



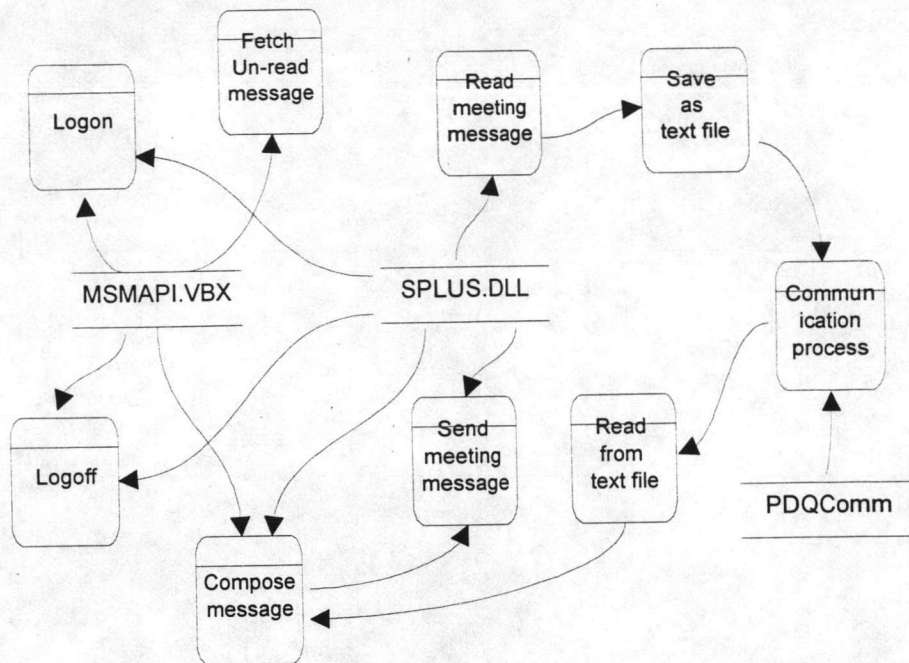
บทที่ 4

แหล่งที่มาของฟังก์ชันในระบบการจัดการนัดหมาย

ในโปรแกรมประยุกต์ระบบการจัดการนัดหมายที่เขียนขึ้น ได้เรียกใช้ฟังก์ชันต่างๆ จากส่วนควบคุมประติษฐ์ภายใต้ชื่อแฟ้มข้อมูล MSMAPI.VBX เช่น ฟังก์ชันในการลงชื่อขอเข้าใช้งาน หรือ ฟังก์ชันการค้นหาข่าวสารที่เข้ามาสู่กล่องเอกสารรับเข้า ตลอดจนการลงชื่อออกจากระบบ ส่วนการอ่านรายละเอียดของข่าวสาร จะใช้ฟังก์ชันจากคลังคำสั่งการเชื่อมแบบพลวัตที่อยู่ภายใต้ชื่อแฟ้มข้อมูลชื่อ SPLUS.DLL เนื่องจากในวิซวลเบสิก ยังไม่มีส่วนควบคุมประติษฐ์ใดที่มีฟังก์ชันที่ทำหน้าที่เกี่ยวกับการจัดการตารางเวลาอย่าง SPLUS.DLL ทำ โดยฟังก์ชันใน SPLUS.DLL จะอ่านเมสเสจเกี่ยวกับรายละเอียดการนัดหมาย ซึ่งเกี่ยวข้องกับตารางเวลาของผู้ใช้ด้วย

นอกจากนี้ ระบบการจัดการนัดหมายยังใช้ฟังก์ชันจากส่วนควบคุมประติษฐ์ภายใต้ชื่อแฟ้มข้อมูลที่ชื่อ PDQCOMM.VBX ซึ่งเป็นฟังก์ชันที่ใช้ในการติดต่อสื่อสารผ่านสายโทรศัพท์และโมเด็ม โดยบริษัทผู้ทำการพัฒนา ได้พัฒนาขึ้นจากพื้นฐานการทำงานของฟังก์ชันจากส่วนควบคุมประติษฐ์เดิมภายใต้ชื่อแฟ้มข้อมูล MSCOMM.VBX ซึ่งเป็นส่วนควบคุมเกี่ยวกับการสื่อสารที่มาพร้อมกับไมโครซอฟต์วิซวลเบสิก รุ่นที่ 3.0 อยู่แล้ว แต่ไม่มีส่วนควบคุมการทำงานของโมเด็มและการโอนถ่ายแฟ้มข้อมูล

รูปที่ 4.1 แสดงการเรียกใช้ฟังก์ชันในกระบวนการต่างๆ ในระบบการจัดการนัดหมาย ตั้งแต่การลงชื่อเข้าสู่ระบบ และในการทำงานตามกระบวนการต่างๆ จนกระทั่งเลิกการทำงานด้วยการลงชื่อออกจากระบบ โดยในส่วนการทำงานของแต่ละกระบวนการ สามารถเขียนเป็นส่วนของโปรแกรมได้ดังตัวอย่าง ซึ่งแยกไว้แล้วในแต่ละกระบวนการ บรรทัดที่ขีดเส้นใต้ หมายถึง บรรทัดที่เรียกใช้ฟังก์ชัน



รูปที่ 4.1 แสดงการใช้ฟังก์ชันในกิจกรรมต่างๆ ในระบบการจัดการนัดหมาย

กระบวนการ Logon

```

user.Reserved = 0
user.RecipClass = 0
user.Name = ""
user.Address = ""
user.EIDSize = 0

VbApmt.MapiSess.Action = SESSION_SIGNON .....1

If Err <> 0 Then
    MsgBox "Logon Failure: " + Error$
Else
    SPlusSession = 0
    e& = SPlusBeginSession(VbApmt.MapiSess.SessionID, 0, 0, 0, SPlusSession) .....2
    If (e& <> success_success) Then
        MsgBox "SPlusBeginSession error=" + Str$(e&)
    End If
End If
    
```

รูปที่ 4.2 แสดงส่วนของโปรแกรมในการเรียกใช้ฟังก์ชันในกระบวนการ Logon

จากรูปที่ 4.2 การเรียกใช้ฟังก์ชันในบรรทัดแรกที่คำสั่ง Action เพื่อลงชื่อขอเข้าทำงานกับระบบไปรษณีย์อิเล็กทรอนิกส์ ส่วนในบรรทัดที่สองเพื่อขอเข้าใช้ข้อมูลต่างๆ ที่เกี่ยวกับตารางเวลา สังเกตคำสั่งในบรรทัดที่สองต้องใช้ SessionID ที่ได้จากคำสั่งในบรรทัดแรกเสมอ

กระบวนการ Fetch Un-read message

VbApmt.MapiMess.FetchUnreadOnly = True1
VbApmt.MapiMess.Action = MESSAGE_FETCH2
MsgNo = VbApmt.MapiMess.MsgCount3

รูปที่ 4.3 แสดงส่วนของโปรแกรมในการเรียกใช้ฟังก์ชันในกระบวนการ Fetch Un-read message

จากรูปที่ 4.3 คำสั่งในบรรทัดแรกเป็นการกำหนดเงื่อนไขว่าจะใช้ข้อมูลที่ยังไม่เคยถูกเปิดออกอ่าน ส่วนบรรทัดที่สองคือบรรทัดของการดึงข้อมูลขึ้นมาตามเงื่อนไขที่ได้ระบุ จำนวนข่าวสารจะถูกเก็บไว้ใน MsgCount ซึ่งเป็นหนึ่งในคุณสมบัติภายใต้ส่วนควบคุม MAPI Messages

กระบวนการ Read Meeting Message

ulResult = SPLUSReadMeeting(VbApmt.MapiSess.SessionID, 0, RqstMessageID, 0, 0, RqstMessage, RqstOriginator, RqstRecipients(), RqstFiles(), RqstSentFor(), RqstAppt, RqstAssoc(), RqstAttendees())	
If (ulResult <> 0) Then	
MsgBox "Error reading requests!" + Str\$(ulResult)	
Exit Sub	
End If	

รูปที่ 4.4 แสดงส่วนของโปรแกรมในการเรียกใช้ฟังก์ชันในกระบวนการ Read Meeting Message

ตัวแปรที่ต้องส่งไปในการเรียกใช้ฟังก์ชัน SPLUSReadMeeting คือ SessionID ซึ่งเป็นหนึ่งในคุณสมบัติภายใต้ส่วนควบคุม MAPI Session ได้มาจากกระบวนการ Logon ผลลัพธ์ที่ได้จากการเรียกใช้ฟังก์ชันจะได้ข้อมูล ดังต่อไปนี้

- RqstMessageID เลขที่เมสเสจ อยู่ในรูปแบบตัวอักษร
- RqstMessage รายละเอียดเนื้อหาข่าวสาร อยู่ในรูปแบบโครงสร้างข้อมูล SPlusMessage
- RqstOriginator รายละเอียดผู้ส่งข่าวสาร อยู่ในรูปแบบโครงสร้างข้อมูล MapiRecip

RqstRecipients	รายละเอียดผู้รับข่าวสาร อยู่ในรูปแบบโครงสร้างข้อมูล MapiRecip
RqstFiles()	รายละเอียดแฟ้มเอกสารแนบ อยู่ในรูปแบบโครงสร้าง MapiFile แต่ยังไม่มีการใช้งานสำหรับข่าวสารที่เกิดจากการนัดหมาย
RqstSentFor()	ถ้าเป็นข่าวสารขอทำการนัดหมาย ข้อมูลส่วนนี้ จะอธิบายรายละเอียดผู้รับเอกสารแต่ละราย (ผู้รับมีได้มากกว่าหนึ่ง ในกรณีทำการนัดหมายโดยปกติ แต่ในกรณีทำการนัดหมายผู้รับ ด้วยการกำหนด custom address โปรแกรมการจัดตารางเวลา ยอมให้กำหนดที่ละหนึ่งราย) แต่ถ้าเป็นข่าวสารตอบการนัดหมาย ข้อมูลส่วนนี้จะอธิบายรายละเอียดผู้ตอบการนัดหมาย ทั้งสองกรณีอยู่ในรูปแบบโครงสร้าง MapiRecip
RqstAppt	รายละเอียดตารางการนัดหมาย อยู่ในรูปแบบโครงสร้างข้อมูล SplusAppt
RqstAssoc()	ไม่มีการใช้งาน (ignored)
RqstAttendees()	ไม่มีการใช้งาน (ignored)

กระบวนการ Compose Message

```

VbAppt.MapiMess.Action = MESSAGE_COMPOSE
.
.
.
e& = SPLUSSaveAppt(SPlusSession, 0, user, Appt, Assoc(0), Attendees(0), 0,
0, ItemID)
If (e& <> success_success) Then
MsgBox "Error Creating Appointment. Error Returned= " + Str$(e&)
Exit Function
End If

```

รูปที่ 4.5 แสดงส่วนของโปรแกรมในการเรียกใช้ฟังก์ชันในกระบวนการ Compose Message

จากรูป คำสั่ง Action = MESSAGE_COMPOSE เป็นการสร้างเมสเสจ บุรุษไปรษณีย์ จะใช้หลังจากที่ได้รับเพิ่มข้อมูลตัวอักษรจากภายนอกระบบ เพื่อสร้างเมสเสจเตรียมส่งต่อให้ผู้รับจริง ในขณะที่เดียวกันก็ต้องสร้างตารางการนัดหมายขึ้น โดยใช้ข้อมูลจากเพิ่มข้อมูลตัวอักษรนั้น และมีบุรุษไปรษณีย์เป็นเจ้าของตาราง เพื่อทำหน้าที่แทนผู้ส่งข่าวสารจริง ข้อมูลตัวแปรที่ส่งไป คือ SPlusSession, Appt และ Attendees() โดย SPlusSession ได้จากการขอเข้าใช้ข้อมูลตารางเวลาในกระบวนการ Logon ส่วน Appt ซึ่งใช้เก็บรายละเอียดเวลานัดหมาย

โดยใช้รูปแบบโครงสร้าง SPlusAppt และใช้ข้อมูลจากแฟ้มข้อมูลตัวอักษร และ Attendees() ซึ่งใช้เก็บข้อมูลของผู้รับข่าวสารและสถานะของคำตอบว่าเป็นการตอบรับ หรือ ปฏิเสธ โดยใช้รูปแบบโครงสร้าง SPlusAttendee และใช้ข้อมูลจากแฟ้มข้อมูลตัวอักษร

กระบวนการ Send Meeting Message

```
e& = SPLUSSendMeeting(VbApmt.MapiSeas.SessionID, 0, Message, Recip(0),
Files, SentFor(0), Appt, Assoc(0), Attendees(0), 0, 0)
If (e& <> success_success) Then
MsgBox "Error Sending Meeting Request. Error Returned= " + Str$(e&)
Exit Function
End If
```

รูปที่ 4.6 แสดงส่วนของโปรแกรมในการเรียกใช้ฟังก์ชันในกระบวนการ Send Meeting Message

จากรูปที่ 4.6 ฟังก์ชันจะทำหน้าที่ส่งเมสเสจที่เกิดจากกระบวนการ Compose Message ส่วนหนึ่ง และเกิดจากการกำหนดค่าด้วยค่าในแฟ้มข้อมูลตัวอักษรเป็นส่วนใหญ่ โดย

Message	รายละเอียดเนื้อหาข่าวสาร อยู่ในรูปแบบโครงสร้างข้อมูล SPlusMessage เก็บข้อมูลที่อ่านขึ้นมาจากแฟ้มข้อมูลตัวอักษร
Recip()	รายละเอียดผู้รับข่าวสาร อยู่ในรูปแบบโครงสร้างข้อมูล MapiRecip
Files	รายละเอียดแฟ้มเอกสารแนบ อยู่ในรูปแบบโครงสร้าง MapiFile แต่ยังไม่มีการใช้งานสำหรับข่าวสารที่เกิดจากการนัดหมาย
SentFor(0)	ถ้าเป็นข่าวสารขอทำการนัดหมาย จะไม่มีการใช้ข้อมูลส่วนนี้ แต่ถ้าเป็นข่าวสารตอบการนัดหมาย ข้อมูลส่วนนี้จะอธิบายรายละเอียดผู้ตอบการนัดหมาย อยู่ในรูปแบบโครงสร้าง MapiRecip
Appt	รายละเอียดตารางการนัดหมาย อยู่ในรูปแบบโครงสร้างข้อมูล SplusAppt
Assoc()	ไม่มีการใช้งาน (ignored)
Attendees()	รายละเอียดผู้มีส่วนร่วมในการนัดหมาย รวมทั้งสถานะคำตอบการนัดหมาย

กระบวนการ Communication Process

```

VbApmt.Comm1.Output = txtFrom$ & CR$ .....1
VbApmt.Comm1.Output = txtFromTel$ & CR$ .....2
VbApmt.Comm1.Output = "u" & CR$ .....3
VbApmt.Comm1.Upload = FileName$ .....4

Do
  Dummy = DoEvents()
Loop Until VbApmt.Comm1.XferStatus

Do While DoEvents()
  CurStatus = VbApmt.Comm1.XferStatus
  If CurStatus = PDQ_XFER_TERM_OK Then
    transStat = True
    Exit Do
  ElseIf CurStatus = PDQ_XFER_TERM_ERROR Then
    transStat = False
    Exit Do
  End If
Loop

If transStat Then
  VbApmt.Comm1.Disp = "Upload successful" & CRLF$
Else
  VbApmt.Comm1.Disp = "Upload failed" & CRLF$
End If

```

รูปที่ 4.7 แสดงส่วนของโปรแกรมในการเรียกใช้ฟังก์ชันในกระบวนการ Communication Process

จากรูปที่ 4.7 คำสั่งที่ส่งออกเริ่มจาก ส่งชื่อ, เบอร์โทรศัพท์, และในคำสั่งที่สามส่ง 'u' เพื่อให้ผู้รับปลายทางทราบว่าต้องการทำการส่งเพิ่มข้อมูลออกไป จากนั้นจึงทำการ upload เพิ่มข้อมูลด้วยคำสั่งที่สี่ ทำนองเดียวกัน ถ้าต้องการรับข้อมูลเข้ามา เช่น ในกรณีทวงถามคำตอบ ก็ จะส่ง 'd' ในคำสั่งที่สาม และคำสั่งที่สี่จะเป็น VbApmt.Comm1.download = FileName\$

คำสั่ง Do-Loop ทั้ง 2 คู่ที่ตามมาสำคัญมาก เนื่องจากคำสั่ง upload และ download เป็นคำสั่งที่ทำงานอยู่เบื้องหลัง ฉะนั้นหากมีคำสั่งตามหลังคำสั่ง upload หรือ download และเป็นคำสั่งที่ใช้ CPU ตลอดเวลา จะทำให้งานที่อยู่เบื้องหลังไม่ได้รับความสนใจ ดังนั้นเพื่อให้การส่งข้อมูลไปตามสายสื่อสารดำเนินไปจนเสร็จสมบูรณ์ จึงให้เพิ่มคำสั่ง Do-Loop และใช้คำสั่ง DoEvents เพื่อให้ CPU ไปดูผลงานที่อยู่เบื้องหลังบ้าง

กระบวนการ Logoff

```

If (SPlusSession <> 0) Then
  e& = SPlusEndSession(SPlusSession, 0, 0, 0) .....1
  SPlusSession = 0
  If (e& <> success_success) Then
    MsgBox "SPlusEndSession error=" + Str$(e&)
  End If
End If

VbApmt.MapiSess.Action = SESSION_SIGNOFF .....2
If Err <> 0 Then
  MsgBox "Logoff Failure: " + ErrorR
End If

```

รูปที่ 4.8 แสดงส่วนของโปรแกรมในการเรียกใช้ฟังก์ชันในกระบวนการ Logoff

จากตัวอย่าง ก่อนที่จะลงชื่อออกจากระบบตามคำสั่งที่สอง จะต้องตรวจสอบให้แน่ใจว่าได้ทำการบอกเลิกการใช้ข้อมูลตารางเวลาแล้วด้วยการใช้คำสั่งที่หนึ่ง

จากตัวอย่างการใช้ฟังก์ชันข้างต้น ส่วนต่อไปนี้เป็นบทสรุปเกี่ยวกับการใช้ส่วนควบคุมประติษฐ์ และคลังคำสั่งการเชื่อมแบบพลวัตที่ใช้ในระบบการจัดการนัดหมาย ซึ่งมีดังนี้

1. ส่วนควบคุม MAPI
2. ส่วนควบคุม PDQComm
3. คลังคำสั่งเพื่อการเข้าถึงข้อมูลในโปรแกรมการจัดการตารางเวลา (The Schedule+ Access Libraries)

ส่วนควบคุม MAPI¹

การพัฒนาโปรแกรมประยุกต์ เพื่อให้สามารถทำงานบนเครื่องคอมพิวเตอร์จากหลากหลายบริษัท เป็นปัญหาใหญ่สำหรับบริษัทผู้ผลิตซอฟต์แวร์ เนื่องจากไม่ต้องการจำกัดความสามารถของโปรแกรมอยู่เพียงบนเครื่องใดเครื่องหนึ่ง Messaging API ที่ก่อกำเนิดขึ้นในปัจจุบัน ทำให้โปรแกรมประยุกต์สามารถเข้าถึงระบบการส่งข่าวสารได้อย่างกว้างขวางขึ้น เนื่องจากเป็น

¹ "Custom Control Reference" in Microsoft Visual Basic Version 3.0, (The United States of America : Microsoft Corporation, 1993), pp. 179-185.

API ที่อิสระจากระบบเครือข่าย (network-independent API) และเหมาะสมกับโปรแกรมประยุกต์ไม่ว่าจะเป็นงานด้านธนาคารหรือด้านอื่นๆ

Messaging API² มีส่วนติดต่อกับผู้ใช้ในลักษณะรูปภาพ (graphical user interface) ที่ดี ซึ่งมีผลทำให้ผู้ใช้ใช้งานง่ายและมีประสิทธิภาพ ถ้าไม่มีส่วนติดต่อเหล่านี้ ผู้ใช้โดยเฉลี่ยแล้ว จะไม่สามารถใช้ประโยชน์จากเครื่องคอมพิวเตอร์อย่างเต็มที่ และโปรแกรมประยุกต์ทางธุรกิจก็เช่นกัน ไม่สามารถใช้ประโยชน์อย่างเต็มที่จากความก้าวหน้าทางเทคโนโลยีของระบบเครือข่าย เนื่องจากไม่สามารถทะลุทะลวงเข้าไปในเทคโนโลยีที่แตกต่างกัน ของเครือข่ายที่แตกต่างกัน จึงเป็นโอกาสดีที่มีผู้คิดค้นและพัฒนาให้สามารถใช้ประโยชน์ได้อย่างกว้างขวาง

Messaging API ในวิซวลเบสิกเช่นกัน เป็น API ที่ทำให้โปรแกรมเมอร์สามารถเพิ่มการทำงานด้านไปรษณีย์อิเล็กทรอนิกส์ในโปรแกรมประยุกต์ได้โดยง่าย ฟังก์ชันส่วนใหญ่เรียกกรอบโต้ตอบมาตรฐาน (standard dialog box) ขึ้นมาใช้งาน ฟังก์ชันเหล่านี้ สามารถใช้งานได้ดี และ ไม่ยุ่งยากสำหรับโปรแกรมประยุกต์ในการเรียกใช้งาน ในวิซวลเบสิกเราจะเรียก Messaging API ว่า เป็น ส่วนควบคุม MAPI

ส่วนควบคุม MAPI ถูกเก็บอยู่ในแฟ้มข้อมูล MSMAPI.VBX โปรแกรมเมอร์สามารถใช้ส่วนควบคุม MAPI เพื่อช่วยในการเขียนโปรแกรมประยุกต์ ที่ทำงานเกี่ยวกับการส่งข่าวสารไปรษณีย์อิเล็กทรอนิกส์ โดยแบ่ง ส่วนควบคุม MAPI ออกเป็น 2 ส่วน คือ ส่วนควบคุม MAPI session และ ส่วนควบคุม MAPI messages โดยส่วนควบคุม MAPI session จะสร้าง MAPI session ในขณะที่ส่วนควบคุม MAPI message ทำให้โปรแกรมเมอร์สามารถเรียกใช้งานหลากหลายฟังก์ชันที่เกี่ยวกับระบบข่าวสาร (ดูรายละเอียดภายใต้หัวข้อ ประเภทของส่วนควบคุม MAPI)

เนื่องจาก ผู้ใช้ไม่สามารถมองเห็นความเคลื่อนไหวของส่วนควบคุม MAPI ในระหว่างการทำงานตามชุดคำสั่ง (run time) ได้ และไม่มีเหตุการณ์ (event) ใดๆ สำหรับส่วนควบคุมเหล่านี้ ฉะนั้นในการใช้งาน จึงต้องกำหนดค่าคุณสมบัติ ของ Action ให้สอดคล้องกันกับงานที่ต้องการ

² Steven S. King, Contributing Editor, "Distributed Software," Data Communications 92 (April 1992): 85.

ส่วนควบคุม MAPI จะสามารถทำงานได้ ต้องมีส่วนบริการ MAPI (MAPI services) ซึ่งมีอยู่ในระบบไปรษณีย์อิเล็กทรอนิกส์ชื่อไมโครซอฟต์แมสอินไมโครซอฟต์วินโดวส์รุ่นที่ 3.0 หรือ 3.0 ขึ้นไป

ประเภทของส่วนควบคุม MAPI

1. MAPI Session
2. MAPI messages

1. MAPI Session

ส่วนควบคุม MAPI session ใช้เพื่อลงทะเบียน และสร้าง MAPI session และยังคงใช้ในการลงทะเบียนก่อนออกจากระบบด้วย โดยหลังการลงทะเบียนสำเร็จ SessionID จะเก็บค่าที่ใช้ในการควบคุม MAPI session ค่า session handle จะต้องถูกส่งผ่านไปส่วนควบคุม MAPI messages มิฉะนั้นจะเกิดข้อผิดพลาดเมื่อมีการเรียกใช้ส่วนควบคุม MAPI messages ในภายหลัง

ก่อนที่จะออกจากระบบ ต้องแน่ใจว่ามีการใช้ส่วนควบคุม MAPI session เพื่อลงทะเบียนออกจากระบบเสมอหากมีการเริ่มต้นด้วยการลงทะเบียน และเพิ่มข้อมูล CONSTANT.TXT ภายใต้วินโดวส์ชื่อ VB จะบอกรายละเอียดสำหรับการกำหนดค่าคุณสมบัติ และค่าคงที่ที่บอเกี่ยวกับข้อผิดพลาดที่อาจจะเกิดขึ้นได้สำหรับส่วนควบคุมนี้

ทุกคุณสมบัติภายใต้อินเตอร์เฟซ MAPI Session แสดงตามตารางที่ 4.1 คุณสมบัติซึ่งใช้ได้เฉพาะส่วนควบคุมนี้ หรือต้องการการพิจารณาเป็นพิเศษเมื่อจะใช้งาน คือคุณสมบัติที่มีดอกจัน (*) อยู่ข้างหน้า

ตารางที่ 4.1 แสดงคุณสมบัติภายใต้อินเตอร์เฟซ MAPI session

Properties		
About	*LogonUI	Tag
*Action	Name ²	Top
*DownloadMail	*NewSession	*UserName
Index ¹	*Password	
Left	*SessionID	

¹ ใช้ใน Visual basic เท่านั้น
² เหมือนกับ Citename property ใน Visual Basic 1.0 และ Visual C++

2. MAPI Messages

หลังจากที่ session ของเมสเสจถูกสร้างขึ้นแล้วพร้อมด้วยส่วนควบคุม MAPI session ส่วนควบคุม MAPI messages สามารถเรียกใช้ฟังก์ชันต่างๆ ของระบบข่าวสาร ด้วยความสามารถของส่วนควบคุม MAPI messages นี้เองที่ทำให้ผู้ใช้สามารถ

1. อ่านข่าวสารซึ่งมีอยู่ในกล่องเอกสารเข้า (Inbox)
2. สร้างข่าวสารใหม่
3. เพิ่มและลบผู้รับข่าวสารและเอกสารแนบ
4. ส่งข่าวสาร โดยมี หรือ ไม่มีจอภาพติดต่อระหว่างผู้ใช้ (Supporting user interface)
5. เก็บ, ทำสำเนา และลบข่าวสาร
6. แสดงจอภาพใช้ในการโต้ตอบเกี่ยวกับสมุดที่อยู่ (address book)
7. แสดงจอภาพรายละเอียด
8. อ่านเอกสารแนบ รวมทั้งเอกสารแนบที่เป็น Object Linking and Embedding (OLE)
9. ตรวจสอบชื่อผู้รับในการกำหนดที่อยู่
10. เรียกใช้ฟังก์ชัน reply, reply-all และ forward ข่าวสารต่างๆ

คุณสมบัติส่วนใหญ่ของส่วนควบคุม MAPI message สามารถที่จะแบ่งออกเป็น 4 หน้าที คือ คุณสมบัติที่ทำหน้าที่เกี่ยวกับสมุดที่อยู่, คุณสมบัติที่ทำหน้าที่เกี่ยวกับเพิ่มเอกสารแนบ, คุณสมบัติที่ทำหน้าที่เกี่ยวกับตัวข่าวสาร และคุณสมบัติที่ทำหน้าที่เกี่ยวกับผู้รับ เพิ่มเอกสารแนบ, ข่าวสาร และผู้รับถูกควบคุมด้วยคุณสมบัติของ AttachmentIndex, msgIndex และ RecipIndex ตามลำดับ

ตัวอย่างเช่น เมื่อค่าดัชนีในคุณสมบัติ MsgIndex มีการเปลี่ยนแปลงไป ทั้งตัวข่าวสาร เอกสารแนบ และผู้รับ ก็จะมีการเปลี่ยนแปลงไปตามค่าดัชนีเพื่อแสดงข้อมูลของข่าวสารตามที่ค่าดัชนีอ้างอิง ชุดของข่าวสารและผู้รับจะทำงานในทิศทางเดียวกัน คุณสมบัติที่เกี่ยวกับสมุดที่อยู่ จะเป็นตัวกำหนดภาพลักษณ์ของจอภาพที่ใช้ในการโต้ตอบกับผู้ใช้เกี่ยวกับสมุดที่อยู่

บัฟเฟอร์ของเมสเสจ (Message Buffers)

การใช้ส่วนควบคุม MAPI messages โปรแกรมเมอร์จำเป็นต้องคอยตรวจสอบ บัฟเฟอร์ 2 บัฟเฟอร์ คือ บัฟเฟอร์เพื่อการสร้างเมสเสจ (Compose buffer) และบัฟเฟอร์เพื่อการอ่านเมสเสจ (read buffer)

บัฟเฟอร์เพื่อการอ่านเมสเสจ ถูกสร้างขึ้นจากชุดของข่าวสาร ซึ่งอ่านขึ้นมา จากช่องเอกสารรับเข้าของผู้ใช้โดยมี MsgIndex เป็นคุณสมบัติที่ใช้ในการเข้าถึงแต่ละข่าวสาร ภายในชุดข่าวสาร โดยเริ่มต้นค่าที่ 0 และเพิ่มค่าทีละหนึ่งสำหรับแต่ละข่าวสารจนครบทั้งชุด หรือกล่าวอีกนัยหนึ่ง ชุดของข่าวสารถูกสร้างขึ้นจากการกำหนดค่าคุณสมบัติ Action ให้เป็น MESSAGE_FETCH ซึ่งจะทำการอ่านข่าวสารทุกข่าวสารที่มีประเภทของข่าวสารตามที่กำหนดไว้ใน คุณสมบัติ FetchMsgType และเรียงลำดับข่าวสารตามที่กำหนดไว้ในคุณสมบัติ FetchSorted โปรแกรมเมอร์สามารถที่จะเลือกเฉพาะข่าวสารที่ยังไม่เคยอ่านเลย โดยกำหนดค่าในคุณสมบัติ FetchUnreadOnly ให้เป็น True

อย่างไรก็ตามข่าวสารที่อยู่ในบัฟเฟอร์เพื่อการอ่าน จะไม่สามารถทำการแก้ไข เนื้อความใดๆ ได้ แต่สามารถถ่ายสำเนาซึ่งถือเป็นการสร้างข่าวสารใหม่ โดยมีเนื้อหาภายใน เหมือนข่าวสารเดิม ไปไว้ที่บัฟเฟอร์เพื่อการสร้างเมสเสจ เพื่อทำการเปลี่ยนแปลงแก้ไขต่อไป

ข่าวสารต่างๆ ถูกสร้างขึ้น หรือแก้ไข ในบัฟเฟอร์เพื่อการสร้างเมสเสจ โดยบัฟเฟอร์เพื่อการสร้างเมสเสจจะทำงาน (active) เมื่อมีการกำหนดค่าประจำตัว MsgIndex ให้ เป็น -1 การกระทำหลายๆ อย่างของข่าวสาร สามารถทำงานได้เฉพาะในบัฟเฟอร์เพื่อการ สร้างเมสเสจเท่านั้น เช่น การส่งข่าวสาร, การส่งข่าวสารโดยมีกรอบเพื่อการโต้ตอบ, การบันทึก ข่าวสารหรือการลบรายชื่อผู้รับและเอกสารแนบ เป็นต้น

ทุกคุณสมบัติสำหรับส่วนควบคุม MAPI Messages สามารถแสดงตามตารางที่ 4.2 คุณสมบัติ ซึ่งใช้ได้เฉพาะส่วนควบคุมนี้ หรือต้องการการพิจารณาเป็นพิเศษเมื่อจะใช้งาน คือ คุณสมบัติที่มีดอกจัน (*) อยู่ข้างหน้า

ตารางที่ 4.2 แสดงคุณสมบัติภายใต้ส่วนควบคุม MAPI Messages

Properties		
About	*FetchMsgType	*MsgreceiptRequested
*Action	*FetchSorted	*MsgSen
*AddressCaption	*FetchUnreadOnly	*MsgSubject
*AddressEditFieldCount	Index ¹	*MsgType
*AddressLabel	*MsgConversationID	Name
*Addressmodifiable	*MsgCount	*RecipAddress
*AddressResolveUI	*MsgDateReceived	*RecipCount
*AttachmentCount	*MsgID	*RecipDisplayName
*AttachmentIndex	*MsgIndex	*RecipIndex
*AttachmentName	*MsgNoteText	*recipType
*AttachmentPathName	*MsgOrigAddress	*SessionID
*AttachmentPosition	*MsgOrigDisplayName	Tag
*AttachmentType	*MsgRead	Top

¹ ใช้ใน Visual basic เท่านั้น

² เหมือนกับ CItem property ใน Visual Basic 1.0 และ Visual C++

คุณสมบัติต่างๆ ข้างต้น จะทำงานกับโครงสร้างข้อมูลดังต่อไปนี้

ตารางที่ 4.3 แสดงโครงสร้างข้อมูลที่ทำงานกับฟังก์ชันในส่วนควบคุม MAPI

ชื่อโครงสร้างข้อมูล	คำอธิบาย
MapiFile	เก็บรายละเอียดแฟ้มข้อมูลเอกสารแนบ
MapiMessage	เก็บรายละเอียดข่าวสาร, จดหมาย
MapiRecip	เก็บรายละเอียดผู้รับ

ส่วนควบคุม PDQComm³

ส่วนควบคุม PDQComm ก็เป็นอีกหนึ่งส่วนควบคุมที่เรียกใช้งานภายใต้วิซวลเบสิก ที่ทำให้โปรแกรมประยุกต์ของผู้ใช้สามารถทำการติดต่อสื่อสารแบบอนุกรม (serial communication) ผ่านโมเด็มและสายโทรศัพท์ โดยยอมให้มีการส่ง (transmission) และการรับ (reception) ข้อมูล ผ่านช่องทางเข้าออกแบบอนุกรม (serial port) ได้ ฟังก์ชันต่างๆ ภายใต้ส่วนควบคุม PDQComm ถูกเก็บไว้ในแฟ้มข้อมูลชื่อ PDQCOMM.VBX และทำงานในลักษณะเดียวกันกับส่วนควบคุม MSCOMM.VBX แต่ได้รับการพัฒนาให้มีประสิทธิภาพในการทำงานมากขึ้น ดังนี้

³ PDQComm for Windows, (Ridgefield, CT 06877 : Crescent Software, 1994), chapter

1. เพิ่ม Baud rate ได้สูงถึง 230400 โดยทำงานร่วมกับไดรฟ์เวอร์เกี่ยวกับการสื่อสาร (communication driver) ภายใต้วินโดวส์
2. เพิ่มโมดูลในการส่งผ่านข้อมูล (File Transfers) แบบ XModem-Checksum, XModem-CRC, XModem-1K, YModem-Batch, YModem-G, ZModem, Kermit และ CompuServe-B+
3. การส่งผ่านข้อมูลทุกรูปแบบทำเป็นงานเบื้องหลัง (background)
4. เพิ่มข้อมูลที่รับเข้ามาถูกเปิดไว้ในโหมดการใช้งานร่วมกัน (shared mode) ฉะนั้นจึงสามารถเข้าถึงข้อมูลได้แม้ในระหว่างกำลังส่งผ่าน
5. สามารถแสดงสถานะปริมาณข้อมูลขณะส่งผ่านภายใต้กรอบโต้ตอบที่ผู้ใช้สร้างขึ้นเอง หรือผู้ใช้อาจจะเลือกที่จะไม่แสดงก็ได้
6. เป็น Terminal emulator ที่เป็นได้ทั้ง TTY, ANSI, VT-52 และ VT-100
7. สามารถเลือกใช้ฟอนต์ (font) มาตรฐานที่ใช้ภายใต้วินโดวส์

ทุกคุณสมบัติสำหรับส่วนควบคุม MAPI Messages สามารถแสดงตามตารางที่ 4.4

ตารางที่ 4.4 แสดงคุณสมบัติภายใต้ส่วนควบคุม PDQComm

<u>properties</u>		
About	*Echo	*Output
*AutoProcess	*Emulation	*ParityReplace
*AutoScroll	FontBold	*PortOpen
*AutoSize	FontItalic	*Rows
*BackColor	FontName	*Rthreshold
*BackSpace	FontSize	*RTSEnable
*Break	FontUnderline	*ScrollRows
*CaptureFilename	*ForeColor	*ScrollText
*CaptureMode	*Handshaking	SelLength
*CDHolding	Height	SelStart
*CDTimeout	HWnd	SelText
*ColorFilter	*InBufferCount	*Settings
*Columns	*InBufferSize	*SmoothScroll
*CommEvent	Index	*Sthreshold
*CommID	*Input	Tag
*CommPort	*InputLen	*Text
*CTSHolding	*Interval	Top
*CTSTimeout	*InTimeout	*Unload
*CursorColumn	*KeyTranslation	Width
*CursorRow	Left	*XferDestFilename
*CursorType	*LineInput	*XferCarrierAbort
*Disp	*Notification	*XferFileSize
*Download	Name	*XferMessage
*DSR Holding	*NullDiscard	*XferProtocol
*DSRTimeout	*OutBufferCount	*XferSourceFilename
*DTR Enable	*OutBufferSize	*XferStatus
		*XferStatusDialog
		*XferTransferred
<u>events</u>		
*OnComm		
* properties หรือ events ที่สร้างขึ้นหรือได้รับการแก้ไข		

คลังคำสั่งเพื่อการเข้าถึงข้อมูลในโปรแกรมการจัดตารางเวลา (The Schedule+ Access Libraries) ⁴

คลังคำสั่งเพื่อการเข้าถึงข้อมูลในโปรแกรมการจัดตารางเวลา เป็นคลังคำสั่งของชุดฟังก์ชัน ที่ช่วยในการพัฒนาโปรแกรมประยุกต์ เพื่อการจัดการนัดหมาย และ ลำดับของงาน ที่ต้องทำในปฏิทินของผู้ใช้ และสามารถแลกเปลี่ยนข่าวสารเพื่อการนัดประชุมระหว่างผู้ใช้โปรแกรมการจัดตารางเวลาด้วยกัน การใช้ฟังก์ชันต่างๆ ในคลังคำสั่งเพื่อการเข้าถึงข้อมูลในโปรแกรมการจัดตารางเวลา นี้ จะใช้ร่วมกันกับฟังก์ชันในชุดของ MAPI ไม่ว่าจะเป็น MAPI จากส่วนควบคุมประดิษฐ์ MSMAPI.VBX หรือ จากคลังคำสั่งการเชื่อมแบบพลวัต MAPI.DLL

ชุดฟังก์ชันในคลังคำสั่งที่ใช้เพื่อการเข้าถึงข้อมูลในโปรแกรมการจัดตารางเวลา ถูกเก็บไว้ในแฟ้มข้อมูลชื่อ SPLUS.DLL โดยสามารถแบ่งตามลักษณะการใช้งานได้ 2 ประเภทคือ

1. Schedule+ Access Libraries meeting message functions
ใช้ในการจัดการข่าวสารในการนัดหมาย
2. Schedule+ Access Libraries calendar function
ใช้ในการจัดการเกี่ยวกับปฏิทิน

ในที่นี้จะกล่าวถึงเฉพาะในส่วนที่เกี่ยวกับการจัดการข่าวสารในการนัดหมายเท่านั้น โดยแบ่งการอธิบายออกเป็น 2 ส่วนดังนี้

1. โครงสร้างข้อมูล (Schedule+ Access Libraries Types)
2. ฟังก์ชันที่ใช้ในการรับส่งข่าวสารการนัดหมาย

1. โครงสร้างข้อมูล อธิบายเกี่ยวกับโครงสร้างของข้อมูล ที่เกี่ยวข้องในการเขียนโปรแกรมประยุกต์เพื่อการอ่านหรือสร้างข่าวสารการนัดหมาย โดยจะต้องกำหนดโครงสร้างของข้อมูลเพื่อรองรับเมสเสจที่อ่านขึ้นมา หรือ เพื่อประกอบเป็นเมสเสจใหม่ ก่อนที่จะส่งออกไป

⁴ Microsoft Development Library, "Schedule+ SDK," (The United States of America: Microsoft Corporation, 1992, 1993), pp.1-22.

โปรแกรมเมอร์สามารถเปลี่ยนแปลงการนิยามค่าของโครงสร้าง จากเพิ่มข้อมูล SPLUS.TXT และค่ารหัสต่างๆ ที่ใช้ในวิซวลเบสิกจากเพิ่มข้อมูล SPLUS.BAS ในที่นี้จะแสดงโครงสร้างข้อมูลเฉพาะในส่วนที่ใช้งานในโปรแกรมการจัดการนัดหมายเท่านั้น โดยมีโครงสร้างที่เกี่ยวข้อง แสดงดังตารางที่ 4.5

ตารางที่ 4.5 แสดงโครงสร้างข้อมูลที่ใช้กับฟังก์ชันใน SPLUS.DLL

ชื่อโครงสร้างข้อมูล	คำอธิบาย
SPlusAppt	เก็บรายละเอียดเกี่ยวกับตารางการนัดหมายซึ่งเป็นข้อมูลหลัก เช่น ผู้สร้างตาราง, จำนวนผู้เข้าร่วมประชุม, ข้อความหรือหมายเหตุการประชุมตลอดจนช่วงเวลาการนัดหมาย
SPlusAttendee	เก็บข้อมูลของผู้เข้าร่วมประชุมด้วยโครงสร้างของ MapiRecip และ สถานะการยืนยันคำตอบว่า ตกลง หรือ ปฏิเสธ แต่ถ้าผู้รับยังไม่ส่งคำตอบก็จะปรากฏเป็นรูปจดหมายปิดผนึก หรือ เปิดผนึกแล้วแต่กรณีว่าผู้รับได้เปิดออกอ่านแล้วหรือไม่
SPlusMessage	เก็บรายละเอียดข่าวสารสำหรับข่าวสารการขอนัดหมาย เช่น ชื่อเรื่อง, ประเภทข่าวสาร, ข้อความข่าวสารและจำนวนผู้รับ เป็นต้น SPlusMessage จะมีรูปแบบคล้ายกับ MapiMessage
SPlusUserInfo	เก็บรายละเอียดเกี่ยวกับสมาชิกในกลุ่มทำงาน ว่ามีผู้ช่วย (assistant) หรือเลขานุการหรือไม่ ถ้ามีก็จะมีโครงสร้างข้อมูล MapiRecip ของผู้ช่วยด้วย

The screenshot shows a dialog box titled "Appointment". It has several sections:

- When:** Contains "Start:" with a time of "8:00 AM" and a date of "Tue 3/28/95", and "End:" with a time of "10:30 AM" and a date of "Tue 3/28/95". There is a "Choose Time..." button and a "Tentative" checkbox.
- Description:** A text field containing "sender->user".
- Attendees:** A list box containing "user : 9694947 : 0 : 0".
- Buttons:** "OK", "Cancel", and "Invite..." are on the right side.
- Options:** At the bottom, there is a checked checkbox "Set Reminder for" followed by "15" in a box, "minute(s)", and "Beforehand". There is also an unchecked checkbox "Private".

รูปที่ 4.9 จอภาพ Appointment แสดงเนื้อหาการนัดหมาย

เนื้อหาของโครงสร้างข้อมูลรูปแบบต่างๆ จากตารางที่ 4.5 จะปรากฏตามจอภาพต่างๆ ของโปรแกรมการจัดตารางเวลา ดูตัวอย่างจอภาพ Appointment ซึ่งแสดงเนื้อหา และแหล่งที่มาของข้อมูล ดังรูปที่ 4.9 ข้อมูลในกรอบของ When มาจากโครงสร้างข้อมูล SPlusAppt ข้อมูลในกรอบของ Description มาจากโครงสร้างข้อมูล SPlusMessage และข้อมูลในกรอบของ Attendees มาจากโครงสร้างข้อมูล SPlusAttendee

2. ฟังก์ชันที่ใช้ในการรับส่งข่าวสารการนัดหมาย สามารถรับ-ส่งประเภทของข่าวสารที่เกี่ยวข้องกับการนัดหมาย โดยแบ่งออกได้เป็น 5 ประเภทดังนี้ คือ

1. ข่าวสารขอทำการนัดหมาย (IPM.Microsoft Schedule.MtgReq)
2. ข่าวสารตอบรับการนัดหมาย (IPM.Microsoft Schedule.MtgRespP)
3. ข่าวสารปฏิเสธการนัดหมาย (IPM.Microsoft Schedule.MtgRespN)
4. ข่าวสารแจ้งความยังไม่แน่นอนสำหรับการนัดหมาย (IPM.Microsoft Schedule.MtgRespA)
5. ข่าวสารยกเลิกการนัดหมาย (IPM.Microsoft Schedule.MtgCncl)

ฟังก์ชันที่ใช้ในการรับส่งข่าวสารการนัดหมาย มีดังนี้คือ

1. SPLUSReadMeeting อ่านข่าวสารการนัดหมาย
2. SPLUSSendMeeting ส่งข่าวสารการนัดหมาย

สำหรับรายละเอียดการใช้ฟังก์ชันและโครงสร้างข้อมูลหาอ่านได้จากหนังสือ Schedule+ Template หรือ Schedule+ SDK ใน Microsoft Development Library โดย Microsoft Corporation