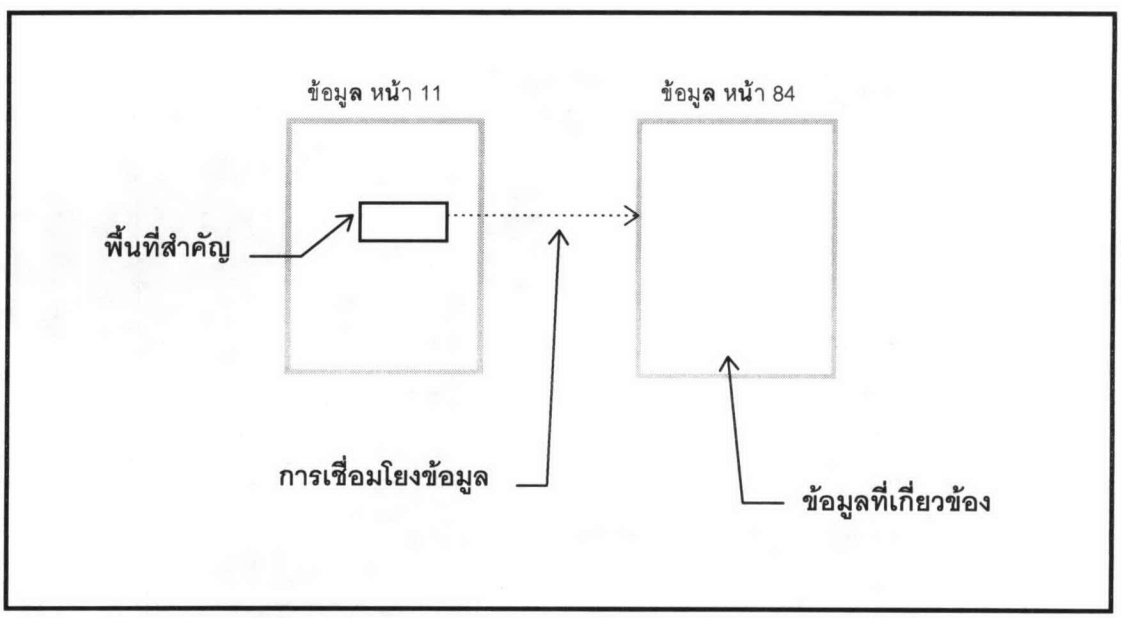
# บทที่ 2

# แนวคิดและทฤษฎี

## 2.1 ไฮเปอร์ดอกคิวเมนต์ [1]

ไฮเปอร์ดอกคิวเมนต์ คือ ข้อมูลที่เก็บอยู่ในระบบคอมพิวเตอร์ และ ผู้ใช้สามารถเรียกใช้งานโดยการคลิกเมาส์บนพื้นที่สำคัญ (hyper area) จากนั้น ข้อมูลที่เกี่ยวข้องหรือเชื่อมโยงอยู่กับพื้นที่สำคัญจะแสดงให้เห็นบนจอภาพ ข้อมูลเหล่านี้อาจเป็นตัวอักษร แผนภาพ รูปภาพ ภาพเคลื่อนไหว เสียงพูด เสียงอื่น ๆ หรือโปรแกรมคอมพิวเตอร์




รูปที่ 2-1 แสดงองค์ประกอบของไฮเปอร์ดอกคิวเมนต์

### 2.1.1 พื้นที่สำคัญ

คือ บริเวณที่เป็นข้อความ ภาพ หรือ พื้นที่ใด ๆ บนข้อมูลที่ผู้ใช้สามารถคลิกเมาส์บนพื้นที่นั้นแล้วข้อมูลที่เกี่ยวข้องจะปรากฏขึ้นบนจอภาพ ตัวอย่างความสามารถในการทำงานเมื่อคลิกเมาส์บนพื้นที่สำคัญ เช่น

- กระโดด (jump) ไปแสดงข้อมูลที่เกี่ยวข้องบนจอภาพ หรือปิดข้อมูลนั้นให้กลายเป็นไอคอนหรือหายไปจากจอภาพ
- แสดงคำจำกัดความ หรือ คำขยายความ บนจอซ้อนแบบผุดขึ้น (popup window)
- แสดงรูปภาพนิ่ง ภาพเคลื่อนไหว หรือเสียงประกอบ

- ทำงานตามคำสั่งของโปรแกรมที่กำหนดให้เป็นของพื้นที่สำคัญนั้น ๆ

2.1.2 ข้อมูลที่เกี่ยวข้อง

คือ ข้อมูลที่เกี่ยวข้องกับข้อมูลที่เป็นพื้นที่สำคัญ เมื่อผู้ใช้คลิกเมาส์บนพื้นที่สำคัญ ข้อมูลที่เกี่ยวข้องกับพื้นที่สำคัญนั้นจะปรากฏขึ้นบนจอภาพ

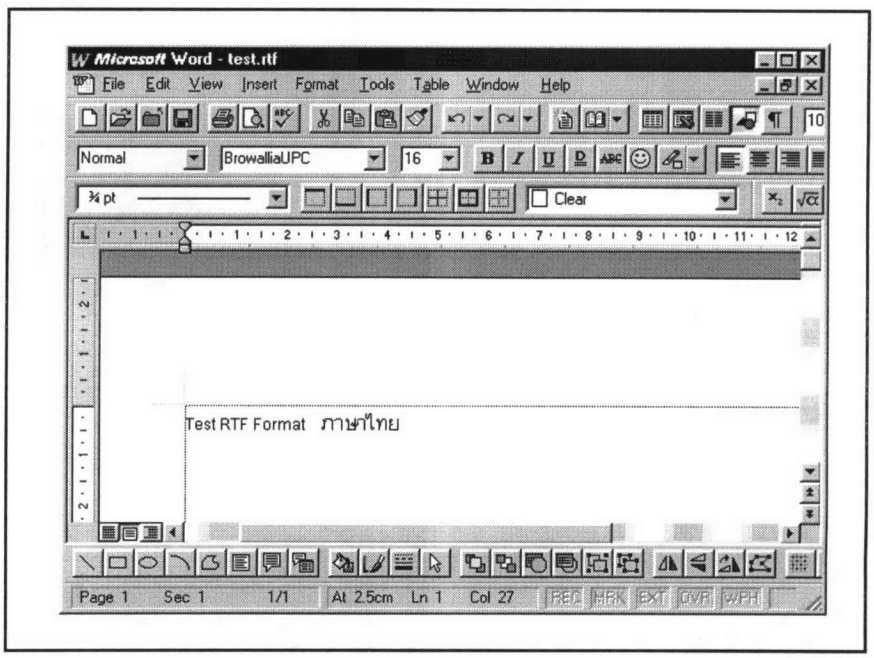
2.1.3 การเชื่อมโยงข้อมูล

การเชื่อมโยงข้อมูลจากพื้นที่สำคัญไปยังข้อมูลที่เกี่ยวข้อง จะเห็นว่าข้อมูลส่วนที่เป็นพื้นที่สำคัญ มีความสัมพันธ์หรือเชื่อมโยงอยู่กับข้อมูลที่เกี่ยวข้อง เพราะเมื่อคลิกเมาส์บนพื้นที่สำคัญ จะปรากฏข้อมูลที่เกี่ยวข้องบนจอภาพ

## 2.2 มาตรฐาน RTF (Rich Text Format) [2]

บริษัทไมโครซอฟต์ เป็นผู้กำหนดมาตรฐาน RTF เพื่อให้โปรแกรมประยุกต์ที่อยู่บนระบบปฏิบัติการต่างชนิดกันสามารถเรียกใช้งานแฟ้มข้อมูลของโปรแกรมการประมวลผลคำที่อยู่ในรูปแบบ RTF แฟ้มเดียวกันได้โดยไม่ต้องใช้โปรแกรมในการแปลงรูปแบบข้อมูล ระบบปฏิบัติการดังกล่าว ได้แก่ เอ็มเอสดอส วินโดวส์ โอเอสทู และ แอปเปิลแมคอินทอช

โปรแกรมที่บันทึกแฟ้มข้อมูลรูปแบบอื่นให้อยู่ในรูปแบบ RTF เรียกว่า โปรแกรมบันทึก (writer) โดยจะแยกข้อมูลอื่น เช่น การควบคุมของโปรแกรมการประมวลผลคำ ออกจากข้อมูลจริง (actual text) แล้วจึงบันทึกลงในแฟ้มข้อมูลแฟ้มใหม่ โดยที่แฟ้มข้อมูลนี้จะประกอบด้วยข้อมูลจริง และ กลุ่ม RTF (RTF group) ที่เกี่ยวข้องกับข้อมูลจริงนั้น สำหรับโปรแกรมที่แปลงแฟ้มข้อมูล RTF ไปเป็นแฟ้มข้อมูลในรูปแบบอื่น เรียกว่า โปรแกรมอ่าน (reader)




รูปที่ 2-2 ตัวอย่างข้อมูลที่แสดงในรูปของเอกสารไมโครซอฟต์เวิร์ด

```
{\rtf1\ansi \deff15\deflang1054{\fonttbl{\f15\fswiss\fcharset222\fprq2 CordiaUPC;}
{\f165\fswiss\fcharset222\fprq2 BrowalliaUPC;}}{\colortbl;\red0\green0\blue0;\red0
\green0\blue255;\red0\green255\blue255;\red0\green255\blue0;
\red255\green0\blue255;\red255\green0\blue0;\red255\green255\blue0;\red255\green255
\blue255;\red0\green0\blue128;\red0\green128\blue128;\red0\green128\blue0;\red128\green0
\blue128;\red128\green0\blue0;\red128\green128\blue0;\red128\green128\blue128;
\red192\green192\blue192;}{\stylesheet{\widctlpar \f15\fs28 \snext0 Normal;}{\*\cs10
 \additive Default Paragraph Font;}}{\info{\author Paitoon}{\operator Paitoon}
{\creatim\yr1998\mo3\dy4\hr10\min35}{\revtim\yr1998\mo3\dy4\hr10\min37}{\version1}{\edmins2}
{\nofpages1}{\nofwords3}{\nofchars22}{\*\company Smart  Office  Ltd.}{\vern57431}}
\paperw11906\paperh16838 \widowctrl\ftnbj\aenddoc\formshade \fet0\sectd \linex0
\headery709\footery709\colsx709\endnhere {\*\pnseclvl1\pnucrm\pnstart1\pnindent720\pnhang
{\pntxta .}}{\*\pnseclvl2\pnucltr\pnstart1\pnindent720\pnhang{\pntxta .}}{\*\pnseclvl3\pndec
\pnstart1\pnindent720\pnhang{\pntxta .}}{\*\pnseclvl4\pnlcltr\pnstart1\pnindent720\pnhang
{\pntxta )}}{\*\pnseclvl5\pndec\pnstart1\pnindent720\pnhang{\pntxtb (}
{\pntxta )}}{\*\pnseclvl6\pnlcltr\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta )}}{\*\pnseclvl7
\pnlcrm\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta )}}{\*\pnseclvl8\pnlcltr\pnstart1\pnindent720
\pnhang{\pntxtb (}{\pntxta )}}{\*\pnseclvl9
\pnlcrm\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta )}}\pard\plain \widctlpar \f15\fs28
Test RTF Format    {\f165\fs32 \'c0\'d2\'c9\'d2\'e4\'b7\'c2}
\par }
```

รูปที่ 2-3 ตัวอย่างข้อมูลในรูปของ RTF

2.2.1 <u>องค์ประกอบของแฟ้มข้อมูล RTF</u>

ข้อมูลในแฟ้มข้อมูล RTF ประกอบด้วย

- ข้อความที่ไม่ได้จัดรูปแบบ
- คำควบคุม
- สัญลักษณ์ควบคุม
- กลุ่มคำสั่ง

ข้อมูลแต่ละแบบจะมีข้อกำหนดหรือรูปแบบตามที่ RTF กำหนดเพื่อให้สามารถเรียกใช้ข้อมูลบนเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการต่างประเภทกันได้

การเก็บข้อมูลชุดอักขระของแฟ้มข้อมูลตามมาตรฐาน RTF จะจัดเก็บเป็นตัวอักษร ASCII แบบ 7 บิตเท่านั้น โดยที่โปรแกรมแปลงข้อมูลที่ทำงานกับไมโครซอฟต์เวิร์ดบนวินโดวส์ และแมคอินทอชใช้ตัวอักษร 8 บิต ดังนั้น กรณีที่เป็นชุดอักขระภาษาไทย ข้อความจะถูกจัดเก็บในรูปแบบ 7 บิตด้วยเช่นกัน

รูปแบบของข้อมูลแต่ละประเภท มีดังนี้

คำควบคุม เป็นคำสั่งในรูปแบบพิเศษที่ RTF ใช้ในการกำหนดรหัสควบคุมเครื่องพิมพ์ และเป็นสิ่งที่โปรแกรมประยุกต์ใช้ในการจัดการกับเอกสาร คำควบคุมจะขึ้นต้นด้วย เครื่องหมาย "\" และมีรูปแบบ ดังนี้

\LetterSequence<Delimiter>

โดยที่ LetterSequence ประกอบด้วยตัวภาษาอังกฤษพิมพ์เล็ก ตั้งแต่ "a" ถึง "z" RTF ให้ความแตกต่างระหว่างตัวพิมพ์ใหญ่และเล็ก โดยที่คำควบคุม RTF ทั้งหมดจะใช้ตัวพิมพ์เล็ก

Delimiter หรือ ตัวจบ เป็นตัวบอกว่าจบคำควบคุม ซึ่งอาจเป็นได้ ดังนี้

- ตัวอักษรว่าง (Space) ในกรณีนี้ ตัวอักษรว่างถือเป็นส่วนหนึ่งของคำควบคุม
- ตัวเลข หรือ "-" เป็นการบอกว่ามีพารามิเตอร์ที่เป็นตัวเลขต่อท้าย ตัวเลขเหล่านี้จะจบด้วยตัวอักษรว่าง หรือตัวอื่นที่ไม่ใช่ชุดอักขระ พารามิเตอร์เป็นได้ทั้งเลขบวก และติดลบ ตั้งแต่ -32767 ถึง 32767 อย่างไรก็ตาม ไมโครซอฟเวิร์ดบนวินโดวส์ โอเอสทู และ แมคอินทอชจำกัดที่ -31680 ถึง 31680 เท่านั้น ตัวเลขที่ต่อท้ายคำควบคุมจะถือเป็นส่วนหนึ่งของคำควบคุม จากนั้น คำควบคุมก็จะจบด้วยตัวอักษรว่าง หรือตัวที่ไม่ใช่ชุดอักขระ เช่นเดียวกันคำควบคุมอื่น ๆ
- ตัวใด ๆ ที่ไม่ใช่ชุดอักขระ ในกรณีนี้ ตัวจบนี้ไม่ถือเป็นส่วนหนึ่งของคำควบคุม ถ้าตัวอักษรว่างเป็นตัวจบคำควบคุม ตัวอักษรว่างจะไม่ปรากฏในเอกสาร ในขณะที่ตัวอักษรอื่นที่ตามท้ายตัวจบ รวมถึงตัวอักษรว่างจะปรากฏในเอกสารด้วย

ตัวอย่างเช่น \rtf LetterSequence คือ rtf และ Delimiter คือ ตัวอักษรว่าง

สัญลักษณ์ควบคุม ประกอบด้วย "\" แล้วตามด้วย ตัวที่ไม่ใช่ตัวเลขตัวอักษร 1 ตัวเท่านั้น เช่น \~ ดังนั้น สัญลักษณ์ควบคุมจะไม่มีตัวจบเสมอ

กลุ่ม ประกอบด้วย ข้อความกับคำควบคุม หรือสัญลักษณ์ควบคุม ที่อยู่ภายในเครื่องหมายปีกกา "{ }" เครื่องหมายปีกกาเปิด "{" แสดงถึงการเริ่มต้นกลุ่ม และเครื่องหมายปีกกาปิด "}" แสดงถึงการจบกลุ่ม ในแต่ละกลุ่มจะเป็นตัวกำหนดคุณสมบัติที่ต่างกันของข้อความ ในแฟ้มข้อมูล RTF สามารถมีกลุ่มของฟอนต์ รูปแบบตัวอักษร สี รูปภาพ footnote annotation header/footer ข้อมูลสรุป field และ bookmarks รวมถึง การกำหนดรูปแบบของเอกสาร section paragraph และตัวอักษร

ตัวอย่างเช่น {\f15\fswiss\fcharset222\fprq2 CordiaUPC;} เป็นการกำหนดชื่อฟอนต์

เนื่องจาก เครื่องหมาย "\" "{" และ "}" มีความหมายพิเศษใน RTF ดังนั้น ถ้าต้องใช้ตัวเหล่านี้เป็นข้อความธรรมดา จะต้องนำหน้าด้วย "\" เช่น \\ \{ และ \}

2.2.2 ขั้นตอนการทำงานของโปรแกรมอ่าน RTF

โปรแกรมอ่านแฟ้มข้อมูล RTF จะต้องคำนึงถึงสิ่งต่อไปนี้

- แยกแยะข้อมูลที่เป็นตัวควบคุม ออกจากข้อความธรรมดา
- ทำงานตามตัวควบคุมนั้น
- เก็บรวบรวมและวางข้อความลงในเอกสารอย่างถูกต้องตามคุณลักษณะที่กำหนดในกลุ่มนั้น

การทำงานตามตัวควบคุม ถูกออกแบบมาให้เป็นขั้นตอนง่าย ๆ คือ ตัวควบคุมบางตัวจะเปลี่ยนลักษณะของข้อความ ตัวควบคุมอีกส่วนหนึ่งจะเปลี่ยนสถานะของโปรแกรม หมายถึง การกำหนดคุณลักษณะใหม่ของเอกสารทั้งหมด และตัวควบคุมที่เหลือเพียงแต่เปลี่ยนสถานะของกลุ่ม ซึ่งจะมีผลกับส่วนหนึ่งของเอกสารเท่านั้น

ในทางปฏิบัติ โปรแกรมอ่าน RTF จะพิจารณาตัวอักษรแต่ละตัวตามลำดับ ดังนี้

- ถ้าเป็น "{" โปรแกรมอ่านจะเก็บสถานะปัจจุบันไว้ใน Stack ถ้าเป็น "}" โปรแกรมอ่านจะดึงสถานะปัจจุบันออกจาก Stack
- ถ้าเป็น "\" โปรแกรมอ่านจะรวบรวมคำควบคุม หรือสัญลักษณ์ควบคุม พร้อมทั้งพารามิเตอร์ (ถ้ามี) แล้วตรวจดูในตารางว่าตัวควบคุมนั้นให้ทำงานอะไร จากนั้นจึงทำงานตามนั้น โดยที่ตัวชี้ของการอ่านอาจอยู่ในตำแหน่งก่อนหรือหลังตัวจบคำควบคุมก็ได้ แล้วแต่ความเหมาะสม
- ถ้าเป็นตัวอื่น ที่ไม่ใช่ "{" "}" และ "\" โปรแกรมอ่านจะเข้าใจว่าเป็นข้อความธรรมดา ก็จะเขียนข้อความนั้นด้วยคุณลักษณะที่ได้รับการกำหนดไว้

2.2.3 ส่วนประกอบของแฟ้มข้อมูล RTF

ข้อมูลในแฟ้มข้อมูล RTF มีรูปแบบ ดังนี้

"{"<header><document>"}"

โดยที่

<header> \rtf<charset>\deff?<fonttbl><filetbl>?<colortbl>?<stylesheet>?<revtbl>?

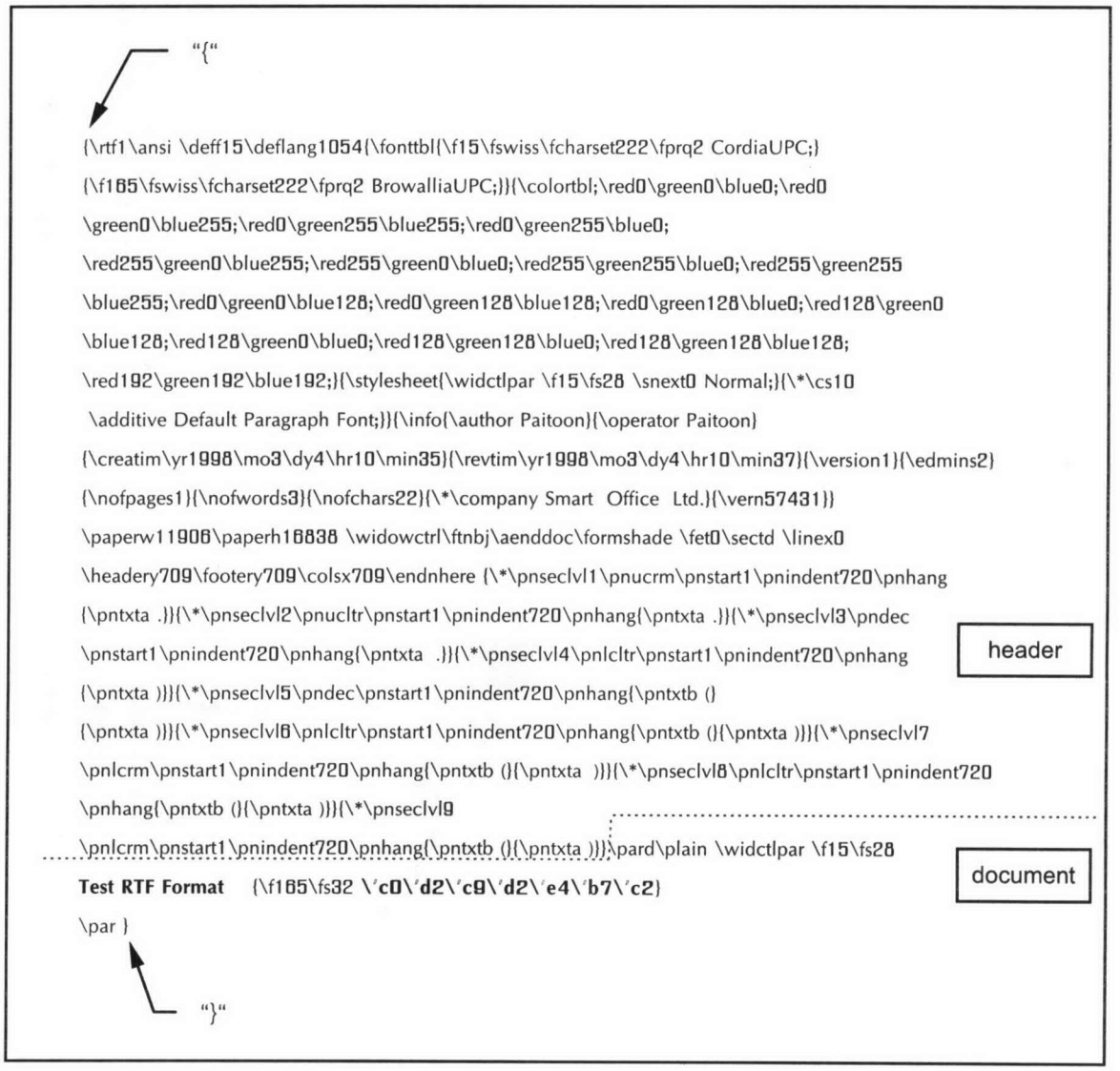
<document> <info>?<docfmt>*<section>+

โดยที่

a? มีหรือไม่มีก็ได้

a* มี 1 หรือมากกว่า 1 ก็ได้

a+ ไม่มีเลย หรือมีมากกว่า 1 ก็ได้




รูปที่ 2-4 แสดงส่วนประกอบของแฟ้มข้อมูล RTF

## 2.3 มาตรฐาน OLE (Object Link Embeded) [3]

OLE เป็นวิธีการสื่อสารระหว่างโปรแกรมประยุกต์บนวินโดวส์ เพื่อเพิ่มประสิทธิภาพให้กับวิธีการแบบ DDE (Dynamic Data Exchange) เราสามารถสร้างโปรแกรมประยุกต์ตามแบบ OLE โดยมีออบเจคที่ยอมให้โปรแกรมประยุกต์อื่นเรียกใช้ได้ ในโปรแกรมประยุกต์หนึ่ง ๆ อาจมีหลาย ๆ ออบเจคที่ยอมให้โปรแกรมประยุกต์อื่นเรียกใช้ โปรแกรมเมอร์สามารถเรียกใช้ออบเจคของโปรแกรมประยุกต์นั้นจากภายนอกผ่าน OLE ที่เป็นตัวกำหนดวิธีการติดต่อกับออบเจคภายในโปรแกรมประยุกต์เดียวกัน หรือระหว่างกัน

OLE มีสมาชิก 2 ประเภทที่จะใช้ติดต่อกับโปรแกรมประยุกต์อื่น ได้แก่

- วิธีการ (Method) คือ หน้าที่หรือความสามารถที่ออบเจคนั้นทำได้
- คุณสมบัติ (Property) คือ คุณลักษณะของออบเจค ซึ่งเป็นข้อมูลทั่วไปประจำตัว ออบเจคนั้น

โปรแกรมประยุกต์ที่พัฒนาจะเรียกว่าเป็น OLE server ก็ต่อเมื่อยอมให้โปรแกรมประยุกต์อื่นเรียกใช้ออบเจคของตัวเองได้ ข้อดีของการสร้างโปรแกรมประยุกต์ให้เป็น OLE ได้แก่

- ทำให้โปรแกรมประยุกต์อื่นหลาย ๆ โปรแกรมสามารถเรียกใช้โปรแกรมประยุกต์ที่เป็น OLE ร่วมกันได้ เป็นการนำโปรแกรมส่วนนั้นกลับมาใช้ใหม่ได้โดยไม่ต้องสิ้นเปลืองเวลาในการพัฒนาใหม่
- ผู้พัฒนาสามารถพัฒนาโปรแกรมประยุกต์ใหม่โดยเรียกใช้ ส่วนประกอบของ OLE ที่สร้างไว้แล้วด้วยเครื่องมือ หรือภาษาคอมพิวเตอร์ภายใต้สภาวะแวดล้อมแบบวินโดว์
- ออบเจคในโปรแกรมประยุกต์ที่เป็น OLE จะสามารถถูกเรียกใช้ได้โดยง่ายจาก macro language หรือ programming tool ที่สนับสนุนมาตรฐาน OLE
- สามารถปรับปรุงประสิทธิภาพให้กับตัวเชื่อมประสานของออบเจคได้โดยปราศจากความผิดพลาดในโปรแกรมประยุกต์ที่ใช้อยู่ในปัจจุบัน