ::GlobalLock((HGLOBAL) h);
        memcpy(lpCopy, lp, dwLen);
        ::GlobalUnlock(hCopy);
        ::GlobalUnlock(h);
    }

    return hCopy;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// myfile.cpp
//
// Source file for Device-Independent Bitmap (DIB) API. Provides
// the following functions:
//
// SaveDIB()           - Saves the specified dib in a file
// ReadDIBFile()       - Loads a DIB from a file
//
//
// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (C) 1992 Microsoft Corporation
// All rights reserved.
//
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and Microsoft
// QuickHelp and/or WinHelp documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.

```

```

#include "stdafx.h"
#include <math.h>
#include <io.h>
#include <direct.h>
#include "dibapi.h"

```

```

/*
 * Dib Header Marker - used in writing DIBs to files
 */
#define DIB_HEADER_MARKER ((WORD) ('M' << 8) | 'B')

#ifdef _MAC
#define SWAPWORD(x) MAKEWORD(HIBYTE(x), LOBYTE(x))
#define SWAPLONG(x) MAKELONG(SWAPWORD(HIWORD(x)), SWAPWORD(LOWORD(x)))
void ByteSwapHeader(BITMAPFILEHEADER* bmiHeader);
void ByteSwapInfo(LPSTR lpHeader, BOOL fWin30Header);
#endif

```

```

/*****
 *
 * SaveDIB()
 *
 * Saves the specified DIB into the specified CFile. The CFile
 * is opened and closed by the caller.
 *
 * Parameters:
 *
 * HDIB hDib - Handle to the dib to save
 *
 * CFile& file - open CFile used to save DIB
 *
 * Return value: TRUE if successful, else FALSE or CFileException
 *
 *****/

```

```

BOOL WINAPI SaveDIB(HDIB hDib, CFile& file)
{
    BITMAPFILEHEADER bmfHdr; // Header for Bitmap file
    LPBITMAPINFOHEADER lpBI; // Pointer to DIB info structure
    DWORD dwDIBSize;

```

เอกสารนี้เป็นเอกสารที่... สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
 * Get a pointer to the DIB memory, the first of which contains
 * a BITMAPINFO structure
 */
lpBI = (LPBITMAPINFOHEADER) ::GlobalLock((HGLOBAL) hDib);
if (lpBI == NULL)
    return FALSE;

if (!IS_WIN30_DIB(lpBI))
{
    ::GlobalUnlock((HGLOBAL) hDib);
    return FALSE;    // It's an other-style DIB (save not supported)
}

/*
 * Fill in the fields of the file header
 */

/* Fill in file type (first 2 bytes must be "BM" for a bitmap) */
bmfHdr.bfType = DIB_HEADER_MARKER;    // "BM"

// Calculating the size of the DIB is a bit tricky (if we want to
// do it right).  The easiest way to do this is to call GlobalSize()
// on our global handle, but since the size of our global memory may have
// been padded a few bytes, we may end up writing out a few too
// many bytes to the file (which may cause problems with some apps).
//
// So, instead let's calculate the size manually (if we can)
//
// First, find size of header plus size of color table.  Since the
// first DWORD in both BITMAPINFOHEADER and BITMAPCOREHEADER contains
// the size of the structure, let's use this.

dwDIBSize = *(LPDWORD)lpBI + ::PaletteSize((LPSTR)lpBI);    // Partial Calculat
ion

// Now calculate the size of the image
if ((lpBI->biCompression == BI_RLE8) || (lpBI->biCompression == BI_RLE4))
{
    // It's an RLE bitmap, we can't calculate size, so trust the
    // biSizeImage field

    dwDIBSize += lpBI->biSizeImage;
}
else
{
    DWORD dwBmBitsSize;    // Size of Bitmap Bits only

    // It's not RLE, so size is Width (DWORD aligned) * Height

    dwBmBitsSize = WIDTHBYTES((lpBI->biWidth)*((DWORD)lpBI->biBitCount)) * lp
BI->biHeight;

    dwDIBSize += dwBmBitsSize;

    // Now, since we have calculated the correct size, why don't we
    // fill in the biSizeImage field (this will fix any .BMP files which
    // have this field incorrect).

    lpBI->biSizeImage = dwBmBitsSize;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Calculate the file size by adding the DIB size to sizeof(BITMAPFILEHEADER)

bmfHdr.bfSize = dwDIBSize + sizeof(BITMAPFILEHEADER);
bmfHdr.bfReserved1 = 0;
bmfHdr.bfReserved2 = 0;

/*
 * Now, calculate the offset the actual bitmap bits will be in
 * the file -- It's the Bitmap file header plus the DIB header,
 * plus the size of the color table.
 */
bmfHdr.bfOffBits = (DWORD)sizeof(BITMAPFILEHEADER) + lpBI->biSize
                  + PaletteSize((LPSTR)lpBI);

#ifdef _MAC
    ByteSwapHeader(&bmfHdr);

    // First swap the size field
    *((LPDWORD)lpBI) = SWAPLONG(*((LPDWORD)lpBI));

    // Now swap the rest of the structure (we don't save < Win30 files)
    ByteSwapInfo((LPSTR)lpBI, TRUE);
#endif

TRY
{
    // Write the file header
    file.Write((LPSTR)&bmfHdr, sizeof(BITMAPFILEHEADER));
    //
    // Write the DIB header and the bits
    //
    file.WriteHuge(lpBI, dwDIBSize);
}
CATCH (CFileException, e)
{
#ifdef _MAC
    // Swap everything back
    *((LPDWORD)lpBI) = SWAPLONG(*((LPDWORD)lpBI));
    ByteSwapInfo((LPSTR)lpBI, TRUE);
#endif

    ::GlobalUnlock((HGLOBAL) hDib);
    THROW_LAST();
}
END_CATCH

#ifdef _MAC
    // Swap everything back
    *((LPDWORD)lpBI) = SWAPLONG(*((LPDWORD)lpBI));
    ByteSwapInfo((LPSTR)lpBI, TRUE);
#endif

    ::GlobalUnlock((HGLOBAL) hDib);
    return TRUE;
}

```

/******

Function: ReadDIBFile (CFile&)

Purpose: Reads in the specified DIB file into a global chunk of memory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Returns: A handle to a dib (hDIB) if successful.
 NULL if an error occurs.

Comments: BITMAPFILEHEADER is stripped off of the DIB. Everything from the end of the BITMAPFILEHEADER structure on is returned in the global memory handle.

*****/

```
HDIB WINAPI ReadDIBfile(CFile& file)
{
    BITMAPFILEHEADER bmfHeader;
    DWORD dwBitsSize;
    HDIB hDIB;
    LPSTR pDIB;

    /*
     * get length of DIB in bytes for use when reading
     */

    dwBitsSize = file.GetLength();

    /*
     * Go read the DIB file header and check if it's valid.
     */
    if (file.Read((LPSTR)&bmfHeader, sizeof(bmfHeader)) != sizeof(bmfHeader))
        return NULL;

#ifdef _MAC
    ByteSwapHeader(&bmfHeader);
#endif
    if (bmfHeader.bfType != DIB_HEADER_MARKER)
        return NULL;

    /*
     * Allocate memory for DIB
     */
    hDIB = (HDIB) ::GlobalAlloc(GMEM_MOVEABLE | GMEM_ZEROINIT, dwBitsSize);
    if (hDIB == 0)
    {
        return NULL;
    }
    pDIB = (LPSTR) ::GlobalLock((HGLOBAL) hDIB);

    /*
     * Go read the bits.
     */
    if (file.ReadHuge(pDIB, dwBitsSize - sizeof(BITMAPFILEHEADER)) !=
        dwBitsSize - sizeof(BITMAPFILEHEADER) )
    {
        ::GlobalUnlock((HGLOBAL) hDIB);
        ::GlobalFree((HGLOBAL) hDIB);
        return NULL;
    }

#ifdef _MAC
    // First swap the size field
    *((LPDWORD)pDIB) = SWAPLONG(*((LPDWORD)pDIB));

    // Now swap the rest of the structure
    ByteSwapInfo(pDIB, IS_WIN30_DIB(pDIB));
#endif
    ::GlobalUnlock((HGLOBAL) hDIB);
    return hDIB;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#ifdef _MAC
void ByteSwapHeader(BITMAPFILEHEADER* bmfHeader)
{
    bmfHeader->bfType = SWAPWORD(bmfHeader->bfType);
    bmfHeader->bfSize = SWAPLONG(bmfHeader->bfSize);
    bmfHeader->bOffBits = SWAPLONG(bmfHeader->bOffBits);
}

void ByteSwapInfo(LPSTR lpHeader, BOOL fWin30Header)
{
    // Note this doesn't swap the bcSize/biSize field. It assumes that the
    // size field was swapped during read or while setting the fWin30Header
    // flag.

    if (fWin30Header)
    {
        LPBITMAPINFOHEADER lpBMITH = &(LPBITMAPINFO(lpHeader)->bmiHeader);

        //lpBMITH->biSize = SWAPLONG(lpBMITH->biSize);
        lpBMITH->biWidth = SWAPLONG(lpBMITH->biWidth);
        lpBMITH->biHeight = SWAPLONG(lpBMITH->biHeight);
        lpBMITH->biPlanes = SWAPWORD(lpBMITH->biPlanes);
        lpBMITH->biBitCount = SWAPWORD(lpBMITH->biBitCount);
        lpBMITH->biCompression = SWAPLONG(lpBMITH->biCompression);
        lpBMITH->biSizeImage = SWAPLONG(lpBMITH->biSizeImage);
        lpBMITH->biXPelsPerMeter = SWAPLONG(lpBMITH->biXPelsPerMeter);
        lpBMITH->biYPelsPerMeter = SWAPLONG(lpBMITH->biYPelsPerMeter);
        lpBMITH->biClrUsed = SWAPLONG(lpBMITH->biClrUsed);
        lpBMITH->biClrImportant = SWAPLONG(lpBMITH->biClrImportant);
    }
    else
    {
        LPBITMAPCOREHEADER lpBMCH = &(LPBITMAPCOREINFO(lpHeader)->bmciHeader);

        lpBMCH->bcWidth = SWAPWORD(lpBMCH->bcWidth);
        lpBMCH->bcHeight = SWAPWORD(lpBMCH->bcHeight);
        lpBMCH->bcPlanes = SWAPWORD(lpBMCH->bcPlanes);
        lpBMCH->bcBitCount = SWAPWORD(lpBMCH->bcBitCount);
    }
}

#endif

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// FlossData.h
```

```
class CFlossData : public CObject
```

```
{
```

```
    DECLARE_SERIAL(CFlossData)
```

```
public:
```

```
    LONG    m_lBlue;
```

```
    LONG    m_lGreen;
```

```
    LONG    m_lRed;
```

```
    CString m_number;
```

```
    CString m_name;
```

```
CFlossData()
```

```
{
```

```
    m_lBlue = m_lGreen = m_lRed = 0;
```

```
}
```

```
CFlossData(const char* szNumber, const char* szName, int lBlue,  
            int lGreen, int lRed) : m_number(szNumber)
```

```
{
```

```
    m_name = szName;
```

```
    m_lBlue = lBlue;
```

```
    m_lGreen = lGreen;
```

```
    m_lRed = lRed;
```

```
}
```

```
CFlossData(const CFlossData& s) : m_number(s.m_number)
```

```
{
```

```
    m_name = s.m_name;
```

```
    m_lBlue = s.m_lBlue;
```

```
    m_lGreen = s.m_lGreen;
```

```
    m_lRed = s.m_lRed;
```

```
}
```

```
const CFlossData& operator = (const CFlossData& s)
```

```
{
```

```
    m_number = s.m_number;
```

```
    m_name = s.m_name;
```

```
    m_lBlue = s.m_lBlue;
```

```
    m_lGreen = s.m_lGreen;
```

```
    m_lRed = s.m_lRed;
```

```
    return *this;
```

```
}
```

```
BOOL operator == (const CFlossData& s) const
```

```
{
```

```
    if((m_number == s.m_number) && (m_name == s.m_name) &&  
        (m_lBlue == s.m_lBlue) && (m_lGreen == s.m_lGreen) &&  
        (m_lRed == s.m_lRed))
```

```
    {
```

```
        return TRUE;
```

```
    }
```

```
    else
```

```
    {
```

```
        return FALSE;
```

```
    }
```

```
}
```

```
BOOL operator != (const CFlossData& s) const
```

```
{
```

```
    // let's make use of the operator we just defined
```

```
    return !(*this == s);
```

```
}
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
virtual void Serialize(CArchive& ar);
```

```
#ifdef _DEBUG  
    void Dump(CDumpContext& dc) const;  
#endif  
};
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "stdafx.h"
#include "FlossData.h"
```

```
IMPLEMENT_SERIAL(CFlossData, CObject, 0)
```

```
void CFlossData::Serialize(CArchive& ar)
{
    TRACE("Entering CFlossData::Serialize\n");
    if(ar.IsStoring())
    {
        ar << m_lBlue << m_lGreen << m_lRed << m_number << m_name;
    }
    else
    {
        ar >> m_lBlue >> m_lGreen >> m_lRed >> m_number >> m_name;
    }
}

#ifdef _DEBUG
void CFlossData::Dump(CDumpContext& dc) const
{
    CObject::Dump(dc);
    dc << "\nm_number = " << m_number << "\nm_name = " << m_name
        << "\nm_lBlue = " << m_lBlue << "\nm_lGreen = " << m_lGreen
        << "\nm_lRed = " << m_lRed;
}
#endif
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// MainFrm.h : interface of the CMainFrame class
//
/////////////////////////////////////////////////////////////////

class CMainFrame : public CFrameWnd
{
protected: // create from serialization only
    afx_msg BOOL OnQueryNewPalette();
    afx_msg void OnPaletteChanged(CWnd* pFocusWnd);
    CPalette* SetPalette(CPalette* pPalette);
    CPalette* m_pPalette;
    ///////////////////////////////////////////////////////////////////
    CSplitterWnd m_wndSplitter;
    CDialogBar m_wndPaletteDialogBar;
    CMainFrame();
    DECLARE_DYNCREATE(CMainFrame)

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CMainFrame)
    public:
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    protected:
    virtual BOOL OnCreateClient(LPCREATESTRUCT lpcs, CCreateContext* pContext);
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected: // control bar embedded members
    CStatusBar m_wndStatusBar;
    CToolBar m_wndToolBar;
    // CDialogBar m_wndPaletteDialogBar;

// Generated message map functions
protected:
    //{{AFX_MSG(CMainFrame)
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
        // NOTE - the ClassWizard will add and remove member functions here.
        // DO NOT EDIT what you see in these blocks of generated code!
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// MainFrm.cpp : implementation of the CMainFrame class
//
```

```
#include "stdafx.h"
#include "xstitch.h"
#include "xstitchDoc.h"
#include "xstitchView.h"
#include "SymbolView.h"
#include "MainFrm.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CMainFrame
```

```
IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)
```

```
BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
```

```
ON_WM_QUERYNEWPALETTE()
ON_WM_PALETTECHANGED()
ON_COMMAND_EX(CG_IDD_PALETTEDIALOGBAR, OnBarCheck)
ON_UPDATE_COMMAND_UI(CG_IDD_PALETTEDIALOGBAR, OnUpdateControlBarMenu)
//{{AFX_MSG_MAP(CMainFrame)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code !
ON_WM_CREATE()
//}}AFX_MSG_MAP
// Global help commands
ON_COMMAND(ID_HELP_FINDER, CFrameWnd::OnHelpFinder)
ON_COMMAND(ID_HELP, CFrameWnd::OnHelp)
ON_COMMAND(ID_CONTEXT_HELP, CFrameWnd::OnContextHelp)
ON_COMMAND(ID_DEFAULT_HELP, CFrameWnd::OnHelpFinder)
```

```
END_MESSAGE_MAP()
```

```
static UINT indicators[] =
{
    ID_SEPARATOR,           // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};
```

```
////////////////////////////////////
// CMainFrame construction/destruction
```

```
CMainFrame::CMainFrame()
```

```
{
    // TODO: add member initialization code here
}
```

```
CMainFrame::~CMainFrame()
```

```
{
}
```

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
```

```
{
    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    // CG: This line was added by the Palette Support component
    m_pPalette = NULL; ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
```

```

return -1;

if (!m_wndToolBar.Create(this) ||
    !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
{
    TRACE0("Failed to create toolbar\n");
    return -1;    // fail to create
}

if (!m_wndStatusBar.Create(this) ||
    !m_wndStatusBar.SetIndicators(indicators,
    sizeof(indicators)/sizeof(UINT)))
{
    TRACE0("Failed to create status bar\n");
    return -1;    // fail to create
}

// TODO: Remove this if you don't want tool tips or a resizable toolbar
m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle() |
    CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC);

// TODO: Delete these three lines if you don't want the toolbar to
// be dockable
m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
EnableDocking(CBRS_ALIGN_ANY);
DockControlBar(&m_wndToolBar);

// TODO: Add a menu item that will toggle the visibility of the
// dialog bar named "Palette Dialog Bar":
// 1. In ResourceView, open the menu resource that is used by
// the CMainFrame class
// 2. Select the View submenu
// 3. Double-click on the blank item at the bottom of the submenu
// 4. Assign the new item an ID: CG_ID_VIEW_PALETTE_DIALOGBAR
// 5. Assign the item a Caption: Palette Dialog Bar

// TODO: Change the value of CG_ID_VIEW_PALETTE_DIALOGBAR to an appropriate va
// lue:
// 1. Open the file resource.h
// 2. Find the definition for the symbol CG_ID_VIEW_PALETTE_DIALOGBAR
// 3. Change the value of the symbol. Use a value in the range
// 0xE804 to 0xEB1A that is not already used by another symbol

// CG: The following block was inserted by the 'Dialog Bar' component
/*
{
    // Initialize dialog bar m_wndPaletteDialogBar
    if (!m_wndPaletteDialogBar.Create(this, CG_IDD_PALETTE_DIALOGBAR,
        CBRS_BOTTOM | CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_HIDE_INPLACE,
        CG_IDD_PALETTE_DIALOGBAR))
    {
        TRACE0("Failed to create dialog bar m_wndPaletteDialogBar\n");
        return -1;    // fail to create
    }

    m_wndPaletteDialogBar.EnableDocking(CBRS_ALIGN_TOP | CBRS_ALIGN_BOTTOM);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndPaletteDialogBar);
}
*/
return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)

```

```

{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CFrameWnd::PreCreateWindow(cs);
}

////////////////////////////////////
// CMainFrame diagnostics

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
    CFrameWnd::Dump(dc);
}

#endif //_DEBUG

////////////////////////////////////
// CMainFrame message handlers

BOOL CMainFrame::OnQueryNewPalette()
{
    // CG: This function was added by the Palette Support component

    if (m_pPalette == NULL)
        return FALSE;

    // BLOCK
    {
        CClientDC dc(this);
        CPalette* pOldPalette = dc.SelectPalette(m_pPalette,
            GetCurrentMessage()->message == WM_PALETTECHANGED);
        UINT nChanged = dc.RealizePalette();
        dc.SelectPalette(pOldPalette, TRUE);

        if (nChanged == 0)
            return FALSE;
    }

    Invalidate();

    return TRUE;
}

void CMainFrame::OnPaletteChanged(CWnd* pFocusWnd)
{
    // CG: This function was added by the Palette Support component

    if (pFocusWnd == this || IsChild(pFocusWnd))
        return;

    OnQueryNewPalette();
}

CPalette* CMainFrame::SetPalette(CPalette* pPalette)
{
    {
        เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
        // CG: This function was added by the Palette Support component
        ไม่วกรณใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    }
}

```

```
// Call this function when the palette changes. It will
// realize the palette in the foreground to cause the screen
// to repaint correctly. All calls to CDC::SelectPalette in
// painting code should select palettes in the background.
```

```
CPalette* pOldPalette = m_pPalette;
m_pPalette = pPalette;
OnQueryNewPalette();
return pOldPalette;
```

```
}
```

```
BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT lpcs, CCreateContext* pContext)
```

```
{
```

```
// TODO: Add your specialized code here and/or call the base class
BOOL rtn = m_wndSplitter.CreateStatic(this, 1, 2);
rtn |= m_wndSplitter.CreateView(0, 0, RUNTIME_CLASS(CSymbolView),
                                CSize(150, 200), pContext);
rtn |= m_wndSplitter.CreateView(0, 1, RUNTIME_CLASS(CXstitchView),
                                CSize(250, 200), pContext);
```

```
return rtn;
```

```
// return CFrameWnd::OnCreateClient(lpcs, pContext);
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// xstitch.h : main header file for the XSTITCH application
//

#ifndef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h"          // main symbols

#define WM_DOREALIZE          (WM_USER + 0)

/////////////////////////////////////////////////////////////////
// CXstitchApp:
// See xstitch.cpp for the implementation of this class
//

class CXstitchApp : public CWinApp
{
public:
    CXstitchApp();

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CXstitchApp)
public:
    virtual BOOL InitInstance();
//}}AFX_VIRTUAL

// Implementation

//{{AFX_MSG(CXstitchApp)
afx_msg void OnAppAbout();
// NOTE - the ClassWizard will add and remove member functions here.
//      DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// xstitch.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "xstitch.h"

#include "MainFrm.h"
#include "xstitchDoc.h"
#include "xstitchView.h"
#include <dos.h>
#include <direct.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchApp

BEGIN_MESSAGE_MAP(CXstitchApp, CWinApp)
//{{AFX_MSG_MAP(CXstitchApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
// Standard file based document commands
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
// Standard print setup command
ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchApp construction

CXstitchApp::CXstitchApp()
{
// TODO: add construction code here,
// Place all significant initialization in InitInstance
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// The one and only CXstitchApp object

CXstitchApp theApp;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchApp initialization

BOOL CXstitchApp::InitInstance()
{
// Standard initialization
// If you are not using these features and wish to reduce the size
// of your final executable, you should remove from the following
// the specific initialization routines you do not need.

#ifdef _AFXDLL
Enable3dControls(); // Call this when using MFC in a shared DLL
#else
Enable3dControlsStatic(); // Call this when linking to MFC statically
#endif
}

```

```

LoadStdProfileSettings(); // Load standard INI file options (including MRU)

// Register the application's document templates. Document templates
// serve as the connection between documents, frame windows and views.

CSingleDocTemplate* pDocTemplate;
pDocTemplate = new CSingleDocTemplate(
    IDR_MAINFRAME,
    RUNTIME_CLASS(CXstitchDoc),
    RUNTIME_CLASS(CMainFrame), // main SDI frame window
    RUNTIME_CLASS(CXstitchView));
AddDocTemplate(pDocTemplate);

// Parse command line for standard shell commands, DDE, file open
CCommandLineInfo cmdInfo;
ParseCommandLine(cmdInfo);

// Dispatch commands specified on the command line
if (!ProcessShellCommand(cmdInfo))
    return FALSE;

return TRUE;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
virtual BOOL OnInitDialog();
//{{AFX_MSG(CAboutDlg)
// No message handlers
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
//{{AFX_DATA_INIT(CAboutDlg)
//}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
//{{AFX_DATA_MAP(CAboutDlg)
//}}AFX_DATA_MAP
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 หากมีข้อผิดพลาดประการใด กรุณาแจ้งให้ทราบทันที ห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
   //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

// App command to run the dialog
void CXstitchApp::OnAppAbout()
{
    CAboutDlg aboutDlg;
    aboutDlg.DoModal();
}

////////////////////////////////////
// CXstitchApp commands

BOOL CAboutDlg::OnInitDialog()
{
    // CG: Folowing code is added by System Info Component
    {
        CDialog::OnInitDialog();
#ifdef _MAC
        CString strFreeDiskSpace;
        CString strFreeMemory;
        CString strFmt;

        // Fill available memory
        MEMORYSTATUS MemStat;
        MemStat.dwLength = sizeof(MEMORYSTATUS);
        GlobalMemoryStatus(&MemStat);
        strFmt.LoadString(CG_IDS_PHYSICAL_MEM);
        strFreeMemory.Format(strFmt, MemStat.dwTotalPhys / 1024L);

        //TODO: Add a static control to your About Box to receive the memory
        // information. Initialize the control with code like this:
        SetDlgItemText(IDC_PHYSICAL_MEM, strFreeMemory);

        // Fill disk free information
        struct _diskfree_t diskfree;
        int nDrive = _getdrive(); // use current default drive
        if (_getdiskfree(nDrive, &diskfree) == 0)
        {
            strFmt.LoadString(CG_IDS_DISK_SPACE);
            strFreeDiskSpace.Format(strFmt,
                (DWORD)diskfree.avail_clusters *
                (DWORD)diskfree.sectors_per_cluster *
                (DWORD)diskfree.bytes_per_sector / (DWORD)1024L,
                nDrive-1 + _T('A'));
        }
        else
            strFreeDiskSpace.LoadString(CG_IDS_DISK_SPACE_UNAVAIL);

        //TODO: Add a static control to your About Box to receive the memory
        // information. Initialize the control with code like this:
        SetDlgItemText(IDC_DISK_SPACE, strFreeDiskSpace);
#endif
    }
    return TRUE;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// xstitchDoc.h : interface of the CXstitchDoc class
//
////////////////////////////////////////////////////////////////////

#include "dibapi.h"

class CXstitchDoc : public CDocument
{
    friend class CXstitchView;
    friend class CSymbolView;
public:
    double m_nPZoomSize;
    CObArray m_FlossArray;
    CString path;

protected: // create from serialization only
    CXstitchDoc();
    DECLARE_DYNCREATE(CXstitchDoc)

// Attributes
public:
    HDIB GetHDIB() const
        { return m_hDIB; }
    CPalette* GetDocPalette() const
        { return m_palDIB; }
    CSize GetDocSize() const
        { return m_sizeDoc; }
    int GetArraySize() const
        { return m_FlossArray.GetUpperBound(); }

// Operations
public:
    void ReplaceHDIB(HDIB hDIB);
    void InitDIBData();

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CXstitchDoc)
public:
    virtual void Serialize(CArchive& ar);
    virtual BOOL OnOpenDocument(LPCTSTR lpszPathName);
    virtual BOOL OnSaveDocument(LPCTSTR lpszPathName);
    virtual void DeleteContents();
    virtual BOOL OnNewDocument();
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CXstitchDoc();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:
    HDIB m_hDIB;
    CPalette* m_palDIB;
    CSize m_sizeDoc;

// Generated message map functions
protected:
    //{{AFX_MSG(CXstitchDoc)
    afx_msg void OnFlossSymbols();
    afx_msg void OnFlossDmc();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
}

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

```

////////////////////////////////////



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// xstitchDoc.cpp : implementation of the CXstitchDoc class
//
```

```
#include "stdafx.h"
#include "xstitch.h"
```

```
#include "dibapi.h"
#include "xstitchDoc.h"
#include "ImportDlg.h"
#include "PaletteDlgBar.h"
#include "FlossTabDlg.h"
#include "stdlib.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CXstitchDoc
```

```
IMPLEMENT_DYNCREATE(CXstitchDoc, CDocument)
```

```
BEGIN_MESSAGE_MAP(CXstitchDoc, CDocument)
    //{{AFX_MSG_MAP(CXstitchDoc)
    ON_COMMAND(ID_FLOSS_SYMBOLS, OnFlossSymbols)
    ON_COMMAND(ID_FLOSS_DMC, OnFlossDmc)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CXstitchDoc construction/destruction
```

```
CXstitchDoc::CXstitchDoc()
{
    // TODO: add one-time construction code here
    m_hDIB = NULL;
    m_palDIB = NULL;
    m_sizeDoc = CSize(1,1); // dummy value to make CScrollView happy
    m_nPZoomSize = 1.0;
    // m_FlossArray.SetSize(0, 5);

    CFile fi;
    fi.Open("Dmc.mai", CFile::modeRead);
    path = fi.GetFilePath();
    TRACE("path = %s", path);
    fi.Close();
}
```

```
CXstitchDoc::~CXstitchDoc()
{
    if (m_hDIB != NULL)
    {
        ::GlobalFree((HGLOBAL) m_hDIB);
    }
    if (m_palDIB != NULL)
    {
        delete m_palDIB;
    }
}
```

```
} เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
BOOL CXstitchDoc::OnNewDocument()
{
    ไม่สามารถนำข้อมูลนี้ไปใช้เพื่อวัตถุประสงค์อื่นใดได้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
}
```

```

if (!CDocument::OnNewDocument())
    return FALSE;

// TODO: add reinitialization code here
// (SDI documents will reuse this document)
return TRUE;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchDoc serialization

void CXstitchDoc::Serialize(CArchive& ar)
{
    if (ar.IsStoring())
    {
        // TODO: add storing code here
    }
    else
    {
        // TODO: add loading code here
    }
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchDoc diagnostics

#ifdef _DEBUG
void CXstitchDoc::AssertValid() const
{
    CDocument::AssertValid();
}

void CXstitchDoc::Dump(CDumpContext& dc) const
{
    CDocument::Dump(dc);
}
#endif // _DEBUG

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchDoc commands

void CXstitchDoc::OnFlossSymbols()
{
    // TODO: Add your command handler code here
    CFontDialog FontDialog;
    if (FontDialog.DoModal() != IDOK)
        return;
    UpdateAllViews(NULL);
}

void CXstitchDoc::OnFlossDmc()
{
    // TODO: Add your command handler code here
    CFlossTabDlg FlossTabDlg;
    if (FlossTabDlg.DoModal() != IDOK)
        return;
    UpdateAllViews(NULL);
}

void CXstitchDoc::InitDIBData()
{
    if (m_palDIB != NULL)
    {
        if (m_palDIB != NULL)
        {
            // TODO: Add your command handler code here
            m_palDIB->SetColorTable(0, 0);
        }
    }
}

```

งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delete m_palDIB;
    m_palDIB = NULL;
}
if (m_hDIB == NULL)
{
    return;
}
// Set up document size
LPSTR lpDIB = (LPSTR) ::GlobalLock((HGLOBAL) m_hDIB);
if (::DIBWidth(lpDIB) > INT_MAX || ::DIBHeight(lpDIB) > INT_MAX)
{
    ::GlobalUnlock((HGLOBAL) m_hDIB);
    ::GlobalFree((HGLOBAL) m_hDIB);
    m_hDIB = NULL;
    CString strMsg;
    strMsg.LoadString(IDS_DIB_TOO_BIG);
    MessageBox(NULL, strMsg, NULL, MB_ICONINFORMATION | MB_OK);
    return;
}

ImportDlg Dlg;

Dlg.m_DesignFloss    = "DMC";
Dlg.m_DesignScale    = 1;
Dlg.m_DesignPalette  = 64;
if(Dlg.DoModal() != IDOK)
{
    return;
}
m_sizeDoc = CSize((int) (::DIBWidth(lpDIB) * 12 * m_nPZoomSize),
                  (int) (::DIBHeight(lpDIB) * 12 * m_nPZoomSize));

UpdateAllViews(NULL);

::GlobalUnlock((HGLOBAL) m_hDIB);
// Create copy of palette
m_palDIB = new CPalette;
if (m_palDIB == NULL)
{
    // we must be really low on memory
    ::GlobalFree((HGLOBAL) m_hDIB);
    m_hDIB = NULL;
    return;
}
BeginWaitCursor();
if (::CreateDIBPalette(m_hDIB, m_palDIB, &m_FlossArray, path) == NULL)
{
    // DIB may not have a palette
    delete m_palDIB;
    m_palDIB = NULL;
    return;
}
EndWaitCursor();
}

BOOL CXstitchDoc::OnOpenDocument(LPCTSTR lpszPathName)
{
    if (!CDocument::OnOpenDocument(lpszPathName))
        return FALSE;

    // TODO: Add your specialized creation code here
    CFile file;
    CFileException fe; ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    if (!file.Open(lpszPathName, CFile::modeRead | CFile::shareDenyWrite, &fe))

```

```

{
    ReportSaveLoadException(lpszPathName, &fe,
        FALSE, AFX_IDP_FAILED_TO_OPEN_DOC);
    return FALSE;
}

DeleteContents();
BeginWaitCursor();

m_hDIB = ::ReadDIBFile(file);
// replace calls to Serialize with ReadDIBFile function
/*
TRY
{
    m_hDIB = ::ReadDIBFile(file);
}
CATCH (CFileException, eLoad)
{
    file.Abort(); // will not throw an exception
    EndWaitCursor();
    ReportSaveLoadException(lpszPathName, eLoad,
        FALSE, AFX_IDP_FAILED_TO_OPEN_DOC);
    m_hDIB = NULL;
    return FALSE;
}
END_CATCH
*/

InitDIBData();
EndWaitCursor();

if (m_hDIB == NULL)
{
    // may not be DIB format
    CString strMsg;
    strMsg.LoadString(IDS_CANNOT_LOAD_DIB);
    MessageBox(NULL, strMsg, NULL, MB_ICONINFORMATION | MB_OK);
    return FALSE;
}
SetPathName(lpszPathName);
SetModifiedFlag(FALSE); // start off with unmodified

return TRUE;
}

BOOL CXstitchDoc::OnSaveDocument(LPCTSTR lpszPathName)
{
/*
// TODO: Add your specialized code here and/or call the base class
CFile file;
CFileException fe;

if (!file.Open(lpszPathName, CFile::modeCreate |
    CFile::modeReadWrite | CFile::shareExclusive, &fe))
{
    ReportSaveLoadException(lpszPathName, &fe,
        TRUE, AFX_IDP_INVALID_FILENAME);
    return FALSE;
}

// replace calls to Serialize with SaveDIB function
BOOL bSuccess = FALSE;
TRY
{
    BeginWaitCursor();

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BeginWaitCursor();
```

```

        bSuccess = ::SaveDIB(m_hDIB, file);
        file.Close();
    }
    CATCH (CException, eSave)
    {
        file.Abort(); // will not throw an exception
        EndWaitCursor();
        ReportSaveLoadException(lpszPathName, eSave,
            TRUE, AFX_IDP_FAILED_TO_SAVE_DOC);
        return FALSE;
    }
    END_CATCH

    EndWaitCursor();
    SetModifiedFlag(FALSE); // back to unmodified

    if (!bSuccess)
    {
        // may be other-style DIB (load supported but not save)
        // or other problem in SaveDIB
        CString strMsg;
        strMsg.LoadString(IDS_CANNOT_SAVE_DIB);
        MessageBox(NULL, strMsg, NULL, MB_ICONINFORMATION | MB_OK);
    }

    return bSuccess;
*/
    return CDocument::OnSaveDocument(lpszPathName);
}

void CXstitchDoc::ReplaceHDIB(HDIB hDIB)
{
    if (m_hDIB != NULL)
    {
        ::GlobalFree((HGLOBAL) m_hDIB);
    }
    m_hDIB = hDIB;
}

void CXstitchDoc::DeleteContents()
{
    // TODO: Add your specialized code here and/or call the base class

    for(int i=m_FlossArray.GetUpperBound(); i >= 0; i--)
    {
        delete m_FlossArray.GetAt(i);
    }
    m_FlossArray.RemoveAll();

    CDocument::DeleteContents();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// xstitchView.h : interface of the CXstitchView class
//
/////////////////////////////////////////////////////////////////

class CXstitchView : public CScrollView
{
private:
    CDC*      m_pDisplayMemDC;
    HBITMAP  m_hOldDisplayBitmap;

    int m_nStitch;      // variable for check stitch
    int m_nCanvas;     // variable for check canvas
    int m_nZoom;       // variable for check zoom
    double m_nZoomSize; // value for multiple image size

    BITMAP BM, BMBLACK, BMX;
    void NewClientRect(); // Clear ViewPort
    // void ShowSymbol(CDC* pDC, int& nPosX, int& nPosY, int nPoints, char symbol);

protected:
    CBitmap m_Bitmap; // m_Bitmap is a canvas (bitmap object)
    int m_BitmapHeight; // dimensions of bitmap
    int m_BitmapWidth;
    int m_nDWidth, m_nDHeight; // dimensions of destination bitmap

    CBitmap m_MaskBitmap;
    int m_MaskBitmapHeight;
    int m_MaskBitmapWidth;
    CBitmap m_ImageBitmap;

    CDC ColorMemDC;
    CBitmap m_ColorBitmap;

protected: // create from serialization only
    CXstitchView();
    DECLARE_DYNCREATE(CXstitchView)

// Attributes
public:
    CXstitchDoc* GetDocument();

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CXstitchView)
public:
    virtual void OnDraw(CDC* pDC); // overridden to draw this view
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    virtual void OnPrepareDC(CDC* pDC, CPrintInfo* pInfo = NULL);
protected:
    virtual void OnInitialUpdate(); // called first time after construct
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnActivateView(BOOL bActivate, CView* pActivateView, CView* pDea
ctiveView);
    //}}AFX_VIRTUAL

    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
// Implementation
public:
    ภารกิจใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    virtual ~CXstitchView();
#ifdef _DEBUG

```

```

    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

```

```
protected:
```

```
// Generated message map functions
```

```
protected:
```

```

//{{AFX_MSG(CXstitchView)
afx_msg void OnStitchSolid();
afx_msg void OnUpdateStitchSolid(CCmdUI* pCmdUI);
afx_msg void OnStitchStitch();
afx_msg void OnUpdateStitchStitch(CCmdUI* pCmdUI);
afx_msg void OnStitchSymbols();
afx_msg void OnUpdateStitchSymbols(CCmdUI* pCmdUI);
afx_msg void OnViewSimple();
afx_msg void OnViewZoomIn();
afx_msg void OnViewZoomOut();
afx_msg void OnCanvasAida();
afx_msg void OnCanvasEverweave();
afx_msg void OnCanvasGrid();
afx_msg void OnCanvasNone();
afx_msg void OnCanvasPlastic();
afx_msg void OnCanvasRug();
afx_msg void OnCanvasTapestry();
afx_msg void OnUpdateCanvasAida(CCmdUI* pCmdUI);
afx_msg void OnUpdateCanvasEverweave(CCmdUI* pCmdUI);
afx_msg void OnUpdateCanvasGrid(CCmdUI* pCmdUI);
afx_msg void OnUpdateCanvasNone(CCmdUI* pCmdUI);
afx_msg void OnUpdateCanvasPlastic(CCmdUI* pCmdUI);
afx_msg void OnUpdateCanvasRug(CCmdUI* pCmdUI);
afx_msg void OnUpdateCanvasTapestry(CCmdUI* pCmdUI);
afx_msg LRESULT OnDoRealize(WPARAM wParam, LPARAM lParam); // user message
afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
afx_msg void OnStitchThin();
afx_msg void OnUpdateStitchThin(CCmdUI* pCmdUI);
//}}AFX_MSG
DECLARE_MESSAGE_MAP()

```

```
};
```

```
#ifndef _DEBUG // debug version in xstitchView.cpp
```

```
inline CXstitchDoc* CXstitchView::GetDocument()
```

```
{ return (CXstitchDoc*)m_pDocument; }
```

```
#endif
```

```
////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// xstitchView.cpp : implementation of the CXstitchView class
//
```

```
#include "stdafx.h"
#include "xstitch.h"
#include "FlossData.h"
#include "xstitchDoc.h"
#include "xstitchView.h"
#include "ImportDlg.h"
#include "dibapi.h"
#include "mainfrm.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CXstitchView
```

```
IMPLEMENT_DYNCREATE(CXstitchView, CScrollView)
```

```
BEGIN_MESSAGE_MAP(CXstitchView, CScrollView)
```

```
    //{{AFX_MSG_MAP(CXstitchView)
    ON_COMMAND(ID_STITCH_SOLID, OnStitchSolid)
    ON_UPDATE_COMMAND_UI(ID_STITCH_SOLID, OnUpdateStitchSolid)
    ON_COMMAND(ID_STITCH_STITCH, OnStitchStitch)
    ON_UPDATE_COMMAND_UI(ID_STITCH_STITCH, OnUpdateStitchStitch)
    ON_COMMAND(ID_STITCH_SYMBOLS, OnStitchSymbols)
    ON_UPDATE_COMMAND_UI(ID_STITCH_SYMBOLS, OnUpdateStitchSymbols)
    ON_COMMAND(ID_VIEW_SIMPLE, OnViewSimple)
    ON_COMMAND(ID_VIEW_ZOOM_IN, OnViewZoomIn)
    ON_COMMAND(ID_VIEW_ZOOM_OUT, OnViewZoomOut)
    ON_COMMAND(ID_CANVAS_AIDA, OnCanvasAida)
    ON_COMMAND(ID_CANVAS_EVERWEAVE, OnCanvasEverweave)
    ON_COMMAND(ID_CANVAS_GRID, OnCanvasGrid)
    ON_COMMAND(ID_CANVAS_NONE, OnCanvasNone)
    ON_COMMAND(ID_CANVAS_PLASTIC, OnCanvasPlastic)
    ON_COMMAND(ID_CANVAS_RUG, OnCanvasRug)
    ON_COMMAND(ID_CANVAS_TAPESTRY, OnCanvasTapestry)
    ON_UPDATE_COMMAND_UI(ID_CANVAS_AIDA, OnUpdateCanvasAida)
    ON_UPDATE_COMMAND_UI(ID_CANVAS_EVERWEAVE, OnUpdateCanvasEverweave)
    ON_UPDATE_COMMAND_UI(ID_CANVAS_GRID, OnUpdateCanvasGrid)
    ON_UPDATE_COMMAND_UI(ID_CANVAS_NONE, OnUpdateCanvasNone)
    ON_UPDATE_COMMAND_UI(ID_CANVAS_PLASTIC, OnUpdateCanvasPlastic)
    ON_UPDATE_COMMAND_UI(ID_CANVAS_RUG, OnUpdateCanvasRug)
    ON_UPDATE_COMMAND_UI(ID_CANVAS_TAPESTRY, OnUpdateCanvasTapestry)
    ON_WM_CREATE()
    ON_COMMAND(ID_STITCH_THIN, OnStitchThin)
    ON_UPDATE_COMMAND_UI(ID_STITCH_THIN, OnUpdateStitchThin)
    //}}AFX_MSG_MAP
    // Standard printing commands
    ON_COMMAND(ID_FILE_PRINT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, CScrollView::OnFilePrintPreview)
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CXstitchView construction/destruction
```

```
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
CXstitchView::CXstitchView()
{
    ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    // TODO: add construction code here
}
```

```

m_nZoom = 0;          // for check menu
m_nStitch = 2;
m_nCanvas = 5;
m_nZoomSize = 1.0;

m_Bitmap.LoadBitmap(IDB_GRIDX); // Load bitmap from resource
/*
m_MaskBitmap.LoadBitmap(IDB_X);
m_MaskBitmap.GetObject(sizeof(BMBLACK), &BMBLACK);
m_MaskBitmapWidth = BMBLACK.bmWidth;
m_MaskBitmapHeight = BMBLACK.bmHeight;
*/
m_pDisplayMemDC = new CDC;

m_ImageBitmap.LoadBitmap(IDB_XWHITE);
}

CXstitchView::~CXstitchView()
{
}

BOOL CXstitchView::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CScrollView::PreCreateWindow(cs);
}

////////////////////////////////////
// CXstitchView drawing

void CXstitchView::OnDraw(CDC* pDC)
{
    CXstitchDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    // TODO: add draw code for native data here
    HDIB hDIB = pDoc->GetHDIB();
    if (hDIB != NULL)
    {
        LPSTR lpDIB = (LPSTR) ::GlobalLock((HGLOBAL) hDIB);
        int cxDIB = (int) ::DIBWidth(lpDIB);          // Size of DIB - x
        int cyDIB = (int) ::DIBHeight(lpDIB);        // Size of DIB - y
        ::GlobalUnlock((HGLOBAL) hDIB);
        CRect rcDIB;
        rcDIB.top = rcDIB.left = 0;
        rcDIB.right = cxDIB;
        rcDIB.bottom = cyDIB;
        CRect rcDest;
        if (pDC->IsPrinting()) // printer DC
        {
            // get size of printer page (in pixels)
            int cxPage = pDC->GetDeviceCaps(HORZRES);
            int cyPage = pDC->GetDeviceCaps(VERTRES);
            // get printer pixels per inch
            int cxInch = pDC->GetDeviceCaps(LOGPIXELSX);
            int cyInch = pDC->GetDeviceCaps(LOGPIXELSY);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 // Best Fit case -- create a rectangle which preserves
 ไม่ว่าจะกรณีใดๆ ก็ตาม มิได้สงวนลิขสิทธิ์ และให้ผู้อื่นไปใช้
 // the DIB's aspect ratio, and fills the page horizontally
 //

```

// The formula in the "->bottom" field below calculates the Y
// position of the printed bitmap, based on the size of the
// bitmap, the width of the page, and the relative size of
// a printed pixel (cyInch / cxInch).
//
rcDest.top = rcDest.left = 0;
rcDest.bottom = (int)((double)cyDIB * cxPage * cyInch
    / ((double)cxDIB * cxInch));
rcDest.right = cxPage;
}
else // not printer DC
{
//
rcDest = rcDIB;
rcDest.top = rcDest.left = 0;
rcDest.bottom = (int)(rcDIB.bottom * 12 * m_nZoomSize);
rcDest.right = (int)(rcDIB.right * 12 * m_nZoomSize);
}

// show cross stitch
if((m_nStitch==0) || (m_nStitch==4))
{
// load background
m_Bitmap.GetObject(sizeof(BM), &BM);
m_BitmapWidth = BM.bmWidth;
m_BitmapHeight = BM.bmHeight;
CDC MemDC;
MemDC.CreateCompatibleDC(NULL);
MemDC.SelectObject(&m_Bitmap);
CRect rect(0, 0, m_BitmapWidth, m_BitmapHeight);
m_nDWidth = (int)(rect.Width() * m_nZoomSize);
m_nDHeight = (int)(rect.Height() * m_nZoomSize);

/*
BeginWaitCursor();
for(int row = 0; row < rcDest.bottom; row+=m_nDHeight)
    for(int col = 0; col < rcDest.right; col+=m_nDWidth)
        pDC->StretchBlt(col, row, m_nDWidth, m_nDHeight,
            &MemDC, 0, 0, m_BitmapWidth,
            m_BitmapHeight, SRCCOPY);
EndWaitCursor();
*/

m_MaskBitmap.DeleteObject();
if(m_nStitch==0)
    m_MaskBitmap.LoadBitmap(IDB_X);
else
    m_MaskBitmap.LoadBitmap(IDB_ANDX);

m_MaskBitmap.GetObject(sizeof(BMBLACK), &BMBLACK);
m_MaskBitmapWidth = BMBLACK.bmWidth;
m_MaskBitmapHeight = BMBLACK.bmHeight;

CDC MaskMemDC;
CDC ImageMemDC;

// create memory device context object and select bitmap object into
it
MaskMemDC.CreateCompatibleDC(NULL);
MaskMemDC.SelectObject(&m_MaskBitmap);
ImageMemDC.CreateCompatibleDC(NULL);
ImageMemDC.SelectObject(&m_ImageBitmap);
// get current dimensions of bitmap
CRect Maskrect(0, 0, m_MaskBitmapWidth, m_MaskBitmapHeight);
m_nDWidth = (int)(Maskrect.Width() * m_nZoomSize); // current dimens

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดก็ตาม ผู้ใช้ต้องรับผิดชอบต่อการใช้งานที่ไม่ถูกต้อง


```

    if(m_nStitch==1)
    {
// show solid stitch before cross stitch
        ::PaintDIB(pDC->m_hDC, &rcDest, pDoc->GetHDIB(),
            &rcDIB, pDoc->GetDocPalette(), SRCCOPY);
// show symbols stitch
        BeginWaitCursor();
        m_Bitmap.DeleteObject();
        m_Bitmap.LoadBitmap(IDB_RECT);
        m_Bitmap.GetObject(sizeof(BM), &BM);
        m_BitmapWidth = BM.bmWidth;
        m_BitmapHeight = BM.bmHeight;
        CDC tempMemDC;
        tempMemDC.CreateCompatibleDC(NULL);
        tempMemDC.SelectObject(&m_Bitmap);
        CRect temprect(0, 0, m_BitmapWidth, m_BitmapHeight);
        m_nDWidth = (int) (temprect.Width() * m_nZoomSize);
        m_nDHeight = (int) (temprect.Height() * m_nZoomSize);
// SetTextColor(pDC->m_hDC, RGB(0, 0, 0));
        COLORREF CheckColor, temp;

        TEXTMETRIC tm;
        CFont testFont;

        testFont.CreateFont(-4*(int)m_nZoomSize * 2, 0, 0, 0, 400, FALSE, FALSE,
            0, SYMBOL_CHARSET, OUT_DEFAULT_PRECIS,
            CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
            DEFAULT_PITCH | FF_ROMAN, "Wingdings");

        for(int y = 0; y < rcDest.bottom; y+=m_nDHeight)
            for(int x = 0; x < rcDest.right; x+=m_nDWidth)
            {
                CheckColor = (COLORREF) GetPixel(pDC->m_hDC, x+6, y+6);
                pDC->StretchBlt(x, y, m_nDWidth, m_nDHeight, &tempMemDC,
                    0, 0, m_BitmapWidth, m_BitmapHeight,
                    SRCCOPY);

                int i=0;
                BOOL found = FALSE;
                while((i<=pDoc->m_FlossArray.GetUpperBound()) && (!found))
                {
                    CFlossData* pFloss = (CFlossData*) pDoc->m_FlossArray.Get
At(i);
                    temp = (COLORREF) RGB(pFloss->m_lRed, pFloss->m_lGreen, p
Floss->m_lBlue);

                    int cred = GetRValue(CheckColor);
                    int cgreen = GetGValue(CheckColor);
                    int cblue = GetBValue(CheckColor);
                    int red = GetRValue(temp);
                    int green = GetGValue(temp);
                    int blue = GetBValue(temp);
                    if((cred > red-16) && (cred < red+16))
                        if((cgreen > green-16) && (cgreen < green+16))
                            if((cblue > blue-16) && (cblue < blue+16))
                                {
                                    CFont* pOldFont = (CFont*) pDC->SelectObject(
&testFont);
                                    pDC->GetTextMetrics(&tm);
                                    pDC->TextOut(x+3, y+3, i+33);
                                    pDC->SelectObject(pOldFont);
                                    found = TRUE;
                                }
                    i++;
                }
            }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    EndWaitCursor();
}
}
}

```

```

LRESULT CXstitchView::OnDoRealize(WPARAM wParam, LPARAM)
{

```

```

    ASSERT(wParam != NULL);
    CXstitchDoc* pDoc = GetDocument();
    if (pDoc->GetHDIB() == NULL)
        return 0L; // must be a new document

```

```

    CPalette* pPal = pDoc->GetDocPalette();
    if (pPal != NULL)
    {

```

```

        CMainFrame* pAppFrame = (CMainFrame*) AfxGetApp()->m_pMainWnd;
        ASSERT_KINDOF(CMainFrame, pAppFrame);

```

```

        CClientDC appDC(pAppFrame);

```

```

        // All views but one should be a background palette.

```

```

        // wParam contains a handle to the active view, so the SelectPalette

```

```

        // bForceBackground flag is FALSE only if wParam == m_hWnd (this view)

```

```

        CPalette* oldPalette = appDC.SelectPalette(pPal, ((HWND)wParam) != m_hWnd
);

```

```

    if (oldPalette != NULL)
    {

```

```

        UINT nColorsChanged = appDC.RealizePalette();

```

```

        if (nColorsChanged > 0)

```

```

            pDoc->UpdateAllViews(NULL);

```

```

            appDC.SelectPalette(oldPalette, TRUE);

```

```

        }

```

```

    else
    {

```

```

        TRACE0("\tSelectPalette failed in CDibView::OnPaletteChanged\n");

```

```

    }
}

```

```

return 0L;
}

```

```

void CXstitchView::OnInitialUpdate()
{

```

```

    CScrollView::OnInitialUpdate();

```

```

    // TODO: calculate the total size of this view

```

```

    ASSERT(GetDocument() != NULL);

```

```

    SetScrollSizes(MM_TEXT, GetDocument()->GetDocSize());
}

```

```

////////////////////////////////////
// CXstitchView printing

```

```

BOOL CXstitchView::OnPreparePrinting(CPrintInfo* pInfo)
{

```

```

    // default preparation

```

```

    return DoPreparePrinting(pInfo);
}

```

```

void CXstitchView::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{

```

```

    // TODO: add extra initialization before printing
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CXstitchView::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add cleanup after printing
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchView diagnostics

#ifdef _DEBUG
void CXstitchView::AssertValid() const
{
    CScrollView::AssertValid();
}

void CXstitchView::Dump(CDumpContext& dc) const
{
    CScrollView::Dump(dc);
}

CXstitchDoc* CXstitchView::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CXstitchDoc));
    return (CXstitchDoc*)m_pDocument;
}
#endif // !_DEBUG

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CXstitchView message handlers

void CXstitchView::OnStitchSolid()
{
    // TODO: Add your command handler code here
    m_nStitch = 2;
    OnCanvasGrid();
    NewClientRect();
}

void CXstitchView::OnStitchStitch()
{
    // TODO: Add your command handler code here
    m_nStitch = 0;
    OnCanvasGrid();
    NewClientRect();
}

void CXstitchView::OnStitchThin()
{
    // TODO: Add your command handler code here
    m_nStitch = 4;
    OnCanvasGrid();
    NewClientRect();
}

void CXstitchView::OnStitchSymbols()
{
    // TODO: Add your command handler code here
    m_nStitch = 1;
    OnCanvasGrid();
    NewClientRect();
}

void CXstitchView::OnUpdateStitchSolid(CCmdUI* pCmdUI)
{
    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    ไม่ว่าจะวิธีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
}

```

```

// TODO: Add your command update UI handler code here
pCmdUI->SetCheck(m_nStitch == 2);
}

void CXstitchView::OnUpdateStitchStitch(CCmdUI* pCmdUI)
{
// TODO: Add your command update UI handler code here
pCmdUI->SetCheck(m_nStitch == 0);
}

void CXstitchView::OnUpdateStitchThin(CCmdUI* pCmdUI)
{
// TODO: Add your command update UI handler code here
pCmdUI->SetCheck(m_nStitch == 4);
}

void CXstitchView::OnUpdateStitchSymbols(CCmdUI* pCmdUI)
{
// TODO: Add your command update UI handler code here
pCmdUI->SetCheck(m_nStitch == 1);
}

void CXstitchView::NewClientRect()
{
CRect ClientRect; // Get current dimensions of view window

GetClientRect(ClientRect);
InvalidateRect(ClientRect, TRUE); // Clear windows viewport
// and call OnDraw
}

void CXstitchView::OnViewSimple()
{
// TODO: Add your command handler code here
m_nZoomSize = 1.0; // set Zoom size to simple
GetDocument()->m_nPZoomSize = m_nZoomSize;
NewClientRect();
}

void CXstitchView::OnViewZoomIn()
{
// TODO: Add your command handler code here
m_nZoomSize+=1; // zoom in by multiple 2
if(m_nZoomSize < 1)
m_nZoomSize = 1;
GetDocument()->m_nPZoomSize = m_nZoomSize;
NewClientRect();
}

void CXstitchView::OnViewZoomOut()
{
// TODO: Add your command handler code here
m_nZoomSize-=1; // zoom out by divide 2
if(m_nZoomSize < 1)
m_nZoomSize = 0.5;
GetDocument()->m_nPZoomSize = m_nZoomSize;
NewClientRect();
}

void CXstitchView::OnCanvasAida()
{
// TODO: Add your command handler code here
m_nCanvas = 0;
m_Bitmap.DeleteObject(); //ตัดแปลงเนื้อหา // to return old resource
m_Bitmap.LoadBitmap(IDB_AIDAX); // load new resource
}

```

```

NewClientRect(); // update view
}

void CXstitchView::OnCanvasEverweave()
{
// TODO: Add your command handler code here
m_nCanvas = 3;
m_Bitmap.DeleteObject();
m_Bitmap.LoadBitmap(IDB_EVENX);
NewClientRect();
}

void CXstitchView::OnCanvasGrid()
{
// TODO: Add your command handler code here
m_nCanvas = 5;
m_Bitmap.DeleteObject();
m_Bitmap.LoadBitmap(IDB_GRIDX);
NewClientRect();
}

void CXstitchView::OnCanvasNone()
{
// TODO: Add your command handler code here
m_nCanvas = 6;
m_Bitmap.DeleteObject();
m_Bitmap.LoadBitmap(IDB_NONEX);
NewClientRect();
}

void CXstitchView::OnCanvasPlastic()
{
// TODO: Add your command handler code here
m_nCanvas = 2;
m_Bitmap.DeleteObject();
m_Bitmap.LoadBitmap(IDB_PLASX);
NewClientRect();
}

void CXstitchView::OnCanvasRug()
{
// TODO: Add your command handler code here
m_nCanvas = 4;
m_Bitmap.DeleteObject();
m_Bitmap.LoadBitmap(IDB_RUGX);
NewClientRect();
}

void CXstitchView::OnCanvasTapestry()
{
// TODO: Add your command handler code here
m_nCanvas = 1;
m_Bitmap.DeleteObject();
m_Bitmap.LoadBitmap(IDB_TAPX);
NewClientRect();
}

void CXstitchView::OnUpdateCanvasAida(CCmdUI* pCmdUI)
{
// TODO: Add your command update UI handler code here
pCmdUI->SetCheck(m_nCanvas == 0);
pCmdUI->Enable((m_nStitch == 0) || (m_nStitch == 4));
}

```

```

void CXstitchView::OnUpdateCanvasEverweave(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nCanvas == 3);
    pCmdUI->Enable((m_nStitch == 0) || (m_nStitch == 4));
}

void CXstitchView::OnUpdateCanvasGrid(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nCanvas == 5);
}

void CXstitchView::OnUpdateCanvasNone(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nCanvas == 6);
}

void CXstitchView::OnUpdateCanvasPlastic(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nCanvas == 2);
    pCmdUI->Enable((m_nStitch == 0) || (m_nStitch == 4));
}

void CXstitchView::OnUpdateCanvasRug(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nCanvas == 4);
    pCmdUI->Enable((m_nStitch == 0) || (m_nStitch == 4));
}

void CXstitchView::OnUpdateCanvasTapestry(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nCanvas == 1);
    pCmdUI->Enable((m_nStitch == 0) || (m_nStitch == 4));
}

void CXstitchView::OnActivateView(BOOL bActivate, CView* pActivateView, CView* pDeactivateView)
{
    // TODO: Add your specialized code here and/or call the base class
    CScrollView::OnActivateView(bActivate, pActivateView, pDeactivateView);

    if (bActivate)
    {
        ASSERT(pActivateView == this);
        OnDoRealize((WPARAM)m_hWnd, 0); // same as SendMessage(WM_DOREALIZE);
    }
}

int CXstitchView::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CScrollView::OnCreate(lpCreateStruct) == -1)
        return -1;

    // TODO: Add your specialized creation code here
    CClientDC dc(this);
    m_pDisplayMemDC->CreateCompatibleDC(&dc);
    CBitmap* pEmptyBitmap = new CBitmap;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เอกสารนี้จัดทำขึ้นโดยอัตโนมัติให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pEmptyBitmap->CreateCompatibleBitmap(&dc, 0, 0);
CBitmap* pOldBitmap = (CBitmap*) m_pDisplayMemDC->SelectObject(pEmptyBitmap);
m_hOldDisplayBitmap = (HBITMAP) pOldBitmap->GetSafeHandle();
return 0;
}

```

```

void CXstitchView::OnPrepareDC(CDC* pDC, CPrintInfo* pInfo)
{

```

```

// TODO: Add your specialized code here and/or call the base class
pDC->SetMapMode(MM_ANISOTROPIC);
pDC->SetWindowExt(1440, 1440);
pDC->SetViewportExt(pDC->GetDeviceCaps(LOGPIXELSX),
    -pDC->GetDeviceCaps(LOGPIXELSY));
CScrollView::OnPrepareDC(pDC, pInfo);
}

```

```

/*
void CXstitchView::ShowSymbol(CDC* pDC, int& nPosX, int& nPosY, int nPoints, char
symbol)
{

```

```

TEXTMETRIC tm;
CFont testFont;

testFont.CreateFont(-nPoints * 2, 0, 0, 0, 400, FALSE, FALSE,
    0, ANSI_CHARSET, OUT_DEFAULT_PRECIS,
    CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
    DEFAULT_PITCH | FF_SWISS, "Wingdings");
CFont* pOldFont = (CFont*) pDC->SelectObject(&testFont);
pDC->GetTextMetrics(&tm);
pDC->TextOut(nPosX+3, nPosY+3, symbol);
pDC->SelectObject(pOldFont);
}
*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// SymbolView.h : header file
//
```

```
////////////////////////////////////
// CSymbolView view
```

```
class CSymbolView : public CScrollView
{
private:
    int m_nStitch;          // variable for check stitch
    void NewClientRect();  // Clear ViewPort

protected:
    CSymbolView();        // protected constructor used by dynamic creation
    DECLARE_DYNCREATE(CSymbolView)
```

```
// Attributes
public:
    CXstitchDoc* GetDocument();
```

```
// Operations
public:
```

```
// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CSymbolView)
public:
    virtual void OnPrepareDC(CDC* pDC, CPrintInfo* pInfo = NULL);
protected:
    virtual void OnInitialUpdate();    // first time after construct
    virtual void OnDraw(CDC* pDC);
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
//}}AFX_VIRTUAL
```

```
// Implementation
protected:
    virtual ~CSymbolView();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
```

```
// Generated message map functions
//{{AFX_MSG(CSymbolView)
afx_msg void OnStitchSolid();
afx_msg void OnStitchStitch();
afx_msg void OnStitchSymbols();
afx_msg void OnUpdateStitchSolid(CCmdUI* pCmdUI);
afx_msg void OnUpdateStitchStitch(CCmdUI* pCmdUI);
afx_msg void OnUpdateStitchSymbols(CCmdUI* pCmdUI);
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};
```

```
#ifndef _DEBUG // debug version in xstitchView.cpp
inline CXstitchDoc* CSymbolView::GetDocument()
{ return (CXstitchDoc*)m_pDocument; }
#endif
```

```
//////////////////////////////////// คำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
// SymbolView.cpp : implementation file
//
```

```
#include "stdafx.h"
#include "xstitch.h"
#include "FlossData.h"
#include "xstitchDoc.h"
#include "SymbolView.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CSymbolView
```

```
IMPLEMENT_DYNCREATE(CSymbolView, CScrollView)
```

```
CSymbolView::CSymbolView()
{
    m_nStitch = 2;
}
```

```
CSymbolView::~CSymbolView()
{
}
```

```
BEGIN_MESSAGE_MAP(CSymbolView, CScrollView)
    //{{AFX_MSG_MAP(CSymbolView)
    ON_COMMAND(ID_STITCH_SOLID, OnStitchSolid)
    ON_COMMAND(ID_STITCH_STITCH, OnStitchStitch)
    ON_COMMAND(ID_STITCH_SYMBOLS, OnStitchSymbols)
    ON_UPDATE_COMMAND_UI(ID_STITCH_SOLID, OnUpdateStitchSolid)
    ON_UPDATE_COMMAND_UI(ID_STITCH_STITCH, OnUpdateStitchStitch)
    ON_UPDATE_COMMAND_UI(ID_STITCH_SYMBOLS, OnUpdateStitchSymbols)
    //}}AFX_MSG_MAP
    // Standard printing commands
    ON_COMMAND(ID_FILE_PRINT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, CScrollView::OnFilePrintPreview)
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CSymbolView drawing
```

```
void CSymbolView::OnInitialUpdate()
{
    CScrollView::OnInitialUpdate();
    CSize sizeTotal;

    sizeTotal.cx = 200;
    sizeTotal.cy = GetDocument()->GetArraySize()*20;
    if(sizeTotal.cy < 0)
        sizeTotal.cy = 200;

    CSize pageSize = CSize(sizeTotal.cx / 2, sizeTotal.cy / 2);
    CSize lineSize = CSize(sizeTotal.cx / 100, sizeTotal.cy / 100);
```

```
// TODO: calculate the total size of this view
SetScrollSizes(MM_LOENGLISH, sizeTotal, pageSize, lineSize);
}
```

```

void CSymbolView::OnDraw(CDC* pDC)
{
    CXstitchDoc* pDoc = GetDocument();

    // TODO: add draw code here
/*
    pDC->TextOut(1, -10, "Symbol");
    pDC->TextOut(30, -10, "Number");
    pDC->TextOut(80, -10, "Color");
*/

    CFlossData* pFlossData;
    COLORREF CheckColor;
    long red, green, blue;
    int nCount = pDoc->m_FlossArray.GetUpperBound();
    int nPos = -20;
    CFont textFont, symbolFont;

    symbolFont.CreateFont(-11, 0, 0, 0, 400, FALSE, FALSE,
        0, SYMBOL_CHARSET, OUT_DEFAULT_PRECIS,
        CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
        DEFAULT_PITCH | FF_SWISS, "WingDings");
    textFont.CreateFont(-12, 0, 0, 0, 400, FALSE, FALSE,
        0, ANSI_CHARSET, OUT_DEFAULT_PRECIS,
        CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
        DEFAULT_PITCH | FF_ROMAN, NULL);

    for(int i=0; i<=nCount; i++)
    {
        pFlossData = (CFlossData*) pDoc->m_FlossArray.GetAt(i);
        red = pFlossData->m_lRed;
        green = pFlossData->m_lGreen;
        blue = pFlossData->m_lBlue;

        CBrush Brush, *PtrOldBrush;
        Brush.CreateSolidBrush(RGB(red,green,blue));
        PtrOldBrush = pDC->SelectObject(&Brush);
        pDC->Rectangle(6, nPos, 20, nPos-14);
        pDC->SelectObject(PtrOldBrush);

        if(m_nStitch==1)
        {
            CheckColor = GetPixel(pDC->m_hDC, 10, (i*nPos)-6);

            CBrush Brush, *PtrOldBrush;
            Brush.CreateSolidBrush(RGB(255, 255 ,255));
            PtrOldBrush = pDC->SelectObject(&Brush);
            pDC->Rectangle(5, nPos+2, 24, nPos-16);
            pDC->SelectObject(PtrOldBrush);

            CFont* pOldFont = (CFont*) pDC->SelectObject(&symbolFont);
            TEXTMETRIC tm;
            pDC->GetTextMetrics(&tm);
            pDC->TextOut(7, nPos, 33+i);
            pDC->SelectObject(pOldFont);
        }

        pDC->SelectObject(&textFont);
        TEXTMETRIC tm;
        pDC->GetTextMetrics(&tm);
        pDC->TextOut(30, nPos, pFlossData->m_number);
        pDC->TextOut(65, nPos, pFlossData->m_name);
        nPos -= 20;//tm.tmHeight + tm.tmExternalLeading;
    }
}

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

////////////////////////////////////
// CSymbolView diagnostics

#ifdef _DEBUG
void CSymbolView::AssertValid() const
{
    CScrollView::AssertValid();
}

void CSymbolView::Dump(CDumpContext& dc) const
{
    CScrollView::Dump(dc);
}

CXstitchDoc* CSymbolView::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CXstitchDoc)));
    return (CXstitchDoc*)m_pDocument;
}
#endif // _DEBUG

////////////////////////////////////
// CSymbolView message handlers

void CSymbolView::OnStitchSolid()
{
    // TODO: Add your command handler code here
    m_nStitch = 2;
    NewClientRect();
}

void CSymbolView::OnStitchStitch()
{
    // TODO: Add your command handler code here
    m_nStitch = 0;
    NewClientRect();
}

void CSymbolView::OnStitchSymbols()
{
    // TODO: Add your command handler code here
    m_nStitch = 1;
    NewClientRect();
}

void CSymbolView::OnUpdateStitchSolid(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nStitch == 2);
}

void CSymbolView::OnUpdateStitchStitch(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nStitch == 0);
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void CSymbolView::OnUpdateStitchSymbols(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->SetCheck(m_nStitch == 1);
}

void CSymbolView::NewClientRect()
{
    CRect ClientRect;    // Get current dimensions of view window

    GetClientRect(ClientRect);
    InvalidateRect(ClientRect, TRUE);    // Clear windows viewport
                                         // and call OnDraw
}

BOOL CSymbolView::OnPreparePrinting(CPrintInfo* pInfo)
{
    // TODO: Add your specialized code here and/or call the base class
    return DoPreparePrinting(pInfo);
// return CScrollView::OnPreparePrinting(pInfo);
}

void CSymbolView::OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo)
{
    // TODO: Add your specialized code here and/or call the base class
    CScrollView::OnBeginPrinting(pDC, pInfo);
}

void CSymbolView::OnEndPrinting(CDC* pDC, CPrintInfo* pInfo)
{
    // TODO: Add your specialized code here and/or call the base class
    CScrollView::OnEndPrinting(pDC, pInfo);
}

void CSymbolView::OnPrepareDC(CDC* pDC, CPrintInfo* pInfo)
{
    // TODO: Add your specialized code here and/or call the base class
    CScrollView::OnPrepareDC(pDC, pInfo);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// FlossTabDlg.h : header file
//

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CFlossTabDlg dialog

class CFlossTabDlg : public CDialog
{
// Construction
public:
    CFlossTabDlg(CWnd* pParent = NULL);    // standard constructor

// Dialog Data
   //{{AFX_DATA(CFlossTabDlg)
    enum { IDD = IDD_FLOSS_TABLE };
        // NOTE: the ClassWizard will add data members here
   //}}AFX_DATA

// Overrides
    // ClassWizard generated virtual function overrides
   //{{AFX_VIRTUAL(CFlossTabDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
   //}}AFX_VIRTUAL

// Implementation
protected:
    afx_msg BOOL OnQueryNewPalette();
    afx_msg void OnPaletteChanged(CWnd* pFocusWnd);
    CPalette* SetPalette(CPalette* pPalette);
    CPalette* m_pPalette;
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);

    // Generated message map functions
   //{{AFX_MSG(CFlossTabDlg)
    afx_msg void OnColor();
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// FlossTabDlg.cpp : implementation file
//
```

```
#include "stdafx.h"
#include "xstitch.h"
#include "FlossTabDlg.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CFlossTabDlg dialog
```

```
CFlossTabDlg::CFlossTabDlg(CWnd* pParent /*=NULL*/)
: CDialog(CFlossTabDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CFlossTabDlg)
    // NOTE: the ClassWizard will add member initialization here
    //}}AFX_DATA_INIT
}
```

```
void CFlossTabDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CFlossTabDlg)
    // NOTE: the ClassWizard will add DDX and DDV calls here
    //}}AFX_DATA_MAP
}
```

```
BEGIN_MESSAGE_MAP(CFlossTabDlg, CDialog)
    ON_WM_QUERYNEWPALETTE()
    ON_WM_PALETTECHANGED()
    ON_WM_CREATE()
    //{{AFX_MSG_MAP(CFlossTabDlg)
    ON_BN_CLICKED(IDC_COLOR, OnColor)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CFlossTabDlg message handlers
```

```
void CFlossTabDlg::OnColor()
{
    // TODO: Add your control notification handler code here
    CColorDialog ColorDialog;

    if(ColorDialog.DoModal() == IDOK)
        return;
}
```

```
BOOL CFlossTabDlg::OnQueryNewPalette()
{
```

```
    // CG: This function was added by the Palette Support component
```

```
    if (m_pPalette == NULL)
```

เอกสารนี้เป็นเอกสารส่วนบุคคลไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// BLOCK
```

```

{
    CClientDC dc(this);
    CPalette* pOldPalette = dc.SelectPalette(m_pPalette,
        GetCurrentMessage()->message == WM_PALETTECHANGED);
    UINT nChanged = dc.RealizePalette();
    dc.SelectPalette(pOldPalette, TRUE);

    if (nChanged == 0)
        return FALSE;
}

Invalidate();

return TRUE;
}

void CFlossTabDlg::OnPaletteChanged(CWnd* pFocusWnd)
{
    // CG: This function was added by the Palette Support component

    if (pFocusWnd == this || IsChild(pFocusWnd))
        return;

    OnQueryNewPalette();
}

CPalette* CFlossTabDlg::SetPalette(CPalette* pPalette)
{
    // CG: This function was added by the Palette Support component

    // Call this function when the palette changes. It will
    // realize the palette in the foreground to cause the screen
    // to repaint correctly. All calls to CDC::SelectPalette in
    // painting code should select palettes in the background.

    CPalette* pOldPalette = m_pPalette;
    m_pPalette = pPalette;
    OnQueryNewPalette();
    return pOldPalette;
}

int CFlossTabDlg::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    // CG: This function was added by the Palette Support component

    if (CDialog::OnCreate(lpCreateStruct) == -1)
        return -1;

    m_pPalette = NULL;
    return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ImportDlg.h : header file
//

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// ImportDlg dialog

class ImportDlg : public CDialog
{
// Construction
public:
    ImportDlg(CWnd* pParent = NULL);    // standard constructor

// Dialog Data
    //{AFX_DATA(ImportDlg)
    enum { IDD = IDD_IMPORT_DESIGN };
    long      m_DesignScale;
    long      m_DesignPalette;
    CString  m_DesignFloss;
    CString  m_dx;
    CString  m_dy;
    CString  m_sx;
    CString  m_sy;
    //}}AFX_DATA

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(ImportDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:

    // Generated message map functions
    //{AFX_MSG(ImportDlg)
    virtual void OnOK();
    virtual void OnCancel();
    afx_msg void OnChangeDesignPalette();
    afx_msg void OnChangeDesignScale();
    virtual BOOL OnInitDialog();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ImportDlg.cpp : implementation file
//
```

```
#include "stdafx.h"
#include "xstitch.h"
#include "ImportDlg.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// ImportDlg dialog
```

```
ImportDlg::ImportDlg(CWnd* pParent /*=NULL*/)
: CDialog(ImportDlg::IDD, pParent)
```

```
{
    //{{AFX_DATA_INIT(ImportDlg)
    m_DesignScale = 0;
    m_DesignPalette = 0;
    m_DesignFloss = _T("");
    m_dx = _T("");
    m_dy = _T("");
    m_sx = _T("");
    m_sy = _T("");
    //}}AFX_DATA_INIT
}
```

```
void ImportDlg::DoDataExchange(CDataExchange* pDX)
```

```
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(ImportDlg)
    DDX_Text(pDX, IDC_DESIGN_SCALE, m_DesignScale);
    DDX_Text(pDX, IDC_DESIGN_PALETTE, m_DesignPalette);
    DDX_CBString(pDX, IDC_DESIGN_FLOSS, m_DesignFloss);
    DDX_Text(pDX, IDC_DX, m_dx);
    DDX_Text(pDX, IDC_DY, m_dy);
    DDX_Text(pDX, IDC_SX, m_sx);
    DDX_Text(pDX, IDC_SY, m_sy);
    //}}AFX_DATA_MAP
}
```

```
BEGIN_MESSAGE_MAP(ImportDlg, CDialog)
```

```
    //{{AFX_MSG_MAP(ImportDlg)
    ON_EN_CHANGE(IDC_DESIGN_PALETTE, OnChangeDesignPalette)
    ON_EN_CHANGE(IDC_DESIGN_SCALE, OnChangeDesignScale)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// ImportDlg message handlers
```

```
void ImportDlg::OnOK()
```

```
{
    // TODO: Add extra validation here
```

```
    CDialog::OnOK();
    // ไม่ควรกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
}
```

```
void ImportDlg::OnCancel()
{
    // TODO: Add extra cleanup here

    CDialog::OnCancel();
}

void ImportDlg::OnChangeDesignPalette()
{
    // TODO: Add your control notification handler code here
}

void ImportDlg::OnChangeDesignScale()
{
    // TODO: Add your control notification handler code here
}

BOOL ImportDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // TODO: Add extra initialization here
    SetDlgItemText(IDC_SX, m_dx);
    SetDlgItemText(IDC_SY, m_dy);

    return TRUE; // return TRUE unless you set the focus to a control
                // EXCEPTION: OCX Property Pages should return FALSE
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Source Code

โปรแกรม MakeFloss

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
=====
MICROSOFT FOUNDATION CLASS LIBRARY : MakeFloss
=====
```

AppWizard has created this MakeFloss application for you. This application not only demonstrates the basics of using the Microsoft Foundation classes but is also a starting point for writing your application.

This file contains a summary of what you will find in each of the files that make up your MakeFloss application.

MakeFloss.h

This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CMakeFlossApp application class.

MakeFloss.cpp

This is the main application source file that contains the application class CMakeFlossApp.

MakeFloss.rc

This is a listing of all of the Microsoft Windows resources that the program uses. It includes the icons, bitmaps, and cursors that are stored in the RES subdirectory. This file can be directly edited in Microsoft Developer Studio.

res\MakeFloss.ico

This is an icon file, which is used as the application's icon. This icon is included by the main resource file MakeFloss.rc.

res\MakeFloss.rc2

This file contains resources that are not edited by Microsoft Developer Studio. You should place all resources not editable by the resource editor in this file.

MakeFloss.clw

This file contains information used by ClassWizard to edit existing classes or add new classes. ClassWizard also uses this file to store information needed to create and edit message maps and dialog data maps and to create prototype member functions.

```
////////////////////////////////////
```

For the main frame window:

MainFrm.h, MainFrm.cpp

These files contain the frame class CMainFrame, which is derived from CFrameWnd and controls all SDI frame features.

res\Toolbar.bmp

This bitmap file is used to create tiled images for the toolbar. The initial toolbar and status bar are constructed in the CMainFrame class. Edit this toolbar bitmap along with the array in MainFrm.cpp to add more toolbar buttons.

```
////////////////////////////////////
```

AppWizard creates one document type and one view:

MakeFlossDoc.h, MakeFlossDoc.cpp - the document

These files contain your CMakeFlossDoc class. Edit these files to add your special document data and to implement file saving and loading (via CMakeFlossDoc::Serialize).
 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัท ไมโครซอฟท์ จำกัด ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่สามารถนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากไมโครซอฟท์ กรุณาแจ้งไปยังไมโครซอฟท์หากพบข้อผิดพลาดใดๆ

MakeFlossView.h, MakeFlossView.cpp - the view of the document
 These files contain your CMakeFlossView class.
 CMakeFlossView objects are used to view CMakeFlossDoc objects.

////////////////////////////////////
 Other standard files:

StdAfx.h, StdAfx.cpp

These files are used to build a precompiled header (PCH) file
 named MakeFloss.pch and a precompiled types file named StdAfx.obj.

Resource.h

This is the standard header file, which defines new resource IDs.
 Microsoft Developer Studio reads and updates this file.

////////////////////////////////////
 Other notes:

AppWizard uses "TODO:" to indicate parts of the source code you
 should add to or customize.

If your application uses MFC in a shared DLL, and your application is
 in a language other than the operating system's current language, you
 will need to copy the corresponding localized resources MFC40XXX.DLL
 from the Microsoft Visual C++ CD-ROM onto the system or system32 directory,
 and rename it to be MFCLOC.DLL. ("XXX" stands for the language abbreviation.
 For example, MFC40DEU.DLL contains resources translated to German.) If you
 don't do this, some of the UI elements of your application will remain in the
 language of the operating system.

////////////////////////////////////

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// stdafx.h : include file for standard system include files,  
// or project specific include files that are used frequently, but  
// are changed infrequently  
//  
  
#define VC_EXTRALEAN // Exclude rarely-used stuff from Windows headers  
  
#include <afxwin.h> // MFC core and standard components  
#include <afxext.h> // MFC extensions  
#ifndef _AFX_NO_AFXCMN_SUPPORT  
#include <afxcmn.h> // MFC support for Windows 95 Common Controls  
#endif // _AFX_NO_AFXCMN_SUPPORT
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// stdafx.cpp : source file that includes just the standard includes  
// MakeFloss.pch will be the pre-compiled header  
// stdafx.obj will contain the pre-compiled type information
```

```
#include "stdafx.h"
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// FlossData.h
```

```
class CFlossData : public CObject
```

```
{
```

```
    DECLARE_SERIAL(CFlossData)
```

```
public:
```

```
    LONG    m_lBlue;
```

```
    LONG    m_lGreen;
```

```
    LONG    m_lRed;
```

```
    CString m_number;
```

```
    CString m_name;
```

```
CFlossData()
```

```
{
```

```
    m_lBlue = m_lGreen = m_lRed = 0;
```

```
}
```

```
CFlossData(const char* szNumber, const char* szName, int lBlue,
            int lGreen, int lRed) : m_number(szNumber)
```

```
{
```

```
    m_name = szName;
```

```
    m_lBlue = lBlue;
```

```
    m_lGreen = lGreen;
```

```
    m_lRed = lRed;
```

```
}
```

```
CFlossData(const CFlossData& s) : m_number(s.m_number)
```

```
{
```

```
    m_name = s.m_name;
```

```
    m_lBlue = s.m_lBlue;
```

```
    m_lGreen = s.m_lGreen;
```

```
    m_lRed = s.m_lRed;
```

```
}
```

```
const CFlossData& operator = (const CFlossData& s)
```

```
{
```

```
    m_number = s.m_number;
```

```
    m_name = s.m_name;
```

```
    m_lBlue = s.m_lBlue;
```

```
    m_lGreen = s.m_lGreen;
```

```
    m_lRed = s.m_lRed;
```

```
    return *this;
```

```
}
```

```
BOOL operator == (const CFlossData& s) const
```

```
{
```

```
    if((m_number == s.m_number) && (m_name == s.m_name) &&
        (m_lBlue == s.m_lBlue) && (m_lGreen == s.m_lGreen) &&
        (m_lRed == s.m_lRed))
```

```
    {
```

```
        return TRUE;
```

```
    }
```

```
    else
```

```
    {
```

```
        return FALSE;
```

```
    }
```

```
}
```

```
BOOL operator != (const CFlossData& s) const
```

```
{
```

```
    // let's make use of the operator we just defined
```

```
    return !(*this == s);
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
virtual void Serialize(CArchive& ar);

```
#ifdef _DEBUG  
    void Dump(CDumpContext& dc) const;  
#endif  
};
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "stdafx.h"
#include "FlossData.h"

IMPLEMENT_SERIAL(CFlossData, CObject, 0)

void CFlossData::Serialize(CArchive& ar)
{
    TRACE("Entering CStudent::Serialize\n");
    if(ar.IsStoring())
    {
        ar << m_lBlue << m_lGreen << m_lRed << m_number << m_name;
    }
    else
    {
        ar >> m_lBlue >> m_lGreen >> m_lRed >> m_number >> m_name;
    }
}

#ifdef _DEBUG
void CFlossData::Dump(CDumpContext& dc) const
{
    CObject::Dump(dc);
    dc << "\nm_number = " << m_number << "\nm_name = " << m_name
        << "\nm_lBlue = " << m_lBlue << "\nm_lGreen = " << m_lGreen
        << "\nm_lRed = " << m_lRed;
}
#endif

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// MainFrm.h : interface of the CMainFrame class
//
/////////////////////////////////////////////////////////////////

class CMainFrame : public CFrameWnd
{
protected: // create from serialization only
    CMainFrame();
    DECLARE_DYNCREATE(CMainFrame)

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CMainFrame)
public:
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    virtual void ActivateFrame(int nCmdShow = -1);
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected: // control bar embedded members
    CStatusBar m_wndStatusBar;
    CToolBar m_wndToolBar;

// Generated message map functions
protected:
    //{{AFX_MSG(CMainFrame)
afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    // NOTE - the ClassWizard will add and remove member functions here.
    // DO NOT EDIT what you see in these blocks of generated code!
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// MainFrm.cpp : implementation of the CMainFrame class
//
```

```
#include "stdafx.h"
#include "MakeFloss.h"
```

```
#include "MainFrm.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CMainFrame
```

```
IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)
```

```
BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
    //{{AFX_MSG_MAP(CMainFrame)
    // NOTE - the ClassWizard will add and remove mapping macros here.
    // DO NOT EDIT what you see in these blocks of generated code !
    ON_WM_CREATE()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
static UINT indicators[] =
{
    ID_SEPARATOR,           // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};
```

```
////////////////////////////////////
// CMainFrame construction/destruction
```

```
CMainFrame::CMainFrame()
{
    // TODO: add member initialization code here
}
```

```
CMainFrame::~CMainFrame()
{
}
```

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndToolBar.Create(this) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;          // fail to create
    }
}
```

```
if (!m_wndStatusBar.Create(this) ||
    !m_wndStatusBar.SetIndicators(indicators,
        sizeof(indicators)/sizeof(UINT)))
    TRACE0("Failed to create status bar\n");
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return -1;        // fail to create
}

// TODO: Remove this if you don't want tool tips or a resizable toolbar
m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle() |
    CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC);

// TODO: Delete these three lines if you don't want the toolbar to
// be dockable
m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
EnableDocking(CBRS_ALIGN_ANY);
DockControlBar(&m_wndToolBar);

return 0;
}

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CFrameWnd::PreCreateWindow(cs);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CMainFrame diagnostics

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
    CFrameWnd::Dump(dc);
}

#endif // _DEBUG

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CMainFrame message handlers

void CMainFrame::ActivateFrame(int nCmdShow)
{
    // TODO: Add your specialized code here and/or call the base class
    TRACE("Entering CMainFrame::ActivateFrame\n");
    CFrameWnd::ActivateFrame(nCmdShow);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// MakeFloss.h : main header file for the MAKEFLOSS application
//

#ifdef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h"          // main symbols

////////////////////////////////////

// CMakeFlossApp:
// See MakeFloss.cpp for the implementation of this class
//

class CMakeFlossApp : public CWinApp
{
public:
    CMakeFlossApp();

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CMakeFlossApp)
public:
    virtual BOOL InitInstance();
//}}AFX_VIRTUAL

// Implementation

//{{AFX_MSG(CMakeFlossApp)
afx_msg void OnAppAbout();
// NOTE - the ClassWizard will add and remove member functions here.
//      DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// MakeFloss.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "MakeFloss.h"

#include "MainFrm.h"
#include "MakeFlossDoc.h"
#include "MakeFlossView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CMakeFlossApp

BEGIN_MESSAGE_MAP(CMakeFlossApp, CWinApp)
//{{AFX_MSG_MAP(CMakeFlossApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
    // NOTE - the ClassWizard will add and remove mapping macros here.
    //      DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
// Standard file based document commands
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
// Standard print setup command
ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

////////////////////////////////////
// CMakeFlossApp construction

CMakeFlossApp::CMakeFlossApp()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

////////////////////////////////////
// The one and only CMakeFlossApp object

CMakeFlossApp theApp;

////////////////////////////////////
// CMakeFlossApp initialization

BOOL CMakeFlossApp::InitInstance()
{
    // Standard initialization
    // If you are not using these features and wish to reduce the size
    // of your final executable, you should remove from the following
    // the specific initialization routines you do not need.

#ifdef _AFXDLL
    Enable3dControls();          // Call this when using MFC in a shared DLL
#else
    Enable3dControlsStatic();    // Call this when linking to MFC statically
#endif

    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    LoadStdProfileSettings();    // Load standard INI file options (including MRU)
    ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    // Register the application's document templates. Document templates

```

```

// serve as the connection between documents, frame windows and views.

CSingleDocTemplate* pDocTemplate;
pDocTemplate = new CSingleDocTemplate(
    IDR_MAINFRAME,
    RUNTIME_CLASS(CMakeFlossDoc),
    RUNTIME_CLASS(CMainFrame),          // main SDI frame window
    RUNTIME_CLASS(CMakeFlossView));
AddDocTemplate(pDocTemplate);

// Parse command line for standard shell commands, DDE, file open
CCommandLineInfo cmdInfo;
ParseCommandLine(cmdInfo);

// Dispatch commands specified on the command line
if (!ProcessShellCommand(cmdInfo))
    return FALSE;

return TRUE;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
    // No message handlers
//}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    //}}AFX_MSG_MAP

```

```
END_MESSAGE_MAP()
```

```
// App command to run the dialog
```

```
void CMakeFlossApp::OnAppAbout()
```

```
{
```

```
    CAboutDlg aboutDlg;
```

```
    aboutDlg.DoModal();
```

```
}
```

```
////////////////////////////////////
```

```
// CMakeFlossApp commands
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// MakeFlossDoc.h : interface of the CMakeFlossDoc class
//
/////////////////////////////////////////////////////////////////

class CMakeFlossDoc : public CDocument
{
protected: // create from serialization only
    CMakeFlossDoc();
    DECLARE_DYNCREATE(CMakeFlossDoc)

private:
    COBList m_flossList;

// Attributes
public:
    COBList* GetList()
    {
        return &m_flossList;
    }

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CMakeFlossDoc)
public:
    virtual BOOL OnNewDocument();
    virtual void Serialize(CArchive& ar);
    virtual void DeleteContents();
    virtual BOOL OnOpenDocument(LPCTSTR lpszPathName);
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CMakFlossDoc();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
protected:

// Generated message map functions
protected:
    //{{AFX_MSG(CMakeFlossDoc)
afx_msg void OnUpdateFileSave(CCmdUI* pCmdUI);
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// MakeFlossDoc.cpp : implementation of the CMakeFlossDoc class
//
```

```
#include "stdafx.h"
#include "MakeFloss.h"
```

```
#include "MakeFlossDoc.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CMakeFlossDoc
```

```
IMPLEMENT_DYNCREATE(CMakeFlossDoc, CDocument)
```

```
BEGIN_MESSAGE_MAP(CMakeFlossDoc, CDocument)
//{{AFX_MSG_MAP(CMakeFlossDoc)
ON_UPDATE_COMMAND_UI(ID_FILE_SAVE, OnUpdateFileSave)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CMakeFlossDoc construction/destruction
```

```
CMakeFlossDoc::CMakeFlossDoc()
{
    // TODO: add one-time construction code here
#ifdef _DEBUG
    afxDump.SetDepth(1);
#endif
}
```

```
CMakeFlossDoc::~CMakeFlossDoc()
{
}
```

```
BOOL CMakeFlossDoc::OnNewDocument()
{
    TRACE("Entering CMakeFlossDoc::OnNewDocument\n");
    if (!CDocument::OnNewDocument())
        return FALSE;

    // TODO: add reinitialization code here
    // (SDI documents will reuse this document)

    return TRUE;
}
```

```
BOOL CMakeFlossDoc::OnOpenDocument(LPCTSTR lpszPathName)
{
    TRACE("Entering CMakeFlossDoc::OnOpenDocument\n");
    if (!CDocument::OnOpenDocument(lpszPathName))
        return FALSE;

    // TODO: Add your specialized creation code here

    return TRUE;
}
```

} เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
////////////////////////////////////
// CMakeFlossDoc serialization
```

```

void CMakeFlossDoc::Serialize(CArchive& ar)
{
    if (ar.IsStoring())
    {
        // TODO: add storing code here
    }
    else
    {
        // TODO: add loading code here
    }
    m_flossList.Serialize(ar);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CMakeFlossDoc diagnostics

#ifdef _DEBUG
void CMakeFlossDoc::AssertValid() const
{
    CDocument::AssertValid();
}

void CMakeFlossDoc::Dump(CDumpContext& dc) const
{
    CDocument::Dump(dc);
}
#endif // _DEBUG

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CMakeFlossDoc commands

void CMakeFlossDoc::DeleteContents()
{
    // TODO: Add your specialized code here and/or call the base class
#ifdef _DEBUG
    Dump(afxDump);
#endif

    while(m_flossList.GetHeadPosition())
    {
        delete m_flossList.RemoveHead();
    }
    CDocument::DeleteContents();
}

void CMakeFlossDoc::OnUpdateFileSave(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->Enable(IsModified());
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// MakeFlossView.h : interface of the CMakeFlossView class
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

class CMakeFlossView : public CFormView
{
protected: // create from serialization only
    CMakeFlossView();
    DECLARE_DYNCREATE(CMakeFlossView)

public:
   //{{AFX_DATA(CMakeFlossView)
    enum{ IDD = IDD_MAKEFLOSS_FORM };
        // NOTE: the ClassWizard will add data members here
    CString m_number;
    CString m_name;
    long     m_lRed;
    long     m_lGreen;
    long     m_lBlue;
    CString m_ListFloss;
    //}}AFX_DATA
    CBrush  m_DialogBrush;
    CRect    m_RectColor;
    enum { nMin = 0 };
    enum { nMax = 255 };

// Attributes
public:
    CMakeFlossDoc* GetDocument();

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CMakeFlossView)
    public:
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    virtual void OnInitialUpdate();
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnPrint(CDC* pDC, CPrintInfo*);
    virtual void OnUpdate(CView* pSender, LPARAM lHint, CObject* pHint);
    virtual void OnDraw(CDC* pDC);
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CMakFlossView();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:
    POSITION m_position;
    COBList* m_pList;

// Generated message map functions
protected:
    //{{AFX_MSG(CMakeFlossView)
    afx_msg void OnDefineColor();

```

```

afx_msg void OnAdd();
afx_msg void OnRemove();
afx_msg void OnPrev();
afx_msg void OnNext();
afx_msg void OnClear();
afx_msg void OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar);
afx_msg void OnChangeRed();
afx_msg void OnChangeGreen();
afx_msg void OnChangeBlue();
afx_msg void OnDblclkListFloss();
//}}AFX_MSG

protected:
    virtual void GetEntry(POSITION position);
    virtual void InsertEntry(POSITION position);
    virtual void ClearEntry();

    virtual void UpdateScrollBar();

    DECLARE_MESSAGE_MAP()
};

#ifdef _DEBUG // debug version in MakeFlossView.cpp
inline CMakeFlossDoc* CMakeFlossView::GetDocument()
    { return (CMakeFlossDoc*)m_pDocument; }
#endif

```

////////////////////////////////////



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// MakeFlossView.cpp : implementation of the CMakeFlossView class
```

```
//
```

```
#include "stdafx.h"
#include "MakeFloss.h"
```

```
#include "FlossData.h"
#include "MakeFlossDoc.h"
#include "MakeFlossView.h"
```

```
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```
////////////////////////////////////
// CMakeFlossView
```

```
IMPLEMENT_DYNCREATE(CMakeFlossView, CFormView)
```

```
BEGIN_MESSAGE_MAP(CMakeFlossView, CFormView)
    //{{AFX_MSG_MAP(CMakeFlossView)
    ON_BN_CLICKED(IDC_DEFINE_COLOR, OnDefineColor)
    ON_BN_CLICKED(IDC_ADD, OnAdd)
    ON_BN_CLICKED(IDC_REMOVE, OnRemove)
    ON_BN_CLICKED(IDC_PREV, OnPrev)
    ON_BN_CLICKED(IDC_NEXT, OnNext)
    ON_BN_CLICKED(IDC_CLEAR, OnClear)
    ON_WM_HSCROLL()
    ON_EN_CHANGE(IDC_RED, OnChangeRed)
    ON_EN_CHANGE(IDC_GREEN, OnChangeGreen)
    ON_EN_CHANGE(IDC_BLUE, OnChangeBlue)
    ON_LBN_DBLCLK(IDC_LIST_FLOSS, OnDbldclkListFloss)
    //}}AFX_MSG_MAP
    // Standard printing commands
    ON_COMMAND(ID_FILE_PRINT, CFormView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, CFormView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, CFormView::OnFilePrintPreview)
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CMakeFlossView construction/destruction
```

```
CMakeFlossView::CMakeFlossView()
    : CFormView(CMakeFlossView::IDD)
{
    TRACE("Entering CMakeFlossView constructor\n");
    //{{AFX_DATA_INIT(CMakeFlossView)
    // NOTE: the ClassWizard will add member initialization here
    m_number = "";
    m_name = "";
    m_lRed = 0;
    m_lGreen = 0;
    m_lBlue = 0;
    m_ListFloss = "";
    //}}AFX_DATA_INIT
    // TODO: add construction code here
    m_position = NULL;
}
```

```
CMakeFlossView::~CMakeFlossView()
{
    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
}
```

```

void CMakeFlossView::DoDataExchange(CDataExchange* pDX)
{
    CFormView::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CMakeFlossView)
        // NOTE: the ClassWizard will add DDX and DDV calls here
    DDX_Text(pDX, IDC_BLUE, m_lBlue);
    DDV_MinMaxLong(pDX, m_lBlue, 0, 255);
    DDX_Text(pDX, IDC_GREEN, m_lGreen);
    DDV_MinMaxLong(pDX, m_lGreen, 0, 255);
    DDX_Text(pDX, IDC_RED, m_lRed);
    DDV_MinMaxLong(pDX, m_lRed, 0, 255);
    DDX_Text(pDX, IDC_NAME, m_name);
    DDX_Text(pDX, IDC_NUMBER, m_number);
    DDX_Text(pDX, IDC_LIST_FLOSS, m_ListFloss);
    //}}AFX_DATA_MAP
}

BOOL CMakeFlossView::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CFormView::PreCreateWindow(cs);
}

//////////////////////////////////////
// CMakeFlossView printing
//////////////////////////////////////

BOOL CMakeFlossView::OnPreparePrinting(CPrintInfo* pInfo)
{
    // default preparation
    return DoPreparePrinting(pInfo);
}

void CMakeFlossView::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add extra initialization before printing
}

void CMakeFlossView::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
    // TODO: add cleanup after printing
}

void CMakeFlossView::OnPrint(CDC* pDC, CPrintInfo*)
{
    // TODO: add code to print the controls
}

//////////////////////////////////////
// CMakeFlossView diagnostics
//////////////////////////////////////

#ifdef _DEBUG
void CMakeFlossView::AssertValid() const
{
    CFormView::AssertValid();
}

void CMakeFlossView::Dump(CDumpContext& dc) const
{
    CFormView::Dump(dc);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

CMakeFlossDoc* CMakeFlossView::GetDocument() // non-debug version is inline
{

```

```

ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CMakeFlossDoc)));
return (CMakeFlossDoc*)m_pDocument;
}
#endif // _DEBUG

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CMakeFlossView message handlers

void CMakeFlossView::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();

    // TODO: Add your specialized code here and/or call the base class
    TRACE("Entering CMakeFlossView::OnInitialUpdate\n");
    OnUpdate(this, NULL, NULL);

    CScrollBar* pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_RED);
    pSB->SetScrollRange(nMin, nMax);
    pSB->SetScrollPos(m_lRed);

    pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_GREEN);
    pSB->SetScrollRange(nMin, nMax);
    pSB->SetScrollPos(m_lGreen);

    pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_BLUE);
    pSB->SetScrollRange(nMin, nMax);
    pSB->SetScrollPos(m_lBlue);

    GetDlgItem(IDC_SHOW_COLOR)->GetWindowRect(&m_RectColor);
    ScreenToClient(&m_RectColor);
    int Border = (m_RectColor.right - m_RectColor.left) / 7;
    m_RectColor.InflateRect(-Border, -Border);

    POSITION pos = m_position;

    CListBox* pLB = (CListBox*) GetDlgItem(IDC_LIST_FLOSS);

    while(pos != NULL) //m_pList->GetTailPosition()
    {
        TRACE("Name = %s\n", m_name);
        pLB->AddString(m_name);
        m_pList->GetNext(pos);

        if (pos)
        {
            CFlossData* pFlossData = (CFlossData*) m_pList->GetAt(pos);
            // m_number = pFlossData->m_number;
            // m_name = pFlossData->m_name;
            // m_lBlue = pFlossData->m_lBlue;
            // m_lGreen = pFlossData->m_lGreen;
            // m_lRed = pFlossData->m_lRed;
        }
    }
}

void CMakeFlossView::OnUpdate(CView* pSender, LPARAM lHint, CObject* pHint)
{
    // TODO: Add your specialized code here and/or call the base class
    TRACE("Entering CMakeFlossView::OnUpdate\n");
    m_pList = GetDocument()->GetList();
    m_position = m_pList->GetHeadPosition();
    GetEntry(m_position);
} เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเผยแพร่ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

void CMakeFlossView::OnDefineColor()
{
    // TODO: Add your control notification handler code here
    CColorDialog ColorDialog;

    if(ColorDialog.DoModal() == IDOK)
        return;
}

////////////////////////////////////
void CMakeFlossView::OnAdd()
{
    // TODO: Add your control notification handler code here
    TRACE("Entering CMakeFlossView::OnAdd\n");
    InsertEntry(m_position);
    GetDocument()->SetModifiedFlag();
    GetDocument()->UpdateAllViews(this);

    CListBox* pLB = (CListBox*) GetDlgItem(IDC_LIST_FLOSS);
    pLB->AddString(m_name);
}

void CMakeFlossView::OnRemove()
{
    // TODO: Add your control notification handler code here
    CListBox* pLB = (CListBox*) GetDlgItem(IDC_LIST_FLOSS);
    TRACE("delname = %s\n", m_name);
    pLB->DeleteString(pLB->FindStringExact(-1, m_name));

    POSITION pos;

    TRACE("Entering CMakeFlossView::OnRemove\n");
    if ((pos = m_position) != NULL)
    {
        m_pList->GetNext(pos);
        if (pos == NULL)
        {
            pos = m_pList->GetHeadPosition();
            TRACE("GetHeadPos = %ld\n", pos);
            if (pos == m_position)
            {
                pos = NULL;
            }
        }
        GetEntry(pos);
        CObject* ps = m_pList->GetAt(m_position);
        m_pList->RemoveAt(m_position);
        delete ps;
        m_position = pos;

        GetDocument()->SetModifiedFlag();
        GetDocument()->UpdateAllViews(this);
    }
    UpdateScrollBar();
}

void CMakeFlossView::OnPrev()
{
    // TODO: Add your control notification handler code here
    POSITION pos;

    TRACE("Entering CMakeFlossView::OnPrev\n");
    if ((pos = m_position) != NULL)
    {
        m_pList->GetPrev(pos);
    }
}

```

เอกสารนี้เป็นทรัพย์สินของสำนักงาน ก.ค.ศ. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใ้แก่กรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (pos)
        {
            GetEntry(pos);
            m_position = pos;
        }
    }
    UpdateScrollBar();
/*
    CClientDC ClientDC(this);
    OnPrepareDC(&ClientDC);
    CRect Rect = m_RectColor;
    ClientDC.LPtoDP(&Rect);
    InvalidateRect(&Rect);
    UpdateWindow();
*/
}

void CMakeFlossView::OnNext()
{
    // TODO: Add your control notification handler code here
    POSITION pos;

    TRACE("Entering CMakeFlossView::OnNext\n");
    if ((pos = m_position) != NULL)
    {
        m_pList->GetNext(pos);
        if (pos)
        {
            GetEntry(pos);
            m_position = pos;
        }
    }
    UpdateScrollBar();

    CClientDC ClientDC(this);
    OnPrepareDC(&ClientDC);
    CRect Rect = m_RectColor;
    ClientDC.LPtoDP(&Rect);
    InvalidateRect(&Rect);
    UpdateWindow();
}

void CMakeFlossView::OnClear()
{
    // TODO: Add your control notification handler code here
    TRACE("Entering CMakeFlossView::OnClear\n");
    ClearEntry();
    UpdateScrollBar();
/*
    CClientDC ClientDC(this);
    OnPrepareDC(&ClientDC);
    CRect Rect = m_RectColor;
    ClientDC.LPtoDP(&Rect);
    InvalidateRect(&Rect);
    UpdateWindow();
*/
}

////////////////////////////////////
void CMakeFlossView::GetEntry(POSITION position)
{
    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    if (position) {
        CMakeFlossView::CMakeFlossData* pFlossData = (CMakeFlossData*)m_pList->GetAt(position);
        m_number = pFlossData->m_number;
    }
}

```

```

    m_name = pFlossData->m_name;
    m_lBlue = pFlossData->m_lBlue;
    m_lGreen = pFlossData->m_lGreen;
    m_lRed = pFlossData->m_lRed;
}
else {
    ClearEntry();
}
UpdateData(FALSE);
}

void CMakeFlossView::InsertEntry(POSITION position)
{
    if (UpdateData(TRUE)) {
        // UpdateData returns FALSE if it detects a user error
        CFlossData* pFlossData = new CFlossData;
        pFlossData->m_number = m_number;
        pFlossData->m_name = m_name;
        pFlossData->m_lBlue = m_lBlue;
        pFlossData->m_lGreen = m_lGreen;
        pFlossData->m_lRed = m_lRed;
        m_position = m_pList->InsertAfter(m_position, pFlossData);
    }
}

void CMakeFlossView::ClearEntry()
{
    TRACE("Entering CMakeFlossView::ClearEntry\n");
    m_number = "";
    m_name = "test";
    m_lBlue = 0;
    m_lGreen = 0;
    m_lRed = 0;
    UpdateData(FALSE);
    ((CDialog*) this)->GotoDlgCtrl(GetDlgItem(IDC_NUMBER));
}

////////////////////////////////////
void CMakeFlossView::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    TRACE("Entering CMakeFlossView::OnHScroll\n");
    // TODO: Add your message handler code here and/or call default
    int nTemp1, nTemp2;

    nTemp1 = pScrollBar->GetScrollPos();
    switch(nSBCode)
    {
    case SB_THUMBPOSITION:
        pScrollBar->SetScrollPos(nPos);
        break;
    case SB_LINEUP:
        nTemp2 = 1;//(nMax - nMin) / 10;
        if((nTemp1 - nTemp2) > nMin)
        {
            nTemp1 -= nTemp2;
        }
        else
        {
            nTemp1 = nMin;
        }
        pScrollBar->SetScrollPos(nTemp1);
        break;
    case SB_LINEDOWN:
        nTemp2 = 1;//(nMax - nMin) / 10;

```

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สามารถทำซ้ำหรือดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if((nTemp1 + nTemp2) < nMax)
    {
        nTemp1 += nTemp2;
    }
    else
    {
        nTemp1 = nMax;
    }
    pScrollBar->SetScrollPos(nTemp1);
}
CScrollBar* pSB;

pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_RED);
m_lRed = pSB->GetScrollPos();

pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_GREEN);
m_lGreen = pSB->GetScrollPos();

pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_BLUE);
m_lBlue = pSB->GetScrollPos();

UpdateData(FALSE);

CClientDC ClientDC(this);
OnPrepareDC(&ClientDC);
CRect Rect = m_RectColor;
ClientDC.LPtoDP(&Rect);
InvalidateRect(&Rect);
UpdateWindow();

// CFormView::OnHScroll(nSBCode, nPos, pScrollBar);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void CMakeFlossView::OnDraw(CDC* pDC)
{
    // TODO: Add your specialized code here and/or call the base class
    TRACE("red = %ld green = %ld blue = %ld\n", m_lRed, m_lGreen, m_lBlue);
    COLORREF Color = RGB(m_lRed, m_lGreen, m_lBlue);
    CBrush Brush(Color);
    pDC->FillRect(&m_RectColor, &Brush);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void CMakeFlossView::OnChangeRed()
{
    // TODO: Add your control notification handler code here
    if(UpdateData(TRUE))
    {
        CScrollBar* pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_RED);
        pSB->SetScrollRange(nMin, nMax);
        pSB->SetScrollPos(m_lRed);
        TRACE("OnChangeRed, Red = %ld\n", m_lRed);
    }
}

void CMakeFlossView::OnChangeGreen()
{
    // TODO: Add your control notification handler code here
    if(UpdateData(TRUE))
    {
        CScrollBar* pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_GREEN);
        pSB->SetScrollRange(nMin, nMax);
        pSB->SetScrollPos(m_lGreen);
        TRACE("OnChangeGreen, Green = %ld\n", m_lGreen);
    }
}

```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของสำนักงานส่งเสริมการค้าในต่างประเทศ (สอ.ท.) กระทรวงพาณิชย์
 ไม่สามารถเผยแพร่หรือใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

void CMakeFlossView::OnChangeBlue()
{
    // TODO: Add your control notification handler code here
    if(UpdateData(TRUE))
    {
        CScrollBar* pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_BLUE);
        pSB->SetScrollRange(nMin, nMax);
        pSB->SetScrollPos(m_lBlue);
        TRACE("OnChangeBlue, Blue = %ld\n", m_lBlue);
    }
}

```

```

void CMakeFlossView::OnDblclkListFloss()
{
    // TODO: Add your control notification handler code here
    CListBox* pLB = (CListBox*) GetDlgItem(IDC_LIST_FLOSS);
    CString str;
    pLB->GetText(pLB->GetCurSel(), str);

    CFlossData* pFlossData;
    POSITION pos = m_pList->GetHeadPosition();
    pFlossData = (CFlossData*) m_pList->GetAt(pos);
    m_name = pFlossData->m_name;

    TRACE("Entering CMakeFlossView::OnDblclkListFloss\n");
    while((m_name != str) && (pos != NULL))
    {
        TRACE("m_name = %s, str = %s, pos = %ld\n", m_name, str, pos);
        m_pList->GetNext(pos);
        pFlossData = (CFlossData*) m_pList->GetAt(pos);
        m_name = pFlossData->m_name;
    }

    if (pos)
    {
        GetEntry(pos);
        m_position = pos;
    }
    UpdateScrollBar();
}
/*
CClientDC ClientDC(this);
OnPrepareDC(&ClientDC);
CRect Rect = m_RectColor;
ClientDC.LPtoDP(&Rect);
InvalidateRect(&Rect);
UpdateWindow();
*/
}

```

```

void CMakeFlossView::UpdateScrollBar()
{
    CScrollBar* pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_RED);
    pSB->SetScrollRange(nMin, nMax);
    pSB->SetScrollPos(m_lRed);

    pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_GREEN);
    pSB->SetScrollRange(nMin, nMax);
    pSB->SetScrollPos(m_lGreen);
    pSB = (CScrollBar*) GetDlgItem(IDC_SCROLLBAR_BLUE);
    pSB->SetScrollRange(nMin, nMax);
}

```

```
pSB->SetScrollPos(m_lBlue);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Ori Gurewich and Nathan Gurewich, teach yourself Visual C++ 4 in 21 days : Sams Publishing
2. Michael J. Young, Mastering Microsoft Visual C++ 4 : SYBEX Inc.
3. โชคชัย เทชพรรุ่ง, การเขียนโปรแกรม Visual C++ เวอร์ชัน 1.5 บนวินโดวส์ : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)
4. A & P, นิตยสาร Computer Today ปีที่ 5 ฉบับที่ 5 เดือนพฤศจิกายน 2538



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้