

## การพัฒนาซียูไรเตอร์ปลั๊กอิน

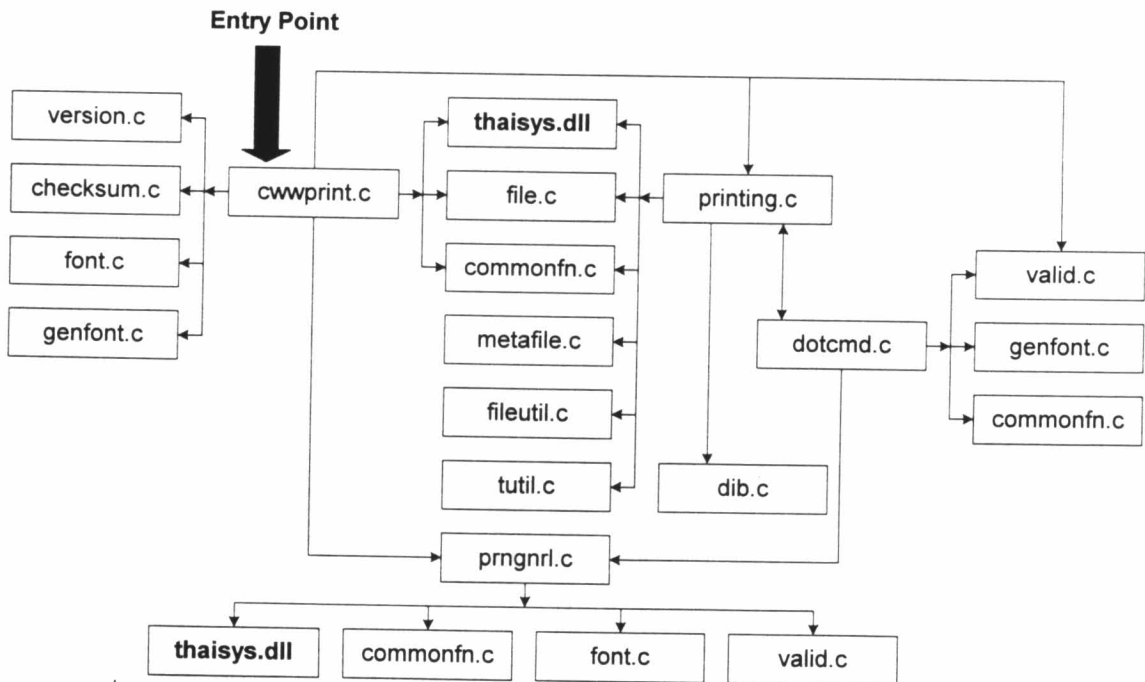
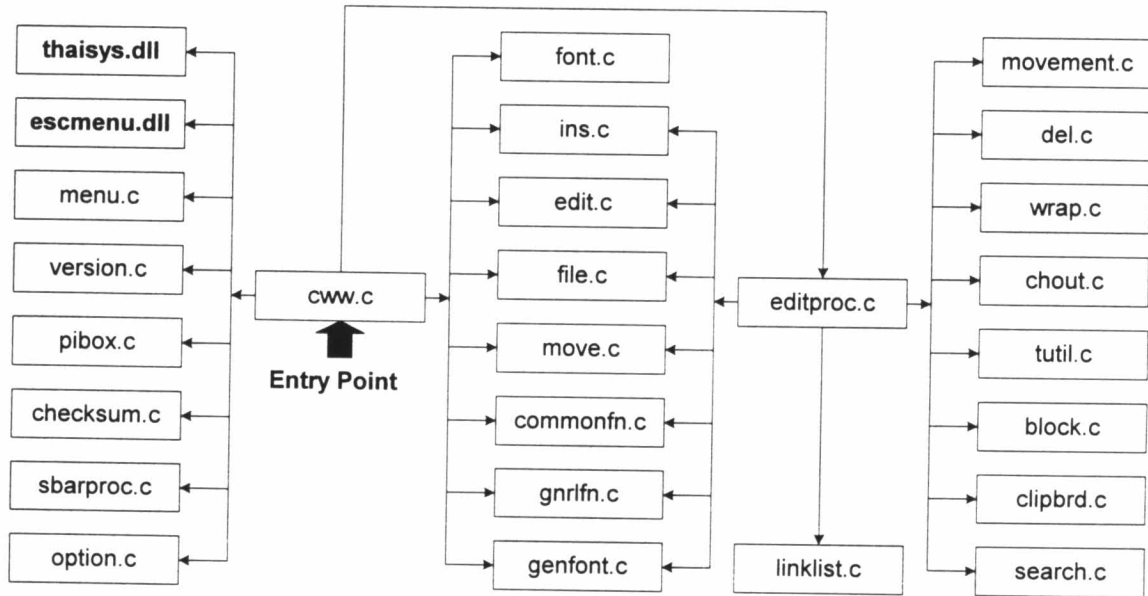
### คุณสมบัติของซียูไรเตอร์ปลั๊กอิน

1. ทำงานบนเครื่องพีซีที่
  - 1.1. มีหน่วยประมวลผลตระกูล 80x86 ของบริษัทอินเทล
  - 1.2. มีไมโครซอฟท์วินโดวส์ 3.1 รุ่นภาษาไทยเป็นระบบปฏิบัติการและมีวิน 32 เอสรุ่น 1.30 หรือมากกว่าขึ้นไปติดตั้งอยู่
  - 1.3. มีเมาส์ติดตั้งอยู่
  - 1.4. มีเน็ตสเคปรุ่น 2.0 หรือมากกว่าขึ้นไปติดตั้งอยู่
  - 1.5. มี NPCWW.INI ติดตั้งอยู่ในสารบบที่ติดตั้งไมโครซอฟท์วินโดวส์ มี THAISYS.DLL CWWSPIN.DLL ติดตั้งอยู่ในสารบบย่อย SYSTEM ของสารบบที่ติดตั้งไมโครซอฟท์วินโดวส์และมี CWWSYS.BIN ติดตั้งอยู่ในสารบบย่อย PLUGINS ของสารบบที่ติดตั้งเน็ตสเคป
  
2. สนับสนุนการทำงานต่อไปนี้
  - 2.1. แสดงผลและพิมพ์เอกสารที่สร้างจากซียูไรเตอร์รุ่น 1.52 หรือจุฬาริกรุ่น 77 ได้ทั้งแบบฝังและแบบเต็มหน้า
  - 2.2. เปลี่ยนแบบอักษรที่ใช้แสดงผลได้
  - 2.3. ค้นหาข้อมูลในเอกสารได้
  - 2.4. ตัดคำได้
  - 2.5. เลือกรหัสภาษาไทยที่ใช้ในการแสดงผลระหว่างสมอ. และเพชรได้
  - 2.6. เลือกข้อความในเมนูและกล่องคำตอบ (dialog box) เป็นไทยหรืออังกฤษได้ ยกเว้นกล่องคำตอบเกี่ยวกับ (about) และกล่องข้อความ (message box)

### เครื่องมือที่ใช้

1. ชุดพัฒนาเน็ตสเคปปลั๊กอินรุ่น 2.0
2. ไมโครซอฟท์วิซวลซีพลัสพลัสรุ่น 1.52
3. โปรแกรมภาษาต้นฉบับของจู่ฟาจาร์รุ่น 77

ความสัมพันธ์ของแฟ้มข้อมูลในโปรแกรมภาษาต้นฉบับของจุฬาริกรุ่น 77



รูปที่ 4- 1 ความสัมพันธ์ระดับบนระหว่างแฟ้มข้อมูลในโปรแกรมภาษาต้นฉบับของจุฬาริกรุ่น 77

ในการพัฒนาซียูไรเตอร์ปลั๊กอินจำเป็นจะต้องทำการศึกษาถึงหน้าที่ของแฟ้มข้อมูลต่างๆภายในโปรแกรมภาษาต้นฉบับของจุฬาริกรุ่น 77 พร้อมกับศึกษาความสัมพันธ์ของแฟ้มข้อมูลเหล่านั้นด้วย ทั้งนี้เพื่อให้ทราบว่าควรจะนำส่วนใดบ้างมาใช้ในการพัฒนาซียูไรเตอร์ปลั๊กอิน รูปที่ 4- 1 จะแสดงถึงความสัมพันธ์ระดับบนระหว่างแฟ้มข้อมูล

ต่างๆภายในโปรแกรมภาษาต้นฉบับของจุฬารีก์รุ่น 77 (สำหรับความสัมพันธ์ระดับอื่นๆสามารถดูได้จากภาคผนวก ก) ความหมายของสัญลักษณ์ต่างๆที่ปรากฏอยู่ในรูปมีดังนี้

1. กล่องสี่เหลี่ยมจะบอกถึงเพิ่มข้อมูลในโปรแกรมภาษาต้นฉบับของจุฬารีก์รุ่น 77 (ยกเว้นกล่องที่มีตัวหนังสือเป็นตัวหนาจะบอกถึงคลังโปรแกรมเชื่อมโยงแบบพลวัต) กล่องสี่เหลี่ยมที่มีชื่อเดียวกันจะหมายถึงเพิ่มข้อมูลเดียวกัน (ยกเว้นเพิ่มข้อมูล FILE.C ที่จะมีอยู่ทั้งในโปรแกรม CWW และ CWWPRINT)
2. ลูกศรขนาดใหญ่จะบอกถึงจุดเริ่มต้นการทำงานซึ่งในรูปนี้จะมีอยู่ 2 จุดเนื่องจากว่าภายในโปรแกรมภาษาต้นฉบับของจุฬารีก์รุ่น 77 จะประกอบไปด้วย 2 โปรแกรมคือ CWW และ CWWPRINT ซึ่งทำงานอิสระจากกัน
3. ลูกศรขนาดเล็กจะบอกถึงการถูกเรียกใช้งาน (เพิ่มข้อมูลที่ถูกเรียกใช้จะอยู่ด้านที่ลูกศรพุ่งเข้าหา)

หน้าที่ของแต่ละเพิ่มข้อมูลภายในรูปที่ 4- 1 แสดงอยู่ในตารางที่ 4- 1 ถึงตารางที่ 4- 3

เพิ่มข้อมูล	หน้าที่ของเพิ่มข้อมูล
CHECKSUM.C	เก็บฟังก์ชันที่ทำหน้าที่คำนวณผลรวมของส่วนทรัพยากรในโปรแกรม
COMMONFN.C	เก็บฟังก์ชันที่ทำหน้าที่ทั่วไป เช่นแสดงข้อความบอกถึงข้อผิดพลาด
DIB.C	เก็บฟังก์ชันที่ทำหน้าที่ในการแสดงรูปภาพที่อยู่ในรูปแบบบิตแมป (bitmap)
FILEUTIL.C	เก็บฟังก์ชันที่ทำหน้าที่ทั่วไปเกี่ยวกับเพิ่มข้อมูล เช่นตรวจสอบความถูกต้องของรูปภาพในเอกสาร
FONT.C	เก็บฟังก์ชันที่ทำหน้าที่ในการเปลี่ยนแบบอักษร จำนวนอักษรต่อนิ้ว และระยะห่างที่ใช้
GENFONT.C	เก็บฟังก์ชันที่ทำหน้าที่ในการสร้างแบบอักษรขนาดคงที่
METAFILE.C	เก็บฟังก์ชันที่ทำหน้าที่ในการแสดงรูปภาพที่อยู่ในรูปแบบเมตาไฟล์ (metafile)
TUTIL.C	เก็บฟังก์ชันที่ทำหน้าที่เกี่ยวกับภาษาไทย เช่นหาความยาว เปลี่ยนรหัสภาษาไทย หรือหาระดับของอักขระภาษาไทย
VERSION.C	เก็บฟังก์ชันที่ทำหน้าที่แสดงรูปภาพของจุฬารีก์ พร้อมกับเล่นเพลง

ตารางที่ 4- 1 หน้าที่ของเพิ่มข้อมูลที่ปรากฏอยู่ในรูปที่ 4- 1 เฉพาะส่วนที่มีการใช้งาน  
ทั้งในโปรแกรม CWW และ CWWPRINT

แฟ้มข้อมูล	หน้าที่ของแฟ้มข้อมูล
BLOCK.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการบล็อก (block) เช่นตรวจสอบว่ามีบล็อกอยู่หรือไม่ ทำสำเนาข้อมูลในบล็อก
CHOUT.C	เก็บฟังก์ชันที่ทำหน้าที่ในการแสดงผลหนึ่งบรรทัดเอกสาร
CLIPBRD.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการเกี่ยวกับคลิปบอร์ด (clipboard) เช่นสำเนาข้อมูลไปและมา
CWW.C	เก็บฟังก์ชัน WinMain ฟังก์ชันอื่นๆที่ทำหน้าที่ในการเริ่มต้นระบบและยุติระบบ ฟังก์ชันที่ทำหน้าที่สร้างเมนู แสดงผลเมนู ทำลายเมนู อ่านตัวเลือกจากแฟ้มข้อมูล CWW.INI และฟังก์ชันที่ทำหน้าที่เป็นวินโดวโพรซีเจอร์ (window procedure)
DEL.C	เก็บฟังก์ชันที่ทำหน้าที่ลบข้อมูลออกจากเอกสาร เช่นลบบรรทัด
EDIT.C	เก็บฟังก์ชันที่ทำหน้าที่ในการแสดงผลหน้าเอกสาร เตรียมข้อมูลสำหรับแก้ไข
EDITPROC.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการกับการกดแป้นต่างๆ การกดเมาส์ บรรทัดแสดงสถานะภาพ (status line) ไม้บรรทัด (ruler)
FILE.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการเอกสารซียูไรเตอร์ เช่นอ่าน เขียนเอกสาร
FILEOPT.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการตัวเลือก (option) ท้ายเอกสารซียูไรเตอร์ เช่นอ่าน เขียนตัวเลือก
GNRLFN.C	เก็บฟังก์ชันที่ทำหน้าที่ทั่วไป เช่นตั้งค่าเริ่มต้นโครงสร้างข้อมูล
INS.C	เก็บฟังก์ชันที่ทำหน้าที่แทรก พิมพ์ทับข้อมูลในเอกสาร จัดการกับตัวย่อ
LINKLIST.C	เก็บฟังก์ชันที่ทำหน้าที่ในการเพิ่มและลบข้อมูลในรายการเชื่อมโยง
MENU.C	เก็บฟังก์ชันที่ทำหน้าที่เลือกเมนูโดยการกดแป้นที่ตรงกับตัวอักษรในเมนูที่ขีดเส้นใต้ไว้
MOVE.C	เก็บฟังก์ชันที่ทำหน้าที่เคลื่อนที่ไปยังส่วนของเอกสาร เช่นไปต้นเอกสาร ท้ายเอกสาร
MOVEMENT.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการการเคลื่อนที่ที่เกิดจากการใช้แป้นลูกศร แป้น PgUp แป้น PgDn
OPTION.C	เก็บฟังก์ชันที่ทำหน้าที่แก้ไขตัวเลือก

ตารางที่ 4- 2 หน้าที่ของแฟ้มข้อมูลที่ปรากฏอยู่ในรูปที่ 4- 1 เฉพาะส่วนที่มีการใช้งาน  
ในโปรแกรม CWW

แฟ้มข้อมูล	หน้าที่ของแฟ้มข้อมูล
PIBOX.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการหน้าต่างลูกที่อยู่บนแถบแสดงสถานะภาพ (status bar)
SBARPROC.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการแถบแสดงสถานะภาพ
SEARCH.C	เก็บฟังก์ชันที่ทำหน้าที่ในการค้นหา ค้นหาและเปลี่ยน
WRAP.C	เก็บฟังก์ชันที่ทำหน้าที่ในการตัดคำ จัดย่อหน้าใหม่

ตารางที่ 4- 2 หน้าที่ของแฟ้มข้อมูลที่ปรากฏอยู่ในรูปที่ 4- 1 เฉพาะส่วนที่มีการใช้งาน  
ในโปรแกรม CWW (ต่อ)

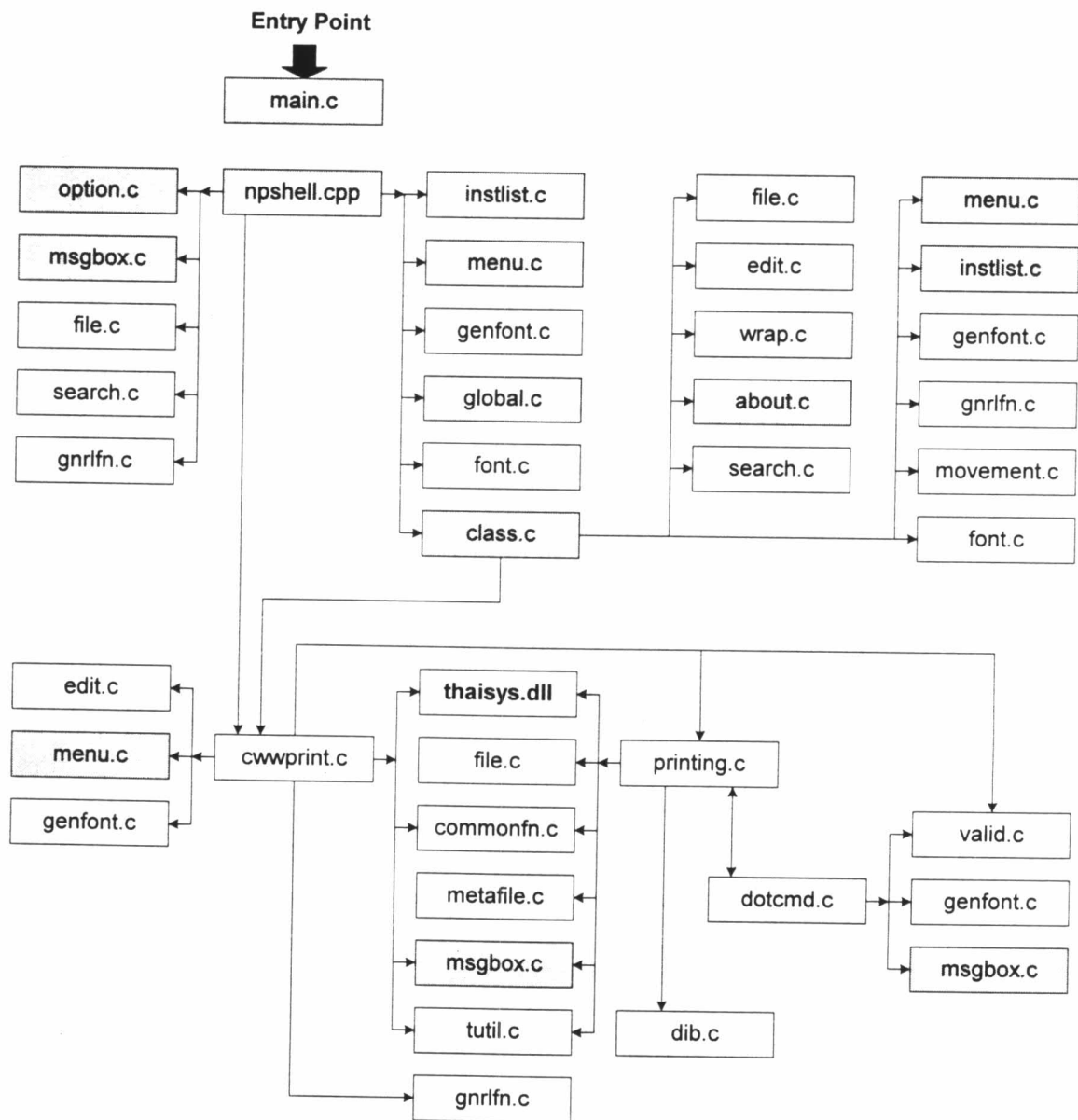
แฟ้มข้อมูล	หน้าที่ของแฟ้มข้อมูล
CWWPRINT.C	เก็บฟังก์ชัน WinMain ฟังก์ชันอื่นๆที่ทำหน้าที่ในการเริ่มต้นระบบและยุติระบบ แสดงกล่องโต้ตอบ อ่านตัวเลือกจากแฟ้มข้อมูล CWW.INI และฟังก์ชันที่ทำหน้าที่เป็นวินโดวโพรซีเยอร์
DOTCMD.C	เก็บฟังก์ชันที่ทำหน้าที่ในการจัดการกับคำสั่งแบบจุด (dot command) และการผสานจ่าหน้าจดหมาย (mail merge)
FILE.C	เก็บฟังก์ชันที่ทำหน้าที่จัดการเอกสารซียูไรเตอร์ เช่นอ่านตัวเลือกท้ายเอกสาร
PRINTING.C	เก็บฟังก์ชันที่ทำหน้าที่ในการพิมพ์เอกสาร
PRNGNRL.C	เก็บฟังก์ชันที่ทำหน้าที่ต่างๆไป เช่นตั้งค่าเริ่มต้นโครงสร้างข้อมูล
VALID.C	เก็บฟังก์ชันที่ทำหน้าที่ในการตรวจสอบค่าของจำนวนบรรทัดต่อหน้า ค่าของขอบขวา

ตารางที่ 4- 3 หน้าที่ของแฟ้มข้อมูลที่ปรากฏอยู่ในรูปที่ 4- 1 เฉพาะส่วนที่มีการใช้งาน  
ในโปรแกรม CWWPRINT

### การพัฒนาซียูไรเตอร์ปลั๊กอิน

หลังจากทำการศึกษาถึงหน้าที่และความสัมพันธ์ระหว่างแฟ้มข้อมูลในโปรแกรมภาษาต้นฉบับของจุฬาริกรุ่น 77 แล้วขั้นตอนถัดมาคือการตัดเฉพาะส่วนที่จำเป็นต้องใช้งานออกมาและทำการแก้ไขเท่าที่จำเป็น (การแก้ไขนี้ไม่ได้ทำให้หน้าที่ของแฟ้มข้อมูลนั้นเปลี่ยนไป เช่นตัดฟังก์ชันหรือคำสั่งที่ไม่ได้ใช้งานออกไป) พร้อมกับทำการพัฒนาส่วนอื่นๆที่จำเป็นเพิ่มเติมเข้าไปเพื่อให้ทำงานได้ตามคุณสมบัติที่กล่าวไว้ข้างต้น รูปที่ 4- 2 จะแสดงถึงความสัมพันธ์ระดับบนของแฟ้มข้อมูลที่นำมาจากโปรแกรมภาษาต้นฉบับของจุฬาริกรุ่น 77 และแฟ้มข้อมูลที่พัฒนาเพิ่มเติมขึ้นมา (สำหรับความ

สัมพันธ์ระดับอื่นๆสามารถดูได้จากภาคผนวก ก) ความหมายของสัญลักษณ์ต่างๆที่ปรากฏอยู่ในรูปจะเหมือนกับในรูปที่ 4-1 ยกเว้นกล่องสี่เหลี่ยมสีเทาจะใช้บอกถึงเพิ่มข้อมูลที่พัฒนาเพิ่มเติมขึ้นมา



รูปที่ 4-2 ความสัมพันธ์ระดับบนของเพิ่มข้อมูลในซียูไรเตอร์ปลั๊กอิน

ในกลุ่มของเพิ่มข้อมูลที่พัฒนาเพิ่มเติมขึ้นมาจะมีอยู่ 2 เพิ่มข้อมูลที่ทำหน้าที่สำคัญคือเพิ่มข้อมูล NPSHELL.CPP เพิ่มข้อมูลนี้จะทำหน้าที่ในการติดต่อสื่อสารกับเน็ตสเคปซึ่งการทำงานทั้งหมดของซียูไรเตอร์ปลั๊กอินจะเริ่มต้นขึ้นที่เพิ่มข้อมูลนี้เนื่องจากเป็นช่องทางในการได้มาซึ่งเอกสารและหน้าต่างที่ใช้ในการแสดงผล อีกเพิ่มข้อมูลคือ CLASS.C ซึ่งทำหน้าที่จัดการกับข้อความต่างๆที่ไม่โครซอฟท์วินโดวส์ส่งมาให้ ข้อความเหล่านี้เป็นช่องทางในการที่ซียูไรเตอร์ปลั๊กอินจะใช้เพื่อโต้ตอบกับระบบและผู้ใช้

เพิ่มข้อมูลอื่นๆนอกเหนือจาก 2 เพิ่มข้อมูลที่กำลังกล่าวไปแล้วจะทำหน้าที่รองรับการทำงานของทั้ง 2 เพิ่มข้อมูลนี้ รายละเอียดของเพิ่มข้อมูลทั้งหมดที่พัฒนาเพิ่มเติมขึ้นมา มีดังนี้ (การอธิบายจะเริ่มจากเพิ่มข้อมูล NPSHELL.CPP และ CLASS.C ก่อนจากนั้นจะตามด้วยเพิ่มข้อมูลที่เหลือเรียงลำดับตามชื่อ)

1. NPSHELL.CPP เพิ่มข้อมูลนี้จะอยู่ในชุดพัฒนาเน็ตสเคปปลั๊กอินและเป็นที่ยึดโยงที่อยู่กับฟังก์ชันที่อยู่ในเอพีไอ กลุ่มเน็ตสเคปปลั๊กอินไว้ (ฟังก์ชันที่มีชื่อขึ้นต้นด้วย NPP) ซึ่งใช้สำหรับติดต่อสื่อสารกับเน็ตสเคป ภายในเพิ่มข้อมูลนี้ จะมีโครงของแต่ละฟังก์ชันมาให้หน้าที่ของเราคือเติมการทำงานที่เหมาะสมกับแต่ละฟังก์ชันลงไป การทำงานของแต่ละฟังก์ชันในเพิ่มข้อมูลนี้มีดังนี้ (การประกาศของแต่ละฟังก์ชันในเพิ่มข้อมูลนี้ดูได้จากบทที่ 3)

1.1. ฟังก์ชัน NPP\_Initialize ฟังก์ชันนี้จะถูกเรียกเมื่อเน็ตสเคปทำการบรรจุซึยูไรเตอร์ปลั๊กอิน ลงสู่หน่วยความจำ ภายในฟังก์ชันนี้จะทำการอ่านตัวเลือกจากเพิ่มข้อมูล NPCWW.INI มาเก็บในโครงสร้างข้อมูล Option (ฟังก์ชัน ReadOptionINI ในเพิ่มข้อมูล OPTION.C) ซึ่งเป็นโครงสร้างข้อมูลในโปรแกรมภาษาต้นฉบับของ จูฬจารีกรุ่น 77 โครงสร้างข้อมูลนี้จะใช้สำหรับการตั้งค่าเริ่มต้นภายในฟังก์ชัน InitPluginInstance (เพิ่มข้อมูล INSTLIST.C) หลังจากนั้นทำการตั้งค่าเริ่มต้นให้กับตัวแปรร่วมต่างๆ (ฟังก์ชัน InitGlobal ในเพิ่มข้อมูล GLOBAL.C) สร้างเมนู (ฟังก์ชัน InitMenu ในเพิ่มข้อมูล MENU.C) และทำการลงทะเบียน (register) วินโดว์คลาส (window class) (Petzold, 1992) ที่มีการใช้งาน (ฟังก์ชัน RegisterAllClass ในเพิ่มข้อมูล CLASS.C)

1.2. ฟังก์ชัน NPP\_Shutdown ฟังก์ชันนี้จะถูกเรียกเมื่อเน็ตสเคปทำการนำซึยูไรเตอร์ปลั๊กอิน ออกจากหน่วยความจำ ภายในฟังก์ชันนี้จะทำหน้าที่ตรงข้ามกับฟังก์ชัน NPP\_Initialize คือทำการยกเลิกการลงทะเบียนวินโดว์คลาส (ฟังก์ชัน UnRegisterAllClass ในเพิ่มข้อมูล CLASS.C) ทำลายเมนู (ฟังก์ชัน DeInitMenu ในเพิ่มข้อมูล MENU.C) และทรัพยากรต่างๆที่ตัวแปรร่วมใช้อยู่ (ฟังก์ชัน DeInitGlobal ในเพิ่มข้อมูล GLOBAL.C)

1.3. ฟังก์ชัน NPP\_New ฟังก์ชันนี้จะถูกเรียกเมื่อเน็ตสเคปทำการสร้างอินสแตนซ์ขึ้นมาเพื่อใช้ทำงาน กับข้อมูลที่ได้รับซึ่งในที่นี้คือเอกสารของซึยูไรเตอร์ หน้าที่ของฟังก์ชันนี้คือสร้างข้อมูลที่จำเป็นต่อการทำงานกับเอกสารนี้ ขึ้นมา ข้อมูลที่สร้างขึ้นมานี้คือโครงสร้างข้อมูล PluginInstance ซึ่งมีรายละเอียดดังนี้

```
typedef struct _PluginInstance PluginInstance;
struct _PluginInstance
{
    HWND        hwnd;
    uint16      fMode;
```



```

FARPROC    lpfnOldWndProc;
PluginInstance *pNext;
NPP        instance;
BOOL       bDataReady;
CwEditVar  cwevar;
OptionOverride ovr;
};

```

ค่าของแต่ละเขตข้อมูลในโครงสร้างข้อมูลนี้มีดังนี้

1.3.1. hwnd และ lpfnOldWndProc จะเก็บแฮนเดิลและตำแหน่งของวินโดวส์โพธิ์เซอร์ของหน้าต่างที่อยู่ในพารามิเตอร์ window ของฟังก์ชัน NPP\_SetWindow (ดูรายละเอียดเพิ่มในหัวข้อ 1.4)

1.3.2. fMode จะเก็บค่าที่อยู่ในพารามิเตอร์ mode ค่านี้จะใช้สำหรับตรวจสอบการพิมพ์แบบเต็มหน้า

1.3.3. pNext จะชี้ไปยังตำแหน่งของโครงสร้างข้อมูลที่อยู่ถัดไปในรายการเชื่อมโยง (ดู รูปที่ 4- 3)

1.3.4. instance จะเก็บค่าที่อยู่ในพารามิเตอร์ instance (ดูรูปที่ 4- 3)

1.3.5. bDataReady เป็นค่าตรรกะบอกว่าได้ข้อมูลในเอกสารมาครบหรือยัง ถ้าเป็นจริงแสดงว่าได้ข้อมูลในเอกสารมาครบแล้ว (ดูรายละเอียดเพิ่มในหัวข้อ 1.8)

1.3.6. cwevar เป็นโครงสร้างข้อมูลที่ใช้ในการจัดการกับเอกสาร โครงสร้างข้อมูลนี้ นำมาจากโปรแกรมภาษาต้นฉบับของจุฬารีกุ่น 77 โดยมีการเพิ่มข้อมูลบางอย่างเข้าไปเพื่อให้เหมาะกับการทำงานของซียูไรเตอร์ปลั๊กอิน โครงสร้างข้อมูลของ cwevar มีดังนี้

```

typedef struct CwEditVartag
{
    BOOL    stdcode;
    BOOL    wordwrap;
    int     thailanguage;
    int     leftmar;
    int     rightmar;
    int     lineperpage;
    int     cpi;
    int     spacing;
    char    fontname[LF_FACESIZE];
};

```

```

LOGFONT    lf[MAXLOGFONT];
BYTE       fonttab[256];
RECT       wind;
int        xSpe;
int        ySpe;
unsigned int  lineno;
int        firstcol;
Lpline_node  sentinel;
Lpline_node  curline;
Lpline_node  curpage;
BOOL       bGenNewFont;
BOOL       thaimode;
BOOL       thaifont;
BYTE       tonemarkcode[15];
BYTE       abovecode[4];
BYTE       above1code[3];
BYTE       belowcode[3];
BYTE       vowelcode[2];
BYTE       fontidx;
int        normalwidth;
int        xCharWidth;
int        yCharHeight;
int        yBaseline;
int        yDescent;
int        yCaret;
HFONT      hFontNormal;
HFONT      hFontShow;
LOGFONT    nlf;
font_attr  fontcur, fontused;
LOGFONT    enlf[MAXLOGFONT];
HFONT      hFont[MAXLOGFONT];
int        chWidthTab[256];

```

```

        BOOL        dispblock;
        Location    blkbegin;
        Location    blkend;
        HFONT       hFontOrg;
        HWND        hWndEdit;
        BOOL        frame;
        HWND        hWndFrame;
        int         totalrow;
        int         maxthaicol;
        int         maxengcol;
        int         maxfixcol;
        int         maxvscri;
        int         posvscri;
        int         maxhscri;
        int         poshscri;
        char        filename[_MAX_PATH];
    } CwEditVar;

```

จากเขตข้อมูล stdcode ถึง hWndEdit เป็นเขตข้อมูลที่อยู่ในโปรแกรมภาษาต้นฉบับของจุฬารีกูร่น 77 และตั้งแต่เขตข้อมูล frame เป็นต้นมาคือส่วนที่เพิ่มเติมเข้าไปซึ่งมีค่าดังนี้

1.3.6.1. frame เป็นค่าตรรกะใช้บอกว่ามีการครอบหน้าต่างหรือไม่ ถ้าเป็นจริงแสดงว่าต้องการให้มีกรอบ (ดูรายละเอียดเพิ่มในฟังก์ชัน FrameWndProc)

1.3.6.2. hWndFrame เก็บแฮนเดิลของหน้าต่างที่ทำหน้าที่เป็นกรอบ (ดูรายละเอียดเพิ่มในหัวข้อ 1.4)

1.3.6.3. totalrow เก็บจำนวนบรรทัดทั้งหมดของเอกสาร จำนวนบรรทัดนี้จะใช้ในการหาว่าจำเป็นต้องมีแถบเลื่อนแนวตั้ง (vertical scroll bar) หรือไม่

1.3.6.4. maxthaicol maxengcol และ maxfixcol เก็บจำนวนคอลัมน์ที่มากที่สุดในเอกสารถ้านับแบบไทย (นับแบบจัดระดับ) นับแบบอังกฤษ (นับแบบไม่จัดระดับ) และนับเฉพาะรูปภาพ จำนวนคอลัมน์นี้ (ตัวใดตัวหนึ่ง) จะใช้ในการหาว่าจำเป็นต้องมีแถบเลื่อนแนวนอน (horizontal scroll bar) หรือไม่ จำนวนคอลัมน์แบบไทยจะใช้ในกรณีที่แบบอักษรที่ใช้แสดงผลมีภาษาไทย จำนวนคอลัมน์แบบอังกฤษจะใช้ในกรณีที่แบบอักษรที่ใช้แสดงผลเป็นไม่มีภาษาไทย จำนวนคอลัมน์เฉพาะรูปภาพจะใช้ในกรณีที่มีการตัดคำ

1.3.6.5. maxvscri posvscri เก็บค่ามากที่สุดและตำแหน่งปัจจุบันของแถบเลื่อนแนว

ตั้ง

1.3.6.6. maxhscri poshscri เก็บค่ามากที่สุดและตำแหน่งปัจจุบันของแถบเลื่อนแนวนอน

นอน

1.3.6.7. filename เก็บชื่อแฟ้มข้อมูลชั่วคราวซึ่งใช้เก็บเอกสารที่ได้จากฟังก์ชัน

NPP\_Write

1.3.7. ovr เป็นโครงสร้างข้อมูลที่ใช้บอกว่ามีเขตข้อมูลใดบ้างใน cwevar ที่ได้รับค่าเริ่มต้นจากรหัสคำสั่ง <EMBED> (การส่งค่าโดยใช้รหัสคำสั่ง <EMBED> สามารถดูได้จากภาคผนวก ข) ทั้งนี้เนื่องจากว่าเขตข้อมูล stdcode cpi spacing และ fontname ใน cwevar สามารถรับค่าเริ่มต้นจากหลายแหล่งได้คือจากตัวเลือกท้ายเอกสารและจากรหัสคำสั่ง <EMBED> ดังนั้นจึงจำเป็นต้องมีวิธีในการตัดสินใจว่าจะใช้ค่าจากแหล่งไหน ในซียูไรเตอร์ปลั๊กอินจะกำหนดให้ใช้ค่าจากรหัสคำสั่ง <EMBED> ถ้าพบว่ามีการกำหนดไว้ และเพื่อป้องกันการรับค่าจากตัวเลือกท้ายเอกสารอีก (เอกสารจะได้รับหลังจากฟังก์ชันนี้ถูกเรียกแล้ว) จึงมีการตั้งค่าในโครงสร้างข้อมูลนี้ไว้เพื่อให้รู้ว่าได้รับค่าแล้ว โครงสร้างข้อมูลของ ovr มีดังนี้

```
typedef struct
{
    BOOL stdcode;
    BOOL cpi;
    BOOL spacing;
    BOOL fontname;
} OptionOverride;
```

ค่าของแต่ละเขตข้อมูลในโครงสร้างข้อมูลนี้มีดังนี้

1.3.7.1. stdcode เป็นค่าตรรกะใช้บอกว่าเขตข้อมูล stdcode ใน cwevar ได้รับค่าเริ่มต้นจากรหัสคำสั่ง <EMBED> หรือไม่ ถ้าเป็นจริงแสดงว่าได้รับค่า

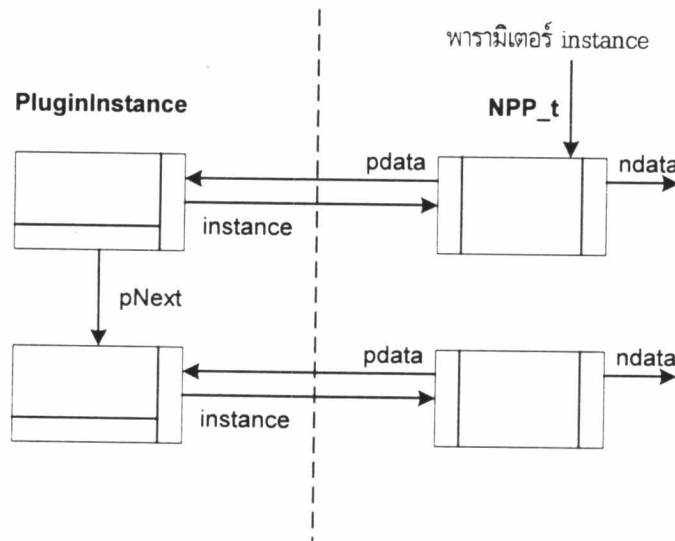
1.3.7.2. cpi เป็นค่าตรรกะใช้บอกว่าเขตข้อมูล cpi ใน cwevar ได้รับค่าเริ่มต้นจากรหัสคำสั่ง <EMBED> หรือไม่ ถ้าเป็นจริงแสดงว่าได้รับค่า

1.3.7.3. spacing เป็นค่าตรรกะใช้บอกว่าเขตข้อมูล spacing ใน cwevar ได้รับค่าเริ่มต้นจากรหัสคำสั่ง <EMBED> หรือไม่ ถ้าเป็นจริงแสดงว่าได้รับค่า

1.3.7.4. fontname เป็นค่าตรรกะใช้บอกว่าเขตข้อมูล fontname ใน cwevar ได้รับค่าเริ่มต้นจากรหัสคำสั่ง <EMBED> หรือไม่ ถ้าเป็นจริงแสดงว่าได้รับค่า

หลังจากโครงสร้างข้อมูล PluginInstance ถูกสร้างขึ้นมาแล้วฟังก์ชัน NPP\_New จะทำการตั้งค่าเริ่มต้นให้กับโครงสร้างข้อมูลนี้ (ฟังก์ชัน InitPluginInstance ในแฟ้มข้อมูล INSTLIST.C) พร้อมกับนำโครงสร้างข้อมูลนี้ไปต่อท้ายรายการเชื่อมโยงซึ่งใช้เชื่อมโครงสร้างข้อมูล PluginInstance ทุกอันที่สร้างขึ้น (ฟังก์ชัน

AssociateInstance ในเพิ่มข้อมูล INSTLIST.C) และเชื่อมโยงโครงสร้างข้อมูลนี้เข้าเป็นส่วนหนึ่งของอินสแตนซ์ด้วย (ดูรูปที่ 4- 3)



รูปที่ 4- 3 รายการเชื่อมโยงของโครงสร้างข้อมูล PluginInstance

1.4. ฟังก์ชัน NPP\_SetWindow ฟังก์ชันนี้จะถูกเรียกเมื่อเน็ตสเคปต้องการให้ข้อมูลเกี่ยวกับหน้าต่างที่ใช้แสดงผลแก่ซียูไรเตอร์ปลั๊กอิน ภายในฟังก์ชันนี้จะทำหน้าที่จัดการเกี่ยวกับหน้าต่างที่ใช้ในการแสดงผลเอกสารและแสดงกรอบ การจัดการนี้ทั้งการสร้างและทำลายหน้าต่างขึ้นอยู่กับสถานะภาพปัจจุบันและค่าในพารามิเตอร์ window ซึ่งสามารถแบ่งได้เป็น 4 กรณีคือ

1.4.1. กรณีค่าของ instance->pdata->hwnd เป็น NULL และค่าของ window->window เป็น NULL ในกรณีนี้จะไม่มีการทำงานใดๆ เนื่องจากเน็ตสเคปไม่ได้ส่งแฮนเดิลของหน้าต่างมาให้ (window->window เป็น NULL) และตัวซียูไรเตอร์ปลั๊กอินเองก็ไม่มีหน้าต่างที่ใช้แสดงผลอยู่ (instance->pdata->hwnd เป็น NULL)

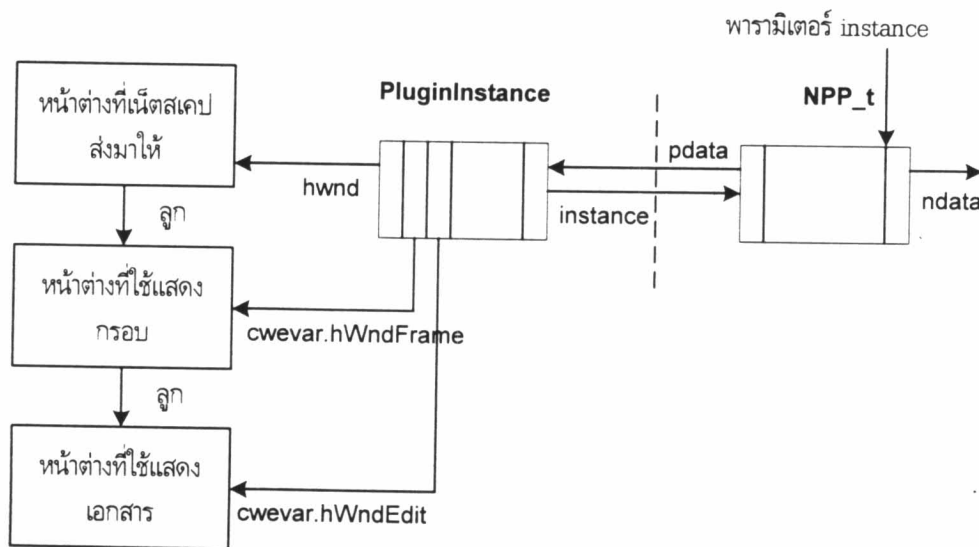
1.4.2. กรณีค่าของ instance->pdata->hwnd เป็น NULL แต่ค่าของ window->window ไม่เป็น NULL ในกรณีนี้แสดงว่าเน็ตสเคปได้ส่งแฮนเดิลของหน้าต่างที่ใช้สำหรับแสดงผลมาให้ ฟังก์ชันนี้จะทำการเก็บค่าแฮนเดิลนี้ไว้ใน instance->pdata->hwnd พร้อมกับทำซับคลาส (subclass) (Petzold, 1992) หน้าต่างนี้โดยเก็บตำแหน่งของวินโดวโพรซีเยอร์เก่าไว้ใน instance->pdata->lpfnOldWndProc และตั้งฟังก์ชัน PluginSubClass (ในเพิ่มข้อมูล CLASS.C) เป็นวินโดวโพรซีเยอร์แทน หลังจากนั้นจะทำการสร้างหน้าต่างลูก (child window) ของหน้าต่างใน window->window เพื่อใช้สำหรับแสดงกรอบขึ้นมาโดยให้มีขนาดของหน้าต่างเท่ากับขนาดของหน้าต่างพ่อ (parent window) และเก็บแฮนเดิลของหน้าต่างลูกนี้ใน instance->pdata->cwevar.hWndFrame

เมื่อสร้างหน้าต่างที่ใช้แสดงกรอบแล้ว ฟังก์ชันนี้จะสร้างหน้าต่างลูกของหน้าต่างที่ใช้แสดงกรอบอีกเพื่อใช้ในการแสดงผลเอกสาร ขนาดของหน้าต่างใหม่นี้จะเท่ากับหน้าต่างพ่อถ้าค่าของ instance->

pdata->cwevar.frame เป็นเท็จ ไม่เช่นนั้นขนาดของหน้าต่างใหม่จะเล็กกว่าของหน้าต่างพ่อ แขนงเดิลของหน้าต่างลูกนี้จะเก็บอยู่ใน instance->pdata->cwevar.hWndEdit (ดูรูปที่ 4- 4)

1.4.3. กรณีค่าของ instance->pdata->hwnd ไม่เป็น NULL แต่ค่าของ window->window เป็น NULL ในกรณีนี้ฟังก์ชันจะทำลายหน้าต่างลูกทั้งหมดที่สร้างขึ้นมาพร้อมกับยกเลิกการจับคลาสที่ทำไว้

1.4.4. กรณีค่าของ instance->pdata->hwnd ไม่เป็น NULL และค่าของ window->window ไม่เท่ากับค่าของ instance->pdata->hwnd ในกรณีนี้แสดงว่ามีการให้หน้าต่างใหม่มา ฟังก์ชันนี้จะทำการยกเลิกจับคลาสเดิมแล้วทำการจับคลาสใหม่พร้อมกับเก็บแสดิลของหน้าต่างใหม่นี้ไว้ใน instance->pdata->hwnd และทำการตั้งหน้าต่างใหม่นี้ให้เป็นหน้าต่างพ่อของหน้าต่างที่ใช้แสดงกรอบ



รูปที่ 4- 4 การจัดการหน้าต่างของซียูไรเตอร์ปลั๊กอิน

1.5. ฟังก์ชัน NPP\_NewStream ฟังก์ชันนี้จะถูกเรียกเมื่อเน็ตสเคปสร้างสตรีมขึ้นมาเพื่อใช้ในการส่งเอกสารมาให้กับซียูไรเตอร์ปลั๊กอิน ภายในฟังก์ชันนี้จะทำหน้าที่เพียงแค่ออกเน็ตสเคปว่าให้ส่งเอกสารผ่านทางฟังก์ชัน NPP\_WriteReady และ NPP\_Write

1.6. ฟังก์ชัน NPP\_WriteReady ฟังก์ชันนี้จะถูกเรียกก่อนที่จะมีการส่งเอกสารมาให้เพื่อถามถึงปริมาณข้อมูลที่ซียูไรเตอร์ปลั๊กอินสามารถรับได้ในขณะนั้น ภายในฟังก์ชันนี้จะทำหน้าที่เพียงแค่ออกเน็ตสเคปว่าสามารถรับข้อมูลได้มากที่สุด 32 กิโลไบต์

1.7. ฟังก์ชัน NPP\_Write ฟังก์ชันนี้จะถูกเรียกเพื่อใช้ ส่งข้อมูลมาให้กับซียูไรเตอร์ปลั๊กอิน ภายในฟังก์ชันนี้จะทำหน้าที่นำข้อมูลที่เน็ตสเคปส่งมาไปเก็บในแฟ้มข้อมูลที่ระบุโดย instance->pdata->cwevar.filename การเก็บข้อมูลนี้เป็นแบบต่อกั้ยข้อมูลเก่าเนื่องจากเอกสารอาจจะถูกแบ่งเป็นส่วนๆแล้วทยอยส่งมาให้ทีละส่วน

1.8. ฟังก์ชัน NPP\_DestroyStream เมื่อเน็ตสเคปมีการเรียกฟังก์ชันนี้หมายถึงเราได้เอกสารอยู่ในแฟ้มข้อมูลที่ระบุโดย instance->pdata->cwevar.filename แล้ว ฟังก์ชันนี้จะทำหน้าที่อ่านตัวเลือกท้ายเอกสารลงสู่โครงสร้างข้อมูลใน instance->pdata->cwevar โดยจะข้ามเขตข้อมูลที่รับค่าจากรหัสคำสั่ง <EMBED> ไป (ฟังก์ชันในแฟ้มข้อมูล FILE.C) และอ่านตัวเอกสารลงสู่หน่วยความจำพร้อมกับสร้างรายการเชื่อมโยงแบบรายการโยงคู่หมุนวน (circular doubly linked list) ขึ้นมาเพื่อใช้ท่องไปในแต่ละบรรทัดของเอกสาร (ฟังก์ชันในแฟ้มข้อมูล FILE.C) จากนั้นจะทำการสร้างแบบอักษรคงที่ที่ใช้สำหรับแสดงผลขึ้นมา (ฟังก์ชันในแฟ้มข้อมูล GENFONT.C) กำหนดขนาดของเซลแสดงผล (ฟังก์ชันในแฟ้มข้อมูล GNRLFN.C) ปรับสัดส่วนของรูปภาพในเอกสาร (ฟังก์ชันในแฟ้มข้อมูล FILE.C) ตั้งค่าของ instance->pdata->bDataReady เป็นจริงเพื่อแสดงว่าได้เอกสารมาแล้วและพร้อมที่จะนำขึ้นแสดงผล ส่งข้อความ WM\_SIZE ไปยังฟังก์ชัน DispWndProc เพื่อให้ทำการเตรียมการแสดงผล

1.9. ฟังก์ชัน NPP\_StreamAsFile ภายในฟังก์ชันนี้ไม่มีการทำงานใดๆ

1.10. ฟังก์ชัน NPP\_Destroy ฟังก์ชันนี้จะถูกเรียกเมื่อเน็ตสเคปจะทำลายอินสแตนซ์ ภายในฟังก์ชันนี้จะทำหน้าที่นำโครงสร้างข้อมูล PluginInstance ออกจากรายการเชื่อมโยง (ฟังก์ชัน UnAssociateInstance ในแฟ้มข้อมูล INSTLIST.C) ยกเลิกขบคลาสที่ทำได้ในฟังก์ชัน NPP\_SetWindow ลบรายการเชื่อมโยงแบบรายการโยงคู่หมุนวนและหน่วยความจำที่ใช้เก็บเอกสาร (ฟังก์ชันในแฟ้มข้อมูล FILE.C) คืนทรัพยากรต่างๆสู่ระบบ ทำลายหน้าต่างทั้งหมดที่สร้างขึ้นมา ลบแฟ้มข้อมูลชั่วคราวที่เก็บเอกสารไว้ และทำลายโครงสร้างข้อมูล PluginInstance

1.11. ฟังก์ชัน NPP\_Print ฟังก์ชันนี้จะถูกเรียกเมื่อเน็ตสเคปต้องการให้ซียูไรเตอร์ปลั๊กอินพิมพ์เอกสารออกทางเครื่องพิมพ์ การพิมพ์นี้จะมี 2 แบบคือแบบเต็มหน้าและแบบฝั่ง แต่ภายในฟังก์ชันนี้จะทำหน้าที่พิมพ์เอกสารในแบบฝั่งเท่านั้น (ฟังก์ชันในแฟ้มข้อมูล CWWPRINT.C) ส่วนการพิมพ์แบบเต็มหน้าจะต้องเรียกผ่านเมนูของซียูไรเตอร์ปลั๊กอิน ข้อมูลที่พิมพ์ออกไปคือส่วนของเอกสารที่ปรากฏบนหน้าต่างที่ใช้แสดงเอกสารในขณะกำลังพิมพ์ การพิมพ์นี้จะไม่มีการใช้พารามิเตอร์ในการพิมพ์ เช่นหัวกระดาษ จำนวนบรรทัดต่อหน้า และจะถือว่าคำสั่งจุดเป็นเพียงข้อมูลธรรมดาเท่านั้น

2. CLASS.C แฟ้มข้อมูลนี้จะเก็บฟังก์ชันที่ทำหน้าที่จัดการเกี่ยวกับวินโดวส์คลาส ฟังก์ชันที่เป็นวินโดวส์โปรซีเยอร์ การทำงานของแต่ละฟังก์ชันในแฟ้มข้อมูลนี้มีดังนี้

2.1. ฟังก์ชัน RegisterAllClass ภายในฟังก์ชันนี้จะทำหน้าที่สร้างและลงทะเบียนวินโดวส์คลาสที่มีการใช้งานในซียูไรเตอร์ปลั๊กอิน วินโดวส์คลาสที่สร้างจะมีอยู่ด้วยกัน 2 คลาสคือ FrameCls ซึ่งเป็นคลาสที่ใช้สำหรับหน้าต่างที่ใช้แสดงกรอบและ DispCls ซึ่งเป็นคลาสที่ใช้สำหรับหน้าต่างที่ใช้แสดงเอกสาร

2.2. ฟังก์ชัน UnRegisterAllClass ภายในฟังก์ชันนี้จะทำหน้าที่ยกเลิกการลงทะเบียนและทำลายวินโดวส์คลาสทั้ง 2 ที่ลงทะเบียนไว้ในฟังก์ชัน RegisterAllClass

2.3. ฟังก์ชัน PluginSubClass ฟังก์ชันนี้จะถูกใช้เป็นวินโดวส์โพรซีเยอร์แทนของเดิมสำหรับหน้าต่างที่เน็ตสเคปสร้างขึ้นและส่งแชนเดิลมาให้ทางฟังก์ชัน NPP\_SetWindow ข้อความที่ฟังก์ชันนี้จะจัดการแสดงอยู่ในตารางที่ 4- 4

ข้อความ	การทำงาน
WM_SIZE	ข้อความนี้จะถูกส่งมาให้เมื่อมีการเปลี่ยนแปลงขนาดของหน้าต่าง ฟังก์ชันนี้จะทำการเปลี่ยนแปลงขนาดของหน้าต่างที่ใช้แสดงกรอบ (ซึ่งเป็นหน้าต่างลูกของหน้าต่างนี้) ตามขนาดใหม่ของหน้าต่างนี้
อื่นๆ	ฟังก์ชันนี้จะส่งต่อไปให้กับวินโดวส์โพรซีเยอร์เดิมจัดการ (ค่าของวินโดวส์โพรซีเยอร์เดิมอยู่ในเขตข้อมูล lpfnOldWndProc ของโครงสร้างข้อมูล PluginInstance)

ตารางที่ 4- 4 ข้อความที่ฟังก์ชัน PluginSubClass จัดการ

แชนเดิลของหน้าต่างที่ใช้แสดงกรอบที่ต้องใช้ในการเปลี่ยนแปลงขนาดและค่าของ lpfnOldWndProc จะอยู่ในโครงสร้างข้อมูล PluginInstance ซึ่งอาจจะมากกว่าหนึ่งโครงสร้าง การจะหาว่าต้องใช้โครงสร้างไหนจะทำโดยการส่งแชนเดิลของหน้าต่างนี้ไปให้ฟังก์ชัน GetInstance (ในแฟ้มข้อมูล INSTLIST.C)

2.4. ฟังก์ชัน FrameWndProc ฟังก์ชันนี้เป็นวินโดวส์โพรซีเยอร์ของหน้าต่างที่มีคลาสเป็น FrameCls หน้าที่หลักของฟังก์ชันนี้คือวาดกรอบรอบหน้าต่าง ภายในฟังก์ชันนี้จะทำหน้าที่จัดการกับข้อความที่ไม่ใช่ซอฟต์แวร์วินโดวส์ส่งมาให้ตามตารางที่ 4- 5



ข้อความ	การทำงาน
WM_SIZE	ข้อความนี้จะถูกส่งมาให้เมื่อมีการเปลี่ยนแปลงขนาดของหน้าต่าง ฟังก์ชันนี้จะทำการเปลี่ยนแปลงขนาดของหน้าต่างที่ใช้แสดงเอกสาร (ซึ่งเป็นหน้าต่างลูกของหน้าต่างนี้) ตามขนาดใหม่ของหน้าต่างนี้
WM_ERASEBKGD	ข้อความนี้จะถูกส่งมาให้เมื่อไมโครซอฟท์วินโดวส์ต้องการให้ลบฉากหลังในหน้าต่าง ฟังก์ชันนี้จะทำการระบายสีฉากหลังใหม่โดยใช้สีฉากหลังของไมโครซอฟท์วินโดวส์
WM_PAINT	ข้อความนี้จะถูกส่งมาให้เมื่อไมโครซอฟท์วินโดวส์ต้องการให้แสดงข้อมูลในหน้าต่างใหม่ ฟังก์ชันนี้จะทำการวาดกรอบรอบหน้าต่างใหม่ โดยใช้สีค่าค่าของเขตข้อมูล <code>cwevar.frame</code> ในโครงสร้างข้อมูล <code>PluginInstance</code> เป็นจริง
WM_MOUSEACTIVATE	ข้อความนี้จะถูกส่งมาให้เมื่อมีการกดเมาส์บนหน้าต่าง ฟังก์ชันนี้จะทำการตั้งโฟกัส (focus) ไปยังหน้าต่างที่ใช้แสดงเอกสารเพื่อให้สามารถรับเป็นพิมพ์ได้
อื่นๆ	ฟังก์ชันนี้จะส่งต่อไปให้กับไมโครซอฟท์วินโดวส์จัดการ

ตารางที่ 4- 5 ข้อความที่ฟังก์ชัน `FrameWndProc` จัดการ

แฮนเดิลของหน้าต่างที่ใช้แสดงเอกสารที่ต้องใช้ในการเปลี่ยนแปลงขนาดและค่าของ `cwevar.frame` จะอยู่ในโครงสร้างข้อมูล `PluginInstance` ซึ่งอาจจะมีมากกว่าหนึ่งโครงสร้าง การจะหาว่าต้องใช้โครงสร้างไหนจะทำโดยการส่งแฮนเดิลของหน้าต่างพ่อของหน้าต่างนี้ (ซึ่งก็คือหน้าต่างที่เน็ตสเคปสร้างขึ้นมา) ไปให้ฟังก์ชัน `GetInstance` (ในแฟ้มข้อมูล `INSTLIST.C`)

2.5. ฟังก์ชัน `DispWndProc` ฟังก์ชันนี้เป็นวินโดวส์โพรซีเยอร์ของหน้าต่างที่มีคลาสเป็น `DispCls` หน้าหลักของฟังก์ชันนี้คือแสดงผลเอกสารและได้ตอบกับการกระทำของผู้ใช้เช่นการกดเมาส์หรือเป็นพิมพ์ ภายในฟังก์ชันนี้จะทำหน้าที่จัดการกับข้อความที่ไมโครซอฟท์วินโดวส์ส่งมาให้ตามตารางที่ 4- 6

ข้อความ	การทำงาน
WM_CREATE	ข้อความนี้จะถูกส่งมาให้เมื่อเกิดการสร้างหน้าต่างขึ้นมาใหม่ ฟังก์ชันนี้จะทำการใส่ข้อมูลที่จำเป็นลงในดีไวซ์คอนเท็กซ์ (device context) (Petzold, 1992) ประจำหน้าต่างนี้ ดีไวซ์คอนเท็กซ์นี้จะใช้ในการแสดงผลเอกสาร
WM_SIZE	ข้อความนี้จะถูกส่งมาให้เมื่อมีการเปลี่ยนแปลงขนาดของหน้าต่าง ฟังก์ชันนี้จะทำการหาขนาดที่แท้จริงของหน้าต่างนี้ (นับส่วนที่เป็นแถบเลื่อนด้วย) จากนั้นจะทำการหาว่าต้องการแถบเลื่อนแนวนอนและแนวตั้งหรือไม่ถ้าต้องการก็จะทำการติดตั้งแถบเลื่อน จากนั้นจะทำการหาจำนวนบรรทัดและคอลัมน์ที่สามารถแสดงได้ในหน้าต่าง พร้อมกับทำการตัดคำถ้ามีการเลือกไว้ (ฟังก์ชันในเพิ่มข้อมูล WRAP.C)
WM_ERASEBKGD	ข้อความนี้จะถูกส่งมาให้เมื่อไมโครซอฟท์วินโดวส์ต้องการให้ลบฉากหลังในหน้าต่าง ฟังก์ชันนี้จะทำการระบายสีฉากหลังใหม่โดยใช้สีฉากหลังของไมโครซอฟท์วินโดวส์
WM_PAINT	ข้อความนี้จะถูกส่งมาให้เมื่อไมโครซอฟท์วินโดวส์ต้องการให้แสดงข้อมูลในหน้าต่างใหม่ ฟังก์ชันนี้จะทำการแสดงส่วนของเอกสารในหน้าต่างใหม่ (ฟังก์ชันในเพิ่มข้อมูล EDIT.C) ถ้าค่าของเขตข้อมูล bDataReady ในโครงสร้างข้อมูล PluginInstance เป็นจริง (ได้เอกสารแล้ว) สีของตัวหนังสือจะใช้สีตามตัวหนังสือในไมโครซอฟท์วินโดวส์
WM_MOUSEACTIVATE	ข้อความนี้จะถูกส่งมาให้เมื่อมีการกดเมาส์บนหน้าต่าง ฟังก์ชันนี้จะทำการตั้งโฟกัสบนหน้าต่างเพื่อให้สามารถรับแป้นพิมพ์ได้
WM_HSCROLL	ข้อความนี้จะถูกส่งมาให้เมื่อมีการกดเมาส์บนแถบเลื่อนแนวนอน ฟังก์ชันนี้จะทำการปรับปรุงการแสดงผลเอกสารเพื่อให้แสดงส่วนที่ถูกต้องกับตำแหน่งของแถบเลื่อนแนวนอน (ฟังก์ชันในเพิ่มข้อมูล MOVEMENT.C)
WM_VSCROLL	ข้อความนี้จะถูกส่งมาให้เมื่อมีการกดเมาส์บนแถบเลื่อนแนวตั้ง ฟังก์ชันนี้จะทำการปรับปรุงการแสดงผลเอกสารเพื่อให้แสดงส่วนที่ถูกต้องกับตำแหน่งของแถบเลื่อนแนวตั้ง (ฟังก์ชันในเพิ่มข้อมูล MOVEMENT.C)

ตารางที่ 4- 6 ข้อความที่ฟังก์ชัน DispWndProc จัดการ

ข้อความ	การทำงาน
WM_KEYDOWN	ข้อความนี้จะถูกส่งมาให้เมื่อมีการกดแป้นพิมพ์ ถ้าแป้นที่กดเป็นแป้นลูกศร PgUp PgDn Home End Tab BackSpace ฟังก์ชันนี้จะจำลองการทำงานเสมือนกับการกดเมาส์บนแถบเลื่อนแนวนอนหรือแนวตั้ง
WM_RBUTTONDOWN	ข้อความนี้จะถูกส่งมาให้เมื่อมีการกดปุ่มขวาของเมาส์บนหน้าต่างนี้ ฟังก์ชันนี้จะทำการแสดงเมนูขึ้นมาเพื่อให้ผู้ใช้เลือกรายการที่ต้องการ
WM_MEASUREITEM	เนื่องจากเมนูที่จะแสดงเมื่อได้รับข้อความ WM_RBUTTONDOWN เป็นเมนูที่ซียูไรเตอร์ปลั๊กอินระบุว่าต้องการแสดงผลเอง (owner draw) (Petzold, 1992) ดังนั้นไมโครซอฟท์วินโดวส์จะส่งข้อความนี้มาเพื่อถามถึงความกว้างและความสูงของแต่ละรายการในเมนู ฟังก์ชันนี้จะทำหน้าที่ตอบความกว้างและความสูงของแต่ละรายการในเมนู (ฟังก์ชัน MenuItemDimension ในแฟ้มข้อมูล MENU.C)
WM_DRAWITEM	ข้อความนี้จะถูกส่งมาให้เมื่อไมโครซอฟท์วินโดวส์ต้องการให้ซียูไรเตอร์ปลั๊กอินแสดงรายการแต่ละอันในเมนู (หลังจากทราบความกว้างและความสูงของแต่ละรายการแล้ว) ฟังก์ชันนี้จะทำการแสดงแต่ละรายการของเมนูตามสถานะและภาษาของแต่ละรายการ เช่นถูกเลือกหรือไม่ถูกเลือก (ฟังก์ชัน DrawSelectMenu ในแฟ้มข้อมูล MENU.C)
WM_COMMAND	ข้อความนี้จะถูกส่งมาให้เมื่อผู้ใช้เลือกรายการในเมนู (รายการทั้งหมดดูได้จากหัวข้อ 7) ฟังก์ชันนี้จะทำงานตามรายการที่เลือก
อื่นๆ	ฟังก์ชันนี้จะส่งต่อไปให้กับไมโครซอฟท์วินโดวส์จัดการ

ตารางที่ 4- 6 ข้อความที่ฟังก์ชัน DispWndProc จัดการ (ต่อ)

ในส่วน of ข้อความ WM\_COMMAND ฟังก์ชันนี้จะทำงานตามการเลือกรายการดังนี้

2.5.1. รายการแบบตัวอักษรหรือ Font: ฟังก์ชันนี้จะเรียกใช้ฟังก์ชันในแฟ้มข้อมูล FONT.C เพื่อทำการเปลี่ยนแบบอักษร

2.5.2. รายการค้นหาหรือ Search: ฟังก์ชันนี้จะเรียกใช้ฟังก์ชันในแฟ้มข้อมูล SEARCH.C เพื่อทำการค้นหาข้อมูลในเอกสาร

2.5.3. รายการพิมพ์หรือ Print: ฟังก์ชันนี้จะเรียกใช้ฟังก์ชันในแฟ้มข้อมูล CWWPRINT.C เพื่อทำการพิมพ์เอกสารในแบบเต็มหน้า

2.5.4. รายการ English Message หรือข้อความภาษาไทย: ฟังก์ชันนี้จะทำการตั้งค่าในเซต

ข้อมูล cwevar.thailanguage ของโครงสร้างข้อมูล PluginInstance ตามภาษาที่เลือก

2.5.5. รายการใช้รหัสภาษาหรือ Use KU code: ฟังก์ชันนี้จะเรียกใช้ฟังก์ชันในแฟ้มข้อมูล FILE.C เพื่อทำการเปลี่ยนรหัสภาษาไทย

2.5.6. รายการตัดคำหรือ Word Wrap: ฟังก์ชันนี้จะเรียกใช้ฟังก์ชันในแฟ้มข้อมูล WRAP.C เพื่อทำการตัดคำ

2.5.7. รายการเกี่ยวกับหรือ About: ฟังก์ชันนี้จะเรียกใช้ฟังก์ชัน About ในแฟ้มข้อมูล ABOUT.C เพื่อทำการแสดงกล่องคำตอบ

ค่าของ bDataReady และ cwevar.thailanguage จะอยู่ในโครงสร้างข้อมูล PluginInstance ซึ่งอาจจะมีมากกว่าหนึ่งโครงสร้าง การจะหาว่าต้องใช้โครงสร้างไหนจะทำโดยการส่งแอสเซนดิคของหน้าต่างพ้อของหน้าต่างพ้อของหน้าต่างนี้ (ซึ่งก็คือหน้าต่างที่เน็ตสเคปสร้างขึ้นมา) ไปให้ฟังก์ชัน GetInstance (ในแฟ้มข้อมูล INSTLIST.C)

3. ABOUT.C แฟ้มข้อมูลนี้จะเก็บฟังก์ชัน About ที่ทำหน้าที่แสดงกล่องคำตอบที่ใช้แสดงข้อมูลเกี่ยวกับตัวโปรแกรม เช่นชื่อ รุ่น ผู้พัฒนา ผู้ที่เกี่ยวข้อง ตัวเชื่อมโยงไปยังโฮมเพจ (home page) ของโครงการซียูไรเตอร์ พร้อมกับเล่นเพลงประกอบ

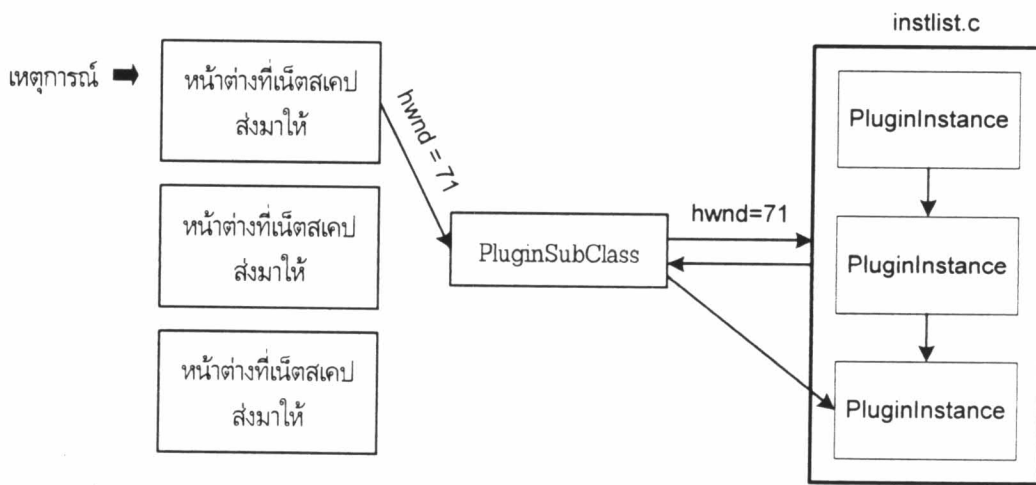
4. GLOBAL.C แฟ้มข้อมูลนี้จะเก็บฟังก์ชันที่ทำหน้าที่ในการตั้งค่าให้กับตัวแปรร่วม จองและคิน ทรัพยากรที่จำเป็นต้องใช้ตลอดการทำงานของซียูไรเตอร์อีกอิน การทำงานของแต่ละฟังก์ชันในแฟ้มข้อมูลนี้มีดังนี้

4.1. ฟังก์ชัน InitGlobal ภายในฟังก์ชันนี้จะทำหน้าที่ตั้งค่าให้กับตัวแปรร่วม จองทรัพยากรที่จำเป็นต้องใช้

4.2. ฟังก์ชัน DeInitGlobal ภายในฟังก์ชันนี้จะทำหน้าที่คืนทรัพยากรที่จองไว้โดยฟังก์ชัน InitGlobal

5. INSTLIST.C แฟ้มข้อมูลนี้จะเก็บฟังก์ชันที่ทำหน้าที่ตั้งค่าเริ่มต้นให้กับโครงสร้างข้อมูล PluginInstance นำโครงสร้างข้อมูลนี้เข้าสู่และออกจากรายการเชื่อมโยง รายการเชื่อมโยงนี้จะใช้เชื่อมโยงโครงสร้างข้อมูล PluginInstance ทั้งหมดที่มีเข้าด้วยกันเพื่อให้วินโดสโพรซีเยอร์สามารถเข้าถึงโครงสร้างข้อมูลที่ใช้ทำงานกับหน้าต่างได้โดยการค้นหาตามค่าแอสเซนดิคของหน้าต่าง รูปที่ 4- 5 แสดงตัวอย่างการค้นหาที่กล่าวถึงโดยสมมุติว่ามีเหตุการณ์เกิดขึ้นที่หน้าต่างที่เน็ตสเคปส่งมาให้ซึ่งมีค่าแอสเซนดิคเท่ากับ 71 ไมโครซอฟท์วินโดสจะทำการส่งแอสเซนดิคนี้พร้อม

กับข้อความไปที่วินโดวส์โพธิ์เซอร์ของหน้าต่างนี้ซึ่งก็คือฟังก์ชัน PluginSubClass (ผลจากการจับคลาส) เพื่อให้จัดการกับเหตุการณ์นั้น ถ้าการจัดการนี้จำเป็นต้องใช้ข้อมูลในโครงสร้างข้อมูล PluginInstance ฟังก์ชัน PluginSubClass จะทำการเรียกใช้ฟังก์ชันในแฟ้มข้อมูล INSTLIST.C เพื่อหาว่าโครงสร้างข้อมูลที่ต้องการอยู่ที่ไหนโดยจะส่งแฮนเดิลของหน้าต่างไปให้ ฟังก์ชันในแฟ้มข้อมูล INSTLIST.C ก็จะทำการค้นหาโครงสร้างข้อมูลจากรายการเชื่อมโยงโดยใช้การเปรียบเทียบค่าแฮนเดิลที่ได้มากับเขตข้อมูล hwnd ของทุกๆโครงสร้างข้อมูล เมื่อค้นพบก็จะส่งตำแหน่งของโครงสร้างกลับไปให้ซึ่งจะทำให้ฟังก์ชัน PluginSubClass สามารถเข้าถึงโครงสร้างข้อมูลนี้ได้โดยตรง



รูปที่ 4- 5 ตัวอย่างการค้นหาโครงสร้างข้อมูล PluginInstance โดยใช้แฮนเดิลของหน้าต่าง

การทำงานของแต่ละฟังก์ชันในแฟ้มข้อมูลนี้มีดังนี้

- 5.1. ฟังก์ชัน AssociateInstance ภายในฟังก์ชันนี้จะทำหน้าที่นำโครงสร้างข้อมูล PluginInstance ที่สร้างขึ้นมาในฟังก์ชัน NPP\_New เข้าไปต่อท้ายรายการเชื่อมโยง
- 5.2. ฟังก์ชัน UnassociateInstance ภายในฟังก์ชันนี้จะทำหน้าที่นำโครงสร้างข้อมูล PluginInstance ที่ระบุออกจากรายการเชื่อมโยง
- 5.3. ฟังก์ชัน GetInstance ภายในฟังก์ชันนี้จะทำหน้าที่ค้นหาโครงสร้างข้อมูล PluginInstance ที่อยู่ในรายการเชื่อมโยงที่มีค่าของเขตข้อมูล hwnd ตรงกับค่าแฮนเดิลของหน้าต่างที่ให้มา (รูปที่ 4- 5)
- 5.4. ฟังก์ชัน InitPluginInstance ภายในฟังก์ชันนี้จะทำหน้าที่ตั้งค่าเริ่มต้นให้กับโครงสร้างข้อมูล PluginInstance ค่าเริ่มต้นที่ตั้งให้จะมาจากโครงสร้างข้อมูล Option และจากรหัสคำสั่ง <EMBED>

6. MAIN.C เพิ่มข้อมูลนี้จะเก็บฟังก์ชัน LibMain ซึ่งเป็นฟังก์ชันที่ไมโครซอฟท์วินโดวส์เรียกใช้เมื่อมีการนำคลังโปรแกรมเชื่อมโยงแบบพลวัตลงสู่หน่วยความจำ (ซียูไรเตอร์ปลั๊กอินก็เป็นคลังโปรแกรมเชื่อมโยงแบบพลวัต) หน้าที่ของฟังก์ชันนี้มีเพียงแค่เก็บค่าอินสแตนซ์ประจำตัวของคลังโปรแกรมเชื่อมโยงแบบพลวัตที่ไมโครซอฟท์วินโดวส์ส่งมาให้เพื่อไว้ใช้งานในภายหลัง

7. MENU.C เพิ่มข้อมูลนี้จะเก็บฟังก์ชันที่ทำหน้าที่จัดการเกี่ยวกับเมนู เช่นสร้างทำลายเมนู รายการทั้งหมดในเมนูแสดงอยู่ในตารางที่ 4- 7

รายการ (เป็นภาษาไทย)	รายการ (เป็นภาษาอังกฤษ)	การทำงาน
แบบตัวอักษร	Font	ใช้เปลี่ยนแบบอักษรที่ใช้ในการแสดงผลและพิมพ์
ค้นหา	Find	ใช้ค้นหาข้อมูลในเอกสาร
พิมพ์	Print	ใช้พิมพ์เอกสารแบบเต็มหน้า
English Message	ข้อความภาษาไทย	ใช้เปลี่ยนข้อความในเมนูและกล่องคำโต้ตอบเป็นไทยหรืออังกฤษ
ใช้รหัสภาษา	Use KU code	ใช้เลือกรหัสภาษาไทย (สมอ. หรือเกษตร)
ตัดคำ	Word Wrap	ใช้เลือกว่าจะตัดคำหรือไม่
เกี่ยวกับ	About	ใช้แสดงกล่องคำโต้ตอบที่แสดงข้อมูลเกี่ยวกับตัวโปรแกรม ตัวเชื่อมไปยังโฮมเพจของโครงการซียูไรเตอร์ พร้อมกับเล่นเพลงประกอบ

ตารางที่ 4- 7 รายการทั้งหมดในเมนูของซียูไรเตอร์ปลั๊กอิน

การทำงานของแต่ละฟังก์ชันในเพิ่มข้อมูลนี้มีดังนี้

7.1. ฟังก์ชัน InitMenu ภายในฟังก์ชันนี้จะสร้างเมนูขึ้นมา เมนูที่สร้างขึ้นมานี้จะเป็นแบบที่ซียูไรเตอร์ปลั๊กอินเป็นผู้แสดงเอง

7.2. ฟังก์ชัน DeInitMenu ภายในฟังก์ชันนี้จะทำลายเมนูที่สร้างขึ้นมา

7.3. ฟังก์ชัน GetMainPopup ภายในฟังก์ชันนี้จะให้ค่าแฮนเดิลของเมนูที่สร้างขึ้นมา

7.4. ฟังก์ชัน MenuItemDimension ภายในฟังก์ชันนี้จะให้ค่าความกว้างและความสูงของรายการในเมนู (ดูรายละเอียดในฟังก์ชัน DispWndProc)

7.5. ฟังก์ชัน DrawSelectMenu ภายในฟังก์ชันนี้จะทำหน้าที่วาดรายการของเมนูบนจอภาพ (ดูรายละเอียดในฟังก์ชัน DispWndProc)

8. MSGBOX.C เพิ่มข้อมูลนี้จะเก็บฟังก์ชัน MsgBox ที่ทำหน้าที่แสดงกล่องข้อความ เพื่อใช้แสดงข้อความแก่ผู้ใช้

9. OPTION.C เพิ่มข้อมูลนี้จะเก็บฟังก์ชันที่ทำหน้าที่อ่านตัวเลือกจากเพิ่มข้อมูล NPCWW.INI ลงสู่โครงสร้างข้อมูล Option ซึ่งมีรายละเอียดดังนี้

```
typedef struct Option_tag
{
    BOOL  stdcode;
    BOOL  wordwrap;
    int   thailanguage;
    int   leftmar;
    int   rightmar;
    int   lineperpage;
    int   cpi;
    int   spacing;
    char  fontname[LF_FACESIZE];
} Option;
```

และนอกจากเก็บฟังก์ชันที่ทำหน้าที่อ่านตัวเลือกแล้วยังเก็บฟังก์ชันที่ทำหน้าที่ตรวจสอบค่าของตัวเลือก การทำงานของแต่ละฟังก์ชันในเพิ่มข้อมูลนี้มีดังนี้

9.1. ฟังก์ชัน ValidateThaiLanguage ภายในฟังก์ชันนี้จะทำหน้าที่ตรวจสอบค่าของตัวเลือก thailanguage ว่ามีค่าถูกต้องหรือไม่ ถ้าไม่ จะทำการตั้งค่าให้ใหม่

9.2. ฟังก์ชัน ValidateMargin ภายในฟังก์ชันนี้จะทำหน้าที่ตรวจสอบค่าของตัวเลือก leftmar และ rightmar ว่ามีค่าถูกต้องหรือไม่ ถ้าไม่ทำการตั้งค่าให้ใหม่

9.3. ฟังก์ชัน ValidateLinePerPage ภายในฟังก์ชันนี้จะทำหน้าที่ตรวจสอบค่าของตัวเลือก lineperpage ว่ามีค่าถูกต้องหรือไม่ ถ้าไม่ทำการตั้งค่าให้ใหม่

9.4. ฟังก์ชัน ValidateCpi ภายในฟังก์ชันนี้จะทำหน้าที่ตรวจสอบค่าของตัวเลือก cpi ว่ามีค่าถูกต้องหรือไม่ ถ้าไม่ทำการตั้งค่าให้ใหม่

9.5. ฟังก์ชัน ValidateSpacing ภายในฟังก์ชันนี้จะทำหน้าที่ตรวจสอบค่าของตัวเลือก spacing ว่ามีค่าถูกต้องหรือไม่ ถ้าไม่ทำการตั้งค่าให้ใหม่

9.6. ฟังก์ชัน ValidateFont ภายในฟังก์ชันนี้จะทำหน้าที่ตรวจสอบค่าของตัวเลือก fontname ว่าเป็นแบบอักษรที่มีอยู่ในแฟ้มข้อมูล WIN.INI หรือไม่ ถ้าไม่ทำการตั้งค่าให้ใหม่

9.7. ฟังก์ชัน ReadOptionINI ภายในฟังก์ชันนี้จะทำหน้าที่อ่านตัวเลือกจากแฟ้มข้อมูล NPCWW.INI ลงสู่โครงสร้างข้อมูล Option พร้อมทั้งทำการตรวจสอบค่าตัวเลือกทั้งหมดด้วย

การระบุชนิดข้อมูลของซียูไรเตอร์ปลั๊กอิน

การระบุชนิดข้อมูลที่ซียูไรเตอร์ปลั๊กอินจัดการได้จะทำโดยการเปลี่ยนชื่อแฟ้มข้อมูล NPDLL16.RC2 ซึ่งมาพร้อมกับชุดพัฒนาเนตสเคปปลั๊กอินมาเป็น NPCWW16.RC2 จากนั้นทำการแก้ไขจนได้ผลดังนี้

VS_VERSION_INFO	VERSIONINFO
FILEVERSION	0,0,1,0
PRODUCTVERSION	1,0,0,0
FILEFLAGSMASK	VS_FFI_FILEFLAGSMASK
FILEFLAGS	(VS_FF_PRERELEASE   VS_FF_DEBUG)
FILEOS	VOS_DOS_WINDOWS16
FILETYPE	VFT_DLL



```
FILESUBTYPE          VFT2_UNKNOWN
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904E4"
        BEGIN
            VALUE "CompanyName", "Chulalongkorn University\0"
            VALUE "FileDescription", "CWW77 Plug-in for Netscape\0"
            VALUE "FileVersion", "0.0.1.0\0"
            VALUE "InternalName", "CWW77 Plug-in for Netscape\0"
            VALUE "LegalCopyright", "Chulalongkorn University\0"
            VALUE "LegalTrademarks", "Chulalongkorn University\0"
            VALUE "OriginalFilename", "NPCWW16.DLL\0"
            VALUE "ProductName", "CWW77 Plug-in for Netscape\0"
            VALUE "ProductVersion", "1.0.0.0\0"
            VALUE "FileExtents", "cw,cww\0"
            VALUE "MIMEType", "application/x-cww\0"
            VALUE "FileOpenName", "CWW Files (*.cw;*.cww)\0"
        END
    END
END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1252
    END
END
```