รหัสโครงการ SUT2-204-40-60-17

รายงานการวิจัย

การพัฒนาสื่อเสริมการสอนแบบ ระบบตอบรับข้อมูลผ่านเครือข่ายโทรศัพท์คอมพิวเตอร์

(Computer Telephony Integration Technology for Course Ware Development)

คณะผู้วิจัย

หัวหน้าโครงการ

ดร. ชนวัฒน์ ศรีสอ้าน

สาขาวิชาเทคโนโลยีสารสนเทศ

สำนักวิชาเทคโนโลยีสังคม

มหาวิทยาลัยเทคโนโลยีสุรนารี

ผู้ร่วมวิจัย

นายสุทธิชัย มณีรัตนรุ่งโรจน์

ได้รับทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ปีงบประมาณ พ.ศ. 2540

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

ตุลาคม 2542

กิตติกรรมประกาศ

งานวิจัยวิชาการเรื่อง ระบบตอบรับข้อมูลผ่านเครือข่ายโทรศัพท์คอมพิวเตอร์ สามารถสำเร็จสมบูรณ์และบรรลุวัตถุประสงค์ได้นั้นเพราะการวิจัยครั้งนี้ได้ทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ประจำปีงบประมาณ 2540 จึงใคร่ขอขอบพระคุณมา ณ โอกาสนี้

รายงานการวิจัยฉบับนี้ สำเร็จลุล่วงได้ด้วยดี จากความร่วมมือจากคณะทำงาน และความกรุณาของผู้ทรงคุณวุฒิหลายด้าน งานวิจัยฉบับนี้ไม่สามารถสำเร็จได้ถ้าปราศจากความช่วยเหลือผู้ช่วยวิจัย นายทศพร แรมจะบก และนางสาววิรัฐติกาล พุทธโฆษ์ และนายอมฤต เที่ยงดาห์ ที่ให้ให้งานวิจัยนี้สำเร็จลุล่วงไปได้

สุดท้ายขอขอบคุณ ผู้ร่วมวิจัย ที่ให้ความร่วมมือช่วยเหลืออย่างดี มา ณ โอกาสนี้

ดร. ชนวัฒน์ ศรีสอ้าน

5 ตุลาคม 2542

บทคัดย่อ

การวิจัยเรื่องนี้มีวัตถุประสงค์หลัก คือเพื่อวิจัย และพัฒนาต้นแบบระบบตอบรับข้อมูลผ่านเครือข่ายโทรศัพท์คอมพิวเตอร์ (Computer Telephony Integration Technology for Course Ware Development) นี้เป็นโครงการนำร่องที่พยายามประยุกต์ใช้เทคโนโลยีที่มีอยู่ในมหาวิทยาลัย เพื่อประโยชน์การเรียนการสอน โดยเบื้องต้น ผู้วิจัยพยายามจะลดเวลา การเฉลยแบบฝึกหัด การประกาศผล และกระจายเสียงเทปการบรรยายผ่านโครงข่ายโทรศัพท์ บนคอมพิวเตอร์ส่วนตัวของอาจารย์

เนื่องด้วย งานวิจัยและพัฒนาการสื่อสารข้อมูล ผ่านเครือข่ายโทรศัพท์ ด้านนี้ยังไม่มีการวิจัยและ พัฒนาอย่างจริงจังในการนำเอาเทคโนโลยีด้านนี้มาใช้เพื่อเสริมการสอนให้ประหยัดเวลาและ ค่าใช้จ่าย โดยแนวคิดของการวิจัย คือ การนำเอาคอมพิวเตอร์ส่วนบุคคลที่มีอยู่ทั่วไปในมหาวิทยาลัยมาใช้เป็นศูนย์บริการข่าวสารโดยไม่ต้องมีบุคลากรคอยบริการรับโทรศัพท์ และตอบคำถาม ซึ่งคอมพิวเตอร์จะปฏิบัติการเสมือนเป็นบุคลากรคอยรับโทรศัพท์และคอยตอบคำถาม บริการข่าวต่าง ๆ โดยคอมพิวเตอร์สามารถรับสายโทรศัพท์ได้หลาย ๆ เบอร์ในเวลาเดียว และการโต้ตอบ สนทนาจะเป็นอิสระจากกันของแต่ละเบอร์โทรศัพท์ นั่นหมายความว่าคอมพิวเตอร์ เครื่องหนึ่งสามารถรับโทรศัพท์แทนบุคลากรได้หลายคนจะทำให้มหาวิทยาลัยไม่ต้องจ้างพนักงานรับโทรศัพท์ไว้คอยตอบคำถามเพิ่มมากขึ้นอีก และเป็นการเบาแรงของบุคลากรคนที่มีหน้าที่เดิมอยู่แล้ว ทำให้สะดวกในการเปลี่ยนข้อความข่าวสาร การเผยแพร่ข่าวสารได้ทั่วถึง ไม่ล่าช้า และถูกต้อง

งานวิจัยและพัฒนานี้ได้สำเร็จในฐานะขอบเขตให้เป็นต้นแบบของระบบการขอทราบเกรด ทราบผลสอบย่อย (QUIZ) แบบฝึกหัด ตลอดจน ฟังเทปการบรรยายทางโทรศัพท์ และในอนาคต เครื่องต้นแบบดังกล่าวสามารถให้บริการนักศึกษาได้อย่างดี ทำให้ทั้งอาจารย์และศิษย์สามารถประหยัดเวลาทั้งสองฝ่าย โครงการวิจัยและพัฒนาจะสามารถ พัฒนาให้นักศึกษาสามารถลงทะเบียนผ่านทางโทรศัพท์ ในกาลต่อไป

ABSTRACT

The objective of the study on **"Computer Telephony Integration Technology for Course Ware Development"** is to conduct research and develop module for Information technology system over computer telephony network. The research is a pilot study to adopt technology available at Suranaree University of Technology (SUT) for teaching as well. I also aims at reducing faculty members' time spent on exercising, grade announcement and voice-taping distribution using computer telephony network.

The research about developing computer telephony has not been done and used to enhance leaning and reduce time and costs in teaching task. The purpose of the research is to use personal computer available for lecture at SUT for providing information on telephone without having personal contact to answer telephone and questions. Computer will act as personnel to answer telephone and provide information simultaneously. Furthermore each computer will work independently and has capacity equal to many people; therefore, SUT can save money and reduce labor for the personnel whose job is to do such jobs.

This result of this research is to be a pilot project for grading, quiz, exercise, tape, telephone for facilitating courses and found that it can serve the requirement perfectly. It also hopes to develop to allow student to register via telephone in the future.

สารบัญ

สารบัญภาพ

บทที่ 1
บทนำ

ความสำคัญและที่มาของปัญหาการวิจัย

โครงการวิจัยและพัฒนานี้ เป็นการประยุกต์ใช้เทคโนโลยีที่มีอยู่แล้วให้สามารถใช้ได้ในชีวิตประจำวันได้ โดยให้เกิดประโยชน์มากที่สุด ซึ่งเป็นผลทำให้เกิดความสะดวกสบาย และเป็นการผ่อนแรงแก่บุคลากร โดยไม่ต้องคอยรับโทรศัพท์และตอบคำถามซ้ำ ๆ และเป็นการทำงานที่ไม่ซ้ำซ้อนด้วย อีกทั้งยังช่วยให้เกิดความสะดวกแก่นักศึกษาและผู้สนใจอีกด้วย โดยจะสามารถโทรศัพท์มาสอบถามข่าว ประกาศต่าง ๆ เช่น ประกาศประชุม สัมมา การอบรมหลักสูตรคอมพิวเตอร์ โดยไม่ต้องเดินทางมาสอบถามด้วยตัวเองที่มหาวิทยาลัย รวมทั้งในเชิงเศรษฐกิจ ยังเป็นการช่วยกันประหยัดอีกด้วย โครงการวิจัยและพัฒนานี้เป็นการประยุกต์เทคโนโลยีในด้านเครือข่ายโทรศัพท์และคอมพิวเตอร์มาใช้งานเพื่อให้ได้ประโยชน์สูงสุดในการทำงาน ดังนั้นผู้วิจัยจำเป็นต้องมีความรู้พื้นฐานทางด้านเทคโนโลยีที่มีอยู่และทราบส่วนที่ต้องทำการวิจัยและพัฒนาต่อไป ดังนั้นในเอกสารโครงการวิจัยและพัฒนานี้จึงขอกล่าวเป็นส่วนซึ่งได้แก่ ต้นแบบอิเล็กทรอนิกส์ การพัฒนาและปรับแต่ง Software และผลของการพัฒนา ในรายงานนี้

วัตถุประสงค์ของการวิจัย

1. ให้ความสะดวกแก่นักศึกษาในการรับฟังข่าวสาร การเฉลยแบบฝึกหัด การประกาศผล และกระจายเสียงเทป การบรรยายผ่านโครงข่ายคอมพิวเตอร์ ในการเรียนการสอนของอาจารย์ผู้สอนได้จากหอพัก เพราะที่หอของนักศึกษามีโทรศัพท์ทุกห้อง

2. นักศึกษาสามารถทราบผลการสอบย่อย (QUIZ) และผลการสอบกลางภาคได้ โดยไม่ต้องเสียเวลาเดินทางมารับผลสอบจากเครื่องคอมพิวเตอร์ของอาจารย์ โดยไม่ต้องใช้เจ้าหน้าที่เพิ่ม

3. ระบบหมายเลขเดียว จะอำนวยความสะดวกเป็นอย่างยิ่ง แก่นักศึกษาหรือผู้สนใจเพราะง่ายต่อการจดจำ

4. ข้อมูลที่บันทึกไว้ในระบบและประกาศต่างๆที่ออกไปทุกครั้งย่อมเป็นข้อมูลที่ถูกต้อง สมบูรณ์และได้มาตรฐานเสมอ ถูกต้องเพราะเป็นข้อมูลล่าสุดที่บันทึกไว้ เพราะหากบุคลากรผู้รู้เรื่องนั้นต้องเป็นผู้รับและตอบทุกสายเกี่ยวกับเรื่องนั้นๆย่อมเสียเวลาต่อบุคลากรคนนั้น ต้องนั่งตอบคำตอบเดิมเป็นสิบหรือร้อยครั้ง บางครั้งต้องเสียเวลาไปเปิดข้อมูล หรือหากบุคคลท่านนั้นไม่อยู่ก็อาจไม่มีใครตอบได้หรือตอบได้แต่ไม่ครบถ้วน ไม่ถูกต้อง เป็นต้น

ข้อตกลงเบื้องต้น

การวิจัยเป็นโครงการนำร่อง ที่พยายามประยุกต์ใช้เทคโนโลยีที่มีอยู่ในมหาวิทยาลัย เพื่อประโยชน์การเรียนการสอน โดยเบื้องต้น ผู้วิจัยพยายามจะลดเวลา การเฉลยแบบฝึกหัด การประกาศผล และกระจายเสียงเทป

การบรรยายผ่านโครงข่ายคอมพิวเตอร์ ซึ่งถือเป็นที่สุดของโครงการวิจัยนี้ ส่วนการขยายผลไปสู่ระบบ ที่ใหญ่ขึ้นในเชิงบริหารต่างๆ จะต้องทำการศึกษาและ ออกแบบอย่างจริงจังในโอกาสต่อไป นอกจากนี้ การประกาศผลจะกระทำเฉพาะ สอบย่อย (QUIZ) และผลการสอบกลางภาค เท่านั้น ซึ่งอยู่ในขบวนการก่อนสอบปลายภาค จึงไม่ซ้ำซ้อนกับหน้าที่ของศูนย์บริการการศึกษา

ขอบเขตของการวิจัย

1. สามารถให้บริการข่าวสารแก่นักศึกษา และบุคลทั่วไปผ่านทางเครือข่ายโทรศัพท์
2. นักศึกษาสามารถขอทราบเกรดและผลสอบผ่านเครือข่ายโทรศัพท์ได้

ประโยชน์ที่ได้รับจากการวิจัย

1. มหาวิทยาลัยเทคโนโลยีสุรนารีไม่ต้องจ้างบุคลกรเพื่อรับโทรศัพท์ ติดต่อสอบถาม และทำหน้าที่งานทะเบียนเพิ่มขึ้น อีกทั้งประหยัดค่ากระดาษพิมพ์ข่าวสาร ข่าวประกาศต่าง ๆ ทำให้ประหยัดค่าใช้จ่าย
2. นักศึกษามหาวิทยาลัยเทคโนโลยีสุรนารีไม่ต้องเสียเวลา เสียค่าเดินทาง ค่าที่พัก เพื่อมาดูเกรดและผลสอบที่มหาวิทยาลัย เพียงแค่โทรศัพท์เข้ามาสอบถามก็สามารถทราบผลได้
3. บุคลากรของมหาวิทยาลัย ไม่ต้องคอยรับโทรศัพท์ และตอบคำถามซ้ำ ๆ กันเบาแรง ง่ายต่อการเปลี่ยนแปลงข้อมูล และในอนาคตจะไม่ต้องยุ่งยากในการแก้ไขการลงทะเบียนแต่ละรายวิชาของนักศึกษา

<u>หมายเหตุ</u> หัวข้อบริการแต่ละรายการผู้โทรสามารถเลือกรายการได้

บทที่ 2
การออกแบบเครื่องต้นแบบอิเล็กทรอนิกส์

ทฤษฎีและแนวคิดในการออกแบบ

ในสถาบันการศึกษาระดับอุดมศึกษา นิสิต นักศึกษาจะได้รับการปฏิบัติแบบผู้ใหญ่ ซึ่งนักศึกษาจะมีอิสระมากขึ้นในการเลือกลงรายวิชาเรียน หรือประกอบกิจกรรมต่างๆ แต่ในขณะเดียวกัน นักศึกษาต้องรับผิดชอบตัวเองมากขึ้นกว่าที่เคยในสมัยมัธยมศึกษา มหาวิทยาลัยจะสื่อสารกับบรรดานักศึกษาเป็นวงกว้างโดยใช้ใบปิดประกาศ ต่างจากการสื่อสารในระดับมัธยมศึกษา ซึ่งมักเป็นการสื่อสารผ่านทางอาจารย์ที่ปรึกษาเป็นหลัก ด้วยเหตุนี้นักศึกษาจึงต้องติดตามข่าวสารและกำหนดการสำหรับกิจกรรมต่างๆของทางมหาวิทยาลัยอย่างใกล้ชิด เพื่อที่ปฏิบัติตนให้ถูกต้องตามกฎระเบียบของทางมหาวิทยาลัย ในบรรดากิจกรรมต่างๆ นี้กิจกรรมที่สำคัญและจำเป็นที่สุดสำหรับนักศึกษาทุกคนได้แก่ กิจกรรมการลงทะเบียนและประเมินผล เพราะเป็นกิจกรรมบันทึกประวัติส่วนตัวทั่ว ๆ ไป ผลและประวัติการเรียนของนักศึกษาแต่ละคน โดยตรง

เทคโนโลยี Computer Telephony Integration (CTI) สามรถนำมาประยุกต์ใช้ในกิจกรรมดังกล่าวข้างต้น เพราะอุปกรณ์ในการทำสามารถใช้อุปกรณ์ที่มีอยู่แล้ว ตัวอย่างบริการ ประยุกต์อย่างง่าย ๆ อาจแบ่งออกเป็น 3 กลุ่มใหญ่

1. **Voice Mail**

2. **Interactive Voice Response System** ระบบที่ผู้ใช้สามารถสอบถามหรือเรียกข้อมูลที่ต้องการได้ด้วยการกดแป้นโทรศัพท์ เพื่อเลือกตัวเลือกแทนการกดแป้น keyboard เพื่อสอบถามข้อมูลบางอย่าง เช่น ข้อมูลเกี่ยวกับบัญชีในธนาคาร

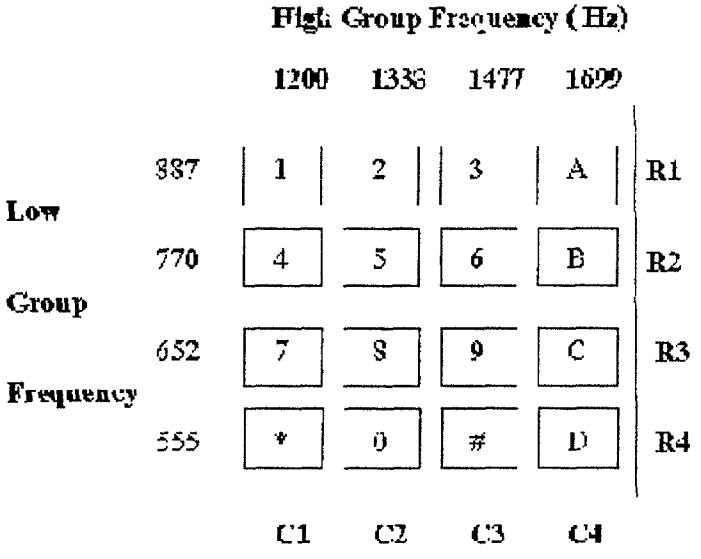
3. **Help Desk** ทำหน้าที่ตรวจหาเลขหมายโทรศัพท์ (บนระบบฐานข้อมูล) ที่ติดต่อเข้ามาและใช้เลขหมายนี้เพื่อค้นหาข้อมูลเพิ่มเติม เช่น หมายเลขโทรศัพท์สายตรง หมายเลขบัญชีธนาคาร หรือข้อมูลอื่น ๆ ที่ผู้โทรเข้ามาต้องการทราบ ในโครงการนี้จะใช้ในการบอกเกรดของนักศึกษาตามรหัสนักศึกษาที่ป้อนเข้ามา

ในการวิจัยนี้ เลือกใช้เทคนิค ระบบ IVR (Interactive Voice Response) เพราะข่าวสารข้อมูลของการเรียนการสอน ส่วนใหญ่จะเป็นข้อมูลประเภทตัวเลขหรือรหัสตัวอักษร เมื่อได้รับการจัดการข้อมูลอย่างเหมาะสมจะสามารถนำเอาระบบโทรศัพท์อัตโนมัติ หรือที่มีชื่อทางเทคนิคว่า ระบบ IVR (Interactive Voice Response) มาให้บริการแบบอัตโนมัติ ระบบโทรศัพท์อัตโนมัติ (IVR) เป็นเครื่องคอมพิวเตอร์ ซึ่งนอกจากความสามารถในการประมาณผลข้อมูลแล้ว ยังสามารถรับสาย เรียกสาย โอนสาย และพูดโทรศัพท์ด้วยข้อความอัตโนมัติที่บันทึกไว้ได้ ระบบโทรศัพท์แบบสายอัตโนมัติจึงเป็นระบบที่นำเอาความฉลาด(Intelligence) ของระบบคอมพิวเตอร์ผนวกเข้ากับความสามารถในการพูดและฟังในการสื่อสารผ่านข่ายโทรศัพท์ ซึ่งองค์ประกอบหลักของคอมพิวเตอร์ดังกล่าว จะขอเน้นที่อุปกรณ์แปลงสัญญาณ DTMF และ Voice Card

หลักการและขั้นตอนการทำงานของโครงข่ายโทรศัพท์

เมื่อคลื่นเสียงกระทบกับแผ่น Diaphragm จะทำให้แผ่น Diaphragm สั่นไปมา พลังงานเสียงก็จะเปลี่ยนเป็นพลังงานกล ในตำแหน่งที่แผ่น Diaphragm ถูกกดจะทำให้ Electrode แผ่นหน้าเคลื่อนที่เข้า เป็นผลทำให้ผงถ่าน (Carbon Granule) ถูกอัดติดกันมากยิ่งขึ้น การอัดตัวของผงถ่านนี้จะทำให้ความต้านทานระหว่างแผ่น Electrode ทั้งสองมีค่าลดลง ในทางตรงกันข้ามเมื่อแผ่น Diaphragm เคลื่อนที่ออกก็จะเป็นผลทำให้ Electrode แผ่นหน้าเคลื่อนที่ออกด้วย จึงทำให้ความต้านทานของ Transmitter เพิ่มขึ้น เมื่อเอาแบตเตอรีต่อเข้าระหว่างแผ่น Electrode ทั้งสองกระแสไฟตรงจะไหลผ่านผงถ่าน และเนื่องจากความต้านทานของ Transmitter มีการเปลี่ยนแปลงเมื่อได้รับสัญญาณเสียง ดังนั้นกระแสที่ไหลผ่าน Transmitter จะเปลี่ยนแปลงไปด้วยนั่นคือพลังงานเสียงสามารถเปลี่ยนเป็นพลังงานไฟฟ้าได้

เครื่องโทรศัพท์ที่มีหน้าปัดเป็นแบบกดปุ่มและใช้กรรมวิธีของ Dual Tone Multifrequency (DTMF) ในการส่งเลขหมายโทรศัพท์นั้น โดยทั่วไปหน้าปัดจะมี 12 ปุ่ม แบ่งเป็น 4 Rows และ 3 Columns และในเครื่องโทรศัพท์บางแบบอาจจะมีถึง 16 ปุ่ม โดยเพิ่ม Column ที่ 4 ขึ้นมาอีก ดังแสดงตามภาพที่ 1 ความถี่ที่ใช้ในแต่ละ Row และ Column จะมีความถี่ต่างกัน ความถี่ของทั้ง 4 Rows เรียกว่าเป็นกลุ่มความถี่ต่ำ (Low Group Frequency) และความถี่ของทั้ง 3 หรือ 4 Columns เรียกว่าเป็นกลุ่มความถี่สูง (High Group Frequency) การกดปุ่มที่เลขหมายใด ๆ จะทำให้วงจรอิเล็กทรอนิกส์ภายในเครื่องโทรศัพท์ผลิตความถี่ออกมา 2 ความถี่ เช่น เมื่อกดเลข 5 ความถี่ที่ผลิตออกมาคือ 770 Hz และ 1336 Hz เป็นต้น

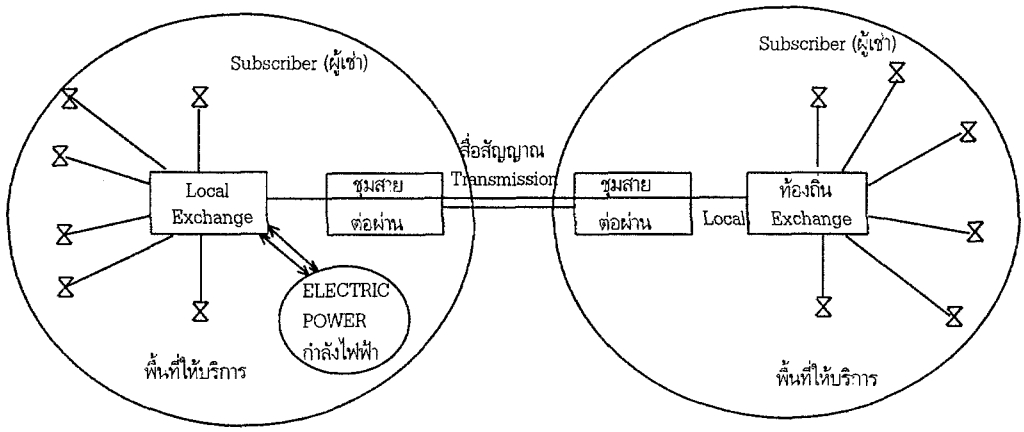


ภาพที่ 1 แสดงหน้าปัดของเครื่องโทรศัพท์แบบกดปุ่มและความถี่ที่ใช้

มาตรฐานของความถี่ที่ใช้และตำแหน่งของเลขหมายต่าง ๆ จะถูกจัดให้มีลักษณะดังแสดงตามภาพที่ 1 สำหรับความผิดพลาดที่ยอมให้เกิดขึ้นได้จะเป็น 1.5% สำหรับการผลิตความถี่ และ 2% สำหรับการรับเลขหมาย

องค์ประกอบของชุมสายโทรศัพท์

องค์ประกอบของชุมสายโทรศัพท์แบ่งออกเป็น 2 องค์ประกอบใหญ่ ๆ ชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ผู้เช่าต่อเข้าโดยตรง เช่น ชุมสายโทรศัพท์ท้องถิ่น (Local Exchange) และชุมสายที่ไม่มีเครื่องโทรศัพท์ต่อเข้าโดยตรง ซึ่งประกอบด้วย ชุมสายต่อผ่านท้องถิ่น (Tandem Exchange) และชุมสายโทรศัพท์ต่อผ่านทางไกล (Transit Exchange)



ภาพที่ 2 ชุมสายโทรศัพท์

- **ชุมสายโทรศัพท์ท้องถิ่น (Local Exchange)** หมายถึงชุมสายโทรศัพท์ที่มีเครื่องโทรศัพท์ของผู้เช่าต่อเข้าโดยตรง ชุมสายโทรศัพท์แบบนี้มีขนาดตั้งแต่เป็นร้อย ๆ เลขหมาย จนถึงหมื่นเลขหมายหรือมากกว่า

- **ชุมสายโทรศัพท์ต่อผ่าน** หมายถึงชุมสายโทรศัพท์ที่ไม่มีเลขหมายโทรศัพท์ของผู้เช่าต่อเข้ามาโดยตรง แต่จะบริการการเรียกระหว่างท้องถิ่นกับชุมสายท้องถิ่นด้วยกัน การเรียกระหว่างโทรศัพท์ 2 เลขหมาย อาจเรียกผ่านไปยังชุมสายต่อผ่านหลาย ๆ ชุมสายก็ได้

ตู้สาขา (PABX)

ตู้สาขา (PABX) เป็นชุมสายโทรศัพท์ที่มีลักษณะคล้ายกันกับชุมสายโทรศัพท์ท้องถิ่น แต่จะใช้ติดต่อกันภายในสำนักงาน โดยไม่ต้องผ่านชุมสายท้องถิ่น ตู้สาขาจะเป็นชุมสายโทรศัพท์ที่มีการบริการพิเศษ (Facility) แก่เลขหมายภายใน (Extension) ได้หลายอย่าง ซึ่งก็ขึ้นอยู่กับว่าตู้สาขานั้น ๆ มีขีดความสามารถเป็นอย่างไร การบริการพิเศษดังกล่าวได้แก่ การย่อเลขหมาย (Abbreviated Dialling) การเรียกกลับอัตโนมัติ

(Automatic Callback) การประชุมกันทางโทรศัพท์ (Conference Call) การโอนการเรียก (Transfer of Call) การโอนโทรศัพท์ติดตามตัว (Follow Me) ฯลฯ นอกจากนี้ในกรณีที่ตู้สาขาได้ทำการต่อเชื่อมกับชุมสายโทรศัพท์ท้องถิ่น ก็จะทำให้โทรศัพท์เลขหมายภายในสามารถติดต่อไปยังเลขหมายภายนอกได้ โดยผ่านชุมสายโทรศัพท์ท้องถิ่นและในทำนองเดียวกันโทรศัพท์จากเลขหมายภายนอกก็สามารถเรียกเข้าไปยังเลขหมายภายใน โดยผ่านตู้สาขาได้ตู้สาขาจะมีขนาดตั้งแต่ไม่ถึงสิบเลขหมายจนถึงหมื่นเลขหมายหรือมากกว่า

PABX กลายเป็นปัจจัยสำหรับสำนักงานที่ต้องการ มีชุมสายในสำนักงาน สาเหตุก็เพราะว่าภายในสำนักงานนั้นมีการใช้โทรศัพท์ภายในกันบ่อยมาก และมีความจำเป็นอย่างยิ่ง ถ้าหน่วยงานใดไม่มีการติดตั้ง PABX ก็จำเป็นที่จะต้องใช้โทรศัพท์พื้นฐาน (จากองค์การโทรศัพท์) ค่าใช้จ่ายในการโทรศัพท์จะสูงมาก และไม่มีการประหยัด ดังนั้นไม่ว่าสำนักงานเล็กหรือใหญ่ก็จะมีระบบชุมสายโทรศัพท์ปลายทางกันทั้งสิ้น คราวนี้สมมุติว่าเราเป็นเจ้าของอาคารสำนักงานต้องการที่จะติดตั้งชุมสาย PABX เพื่อให้บริการพนักงานของเรา เราจะเลือกพิจารณาอย่างไรบ้าง กล่าวเป็นข้อ ๆ ดังนี้คือ

1. ดูจำนวนพนักงานที่จำเป็นต้องใช้โทรศัพท์ว่ามีจำนวนเท่าใดและเผื่อไว้สัก 10 % เพื่อสำรองไว้ในอนาคต โดยไม่จำเป็นต้องเผื่อคนละ 2 เครื่อง เพราะการใช้โทรศัพท์ของ PABX ที่ดีนั้นเครื่องเดียวจะสามารถต่อได้ทุกอย่างทั้งภายในและภายนอก

2. กำหนดคู่สายโทรศัพท์พื้นฐานที่จะนำมาต่อเข้ากับ PABX ซึ่งโดยทั่วไปจะใช้อัตราส่วน 1:10 เช่น โทรศัพท์ภายในมี 100 หมายเลขจะใช้คู่สายภายนอก 10 คู่สาย เป็นต้น แต่ถ้าบริษัทใดเป็นบริษัทที่ต้องการให้บริการทางโทรศัพท์เป็นพิเศษ เช่น ขายตั๋วเครื่องบิน ก็จำเป็นต้องใช้อัตราส่วนที่ดีขึ้น เช่น ใช้อัตราส่วน 1:5 หรือ 1:6 เป็นต้น

3. เลือกชุมสายโทรศัพท์ชนิดที่มีประสิทธิภาพสูง เช่นระบบดิจิตอล แต่ต้องมีความง่ายต่อการใช้งานในการตั้งโปรแกรมเบอร์ต่าง ๆ หรือว่าโปรแกรมฟังก์ชั่นพิเศษจะต้องไม่ยุ่งยากถึงขนาดต้องเรียกผู้เชี่ยวชาญทุกครั้งไป ซึ่งตรงนี้เราเรียกว่า ทำคนและเครื่องจักรให้ใกล้ชิดกันมากขึ้นนั่นเอง

4. เลือกคุณสมบัติที่จำเป็นในการใช้งานมากที่สุดที่จะหาได้ ซึ่งฟังก์ชันการทำงานที่จำเป็นในสำนักงานที่สำคัญ ๆ มีดังนี้คือ

- ต้องมีระบบตอบรับอัตโนมัติเพื่อลดการทำงานของพนักงานโอนสายลง และสามารถเรียกไปยังเลขหมายปลายทางได้โดยตรง เราเรียกฟังก์ชั่นนี้เรียกว่า DID (Direct Inward Dialing)

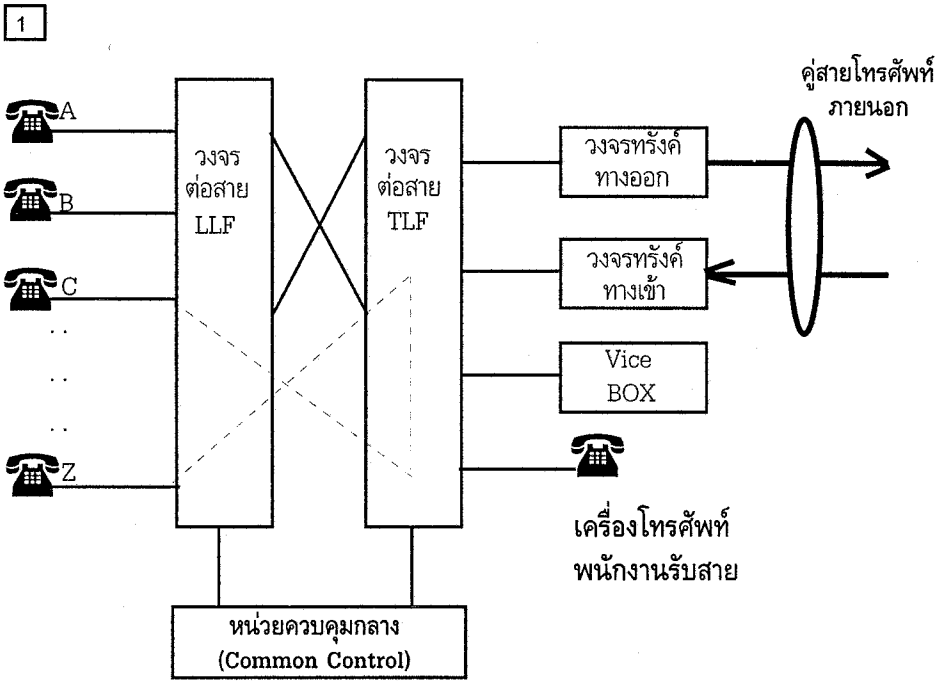
- ต้องมีระบบโอนสาย (Call Transfer) ที่สมบูรณ์ คือ สามารถโอนสายจากเครื่องใดก็ได้และไปยังเครื่องไหน ๆ ก็ได้ ทั้งนี้เพื่อลดเวลาและค่าใช้จ่ายในการโทรใหม่นั่นเอง

- ต้องมีระบบประชุมทางสาย (Conference Call) เพื่อสามารถสนทนากันได้หลายคนซึ่งจำเป็นในการประชุมย่อย ทำให้ประหยัดเวลาได้อย่างดี โดยทั่วไปจะมีระบบประชุมทางสายได้ตั้งแต่ 3-8 เครื่องในเวลาพร้อม ๆ กัน

- มีระบบเรียกใหม่ (Call Forwarding) ในกรณีที่ไม่มีผู้รับหรือไม่ตอบก็จะเรียกไปยังเบอร์ใหม่ที่อยู่ในกลุ่มเดียวกัน (ในห้องเดียวกัน) ระบบนี้จะมีประโยชน์มากในกรณีที่พนักงานบางคนลาหยุดไป จึงไม่มีคนรับสาย เครื่องชุมสายก็จะทำการเรียกไปยังโทรศัพท์อีกหมายเลขหนึ่งที่อยู่ใกล้เคียงกัน

- มีระบบคอยคิว (Call Queing) เมื่อมีการใช้โทรศัพท์กันมากคนที่โทรเข้ามาทีหลังจะได้ทราบว่าสามารถตามคิวได้ โดยไม่ต้องยกเลิกการติดต่อไปเสียก่อน

- ที่จำเป็นมากก็คือ จะต้องมีโทรศัพท์พิเศษสำหรับพนักงานรับสายไว้ทำการโอนสาย ซึ่งภาษาอังกฤษใช้คำว่า Attendant Console เพื่อให้ผู้เรียกสามารถสอบถามรายละเอียดหมายเลขภายในสำนักงานได้ บล็อกไดอะแกรมของชุมสายปลายทางอัตโนมัติหรือ PABX มีการทำงานดังภาพที่ 3



ภาพที่ 3 แสดงบล็อกไดอะแกรมของชุมสายโทรศัพท์ปลายทางที่ใช้กันในสำนักงานทั่ว ๆ ไป

หลักการพื้นฐานของทฤษฎีคิวอิงค์ (Queueing Theory)

เครื่องมือทางวิทยาศาสตร์ที่สามารถอธิบาย และวิเคราะห์ระบบสื่อสารข้อมูลที่ดีที่สุดในยุคปัจจุบันก็คือ Queueing Theory ทฤษฎีนี้ถูกนำไปใช้วิเคราะห์พฤติกรรมเชิงสถิติ (Statistical behavior) ของระบบโทรศัพท์อยู่บ่อยครั้งในวิชา Data communication นี้ จะนำเสนอเฉพาะทฤษฎีพื้นฐานที่จำเป็น ในการ วิเคราะห์ระบบอย่างง่าย ๆ โดยทั่วไป เราจะถือทักเอาว่า เวลาที่ Server ใช้ในการให้บริการเป็น การกระจายแบบ เอ็กโปรเนเชียล (Exponential Distribution) เพราะจะเห็นว่าในชีวิตความเป็นจริง เวลาที่ที่ใช้ในการบริการ จะ ไม่มีทางเป็นแบบเชิงเส้น เพราะ

ยิ่งผู้ใช้บริการมาก ขึ้นเรื่อย ๆ เวลาที่ให้บริการก็จะนานขึ้น เป็นเงาตามตัว เฉกเช่น เวลาที่เราไปกินอาหารที่ภัตตาคารที่แน่นไปด้วยลูกค้า ถ้ายิ่งลูกค้าเข้ามามากเท่าไร เวลารออาหารก็ยิ่งนานขึ้น

การกระจายแบบเอ็กโปรเนเชียล (Exponential Distribution)

$$F(x) = 1 - e^{-\mu x} \qquad \text{--------(1)}$$

หาค่า density function ได้จาก

$$f(x) = \{ \ d(F(x)) / dx \ \}$$

จะได้

$$f(x) = \mu e^{-\mu x}$$

หรือ

$$f_x(t) = \begin{cases} \mu e^{-\mu t}, t > 0 \\ 0, \text{อื่น ๆ} \end{cases} \qquad \text{---------(2)}$$

Y มีค่าฟังก์ชั่นที่น่าจะเป็นไปได้ดังนี้ เมื่อ Y คือค่าแปรเปลี่ยนของเวลาบริการ และอัตราบริการแสดงโดย

$\mu = (E(Y))^{-1}$

โอกาสในการเข้ามาของลูกค้า ก็ไม่ใช่การกระจายแบบเชิงเส้น เพราะในชีวิตจริง เราคงกะเกณฑ์ไม่ได้ว่าให้ลูกค้าเข้ามาทุก ๆ 5 นาที แต่การแจกแจงหรือการกระจายจะเป็นแบบ Poisson Distribution กล่าวคือ โอกาสในการเข้ามาของลูกค้า จะขึ้นอยู่กับอัตราการเข้าร้านของลูกค้า โอกาสในการเข้ามาของลูกค้า จะมีค่าเพิ่มขึ้นจนถึงระดับหนึ่ง แล้วถ้าอัตราการเข้าของลูกค้ายังคงเพิ่มขึ้น โอกาสในการเข้ามาของลูกค้า จะเริ่มลดลงในทุก ๆ อัตราดังกล่าวสูงขึ้น

$$Pr[X = k] = (\lambda^k e^{-\lambda}) / k!$$

$$E[X] = \sigma x = \lambda$$

สำหรับช่วงเวลาที่ลูกค้ามาถึงระบบคิวนั้นเราเรียกว่าช่วงเวลาที่มาถึง (Interarrival Time) เวลาที่ผู้บริการใช้ในการบริการลูกค้าเราเรียกว่าเวลาบริการ (Service Time) เนื่องจากทั้งสองคู่แน่นอนเราจะแสดงค่าทั้งสองในรูปของฟังก์ชั่นของความน่าจะเป็นไปได้ (Probability De Function) ตัวอย่างเช่น

ถ้าให้ช่วงเวลาที่มาถึง x นั้นมีค่าฟังก์ชั่นที่น่าจะเป็นไปได้ดังนี้

$$f_x(t) = \begin{cases} \gamma e^{-\lambda t}, t > 0 \\ 0, \text{อื่น ๆ} \end{cases}$$

โดยที่ค่า x เป็นค่าแปรเปลี่ยนของช่วงเวลาที่มาถึง และ t คือเวลา λ คืออัตราการมาถึงมีค่าเท่ากับ $(E(x))^{-1}$

ถ้าเรากำหนดให้จำนวนลูกค้าโดยเฉลี่ยที่อยู่ในระบบเท่ากับ N ในระบบประกอบด้วยคิวแถวเดียวและผู้บริการคนเดียว เมื่อ T เป็นค่าเฉลี่ยที่ลูกค้าคนหนึ่งใช้ในระบบ และถ้าให้จำนวนเฉลี่ยของลูกค้าที่รออยู่ในคิวนั้นเท่ากับ N_q โดยที่ W เป็นเวลาเฉลี่ยที่ลูกค้าคนหนึ่งใช้ในคิว จะเห็นได้ว่า $N \geq N_q$ และ $T \geq W$ แสดงผลที่ง่ายที่สุดกฎหนึ่งในระบบคิวอิงค์โดยจะอธิบายถึงความสัมพันธ์ระหว่าง N, T และ λ ดังนี้

$$N = \lambda T$$

ซึ่งกล่าวว่า ค่าเฉลี่ยของลูกค้าในระบบจะเท่ากับ ผลคูณของอัตราเฉลี่ยที่ลูกค้าเข้าระบบกับค่าเฉลี่ยที่ลูกค้าใช้ในระบบ

นอกจากนี้กฎนี้ยังแสดงถึง

$$N_q = \lambda W$$

$$T = W + 1/\mu$$

$$N = N_q + \lambda / \mu$$

ตัวแปรที่สำคัญในระบบคิวอิงค์ก็คือ ช่วงเวลาการมาถึง เวลาบริการและจำนวนผู้บริการ ดังนั้นจึงมีสัญลักษณ์ A/B/M ใช้ในระบบคิวอิงค์ โดยที่ A แสดงถึงการกระจายที่เป็นไปได้ของช่วงเวลาการมาถึง B แสดงถึงการกระจายของเวลาบริการและ M คือจำนวนผู้บริการ ตารางที่ 1 แสดงถึงการกระจายบางค่าของ A และ B ตัวอย่าง สัญลักษณ์ M/M/1 แสดงถึงระบบคิวอิงค์ที่มีผู้บริการหนึ่งคน เวลาบริการและช่วงเวลาการมาถึงเป็นการกระจายแบบเอกซ์โปเนนเทียล

ตารางที่ 1 การกระจายของช่วงเวลาการมาถึงและเวลาบริการสำหรับระบบคิวอิงค์ A/B/M

สัญลักษณ์	ชื่อ	คำอธิบาย
M	Markovain	การกระจายเป็นแบบเอคโปเนนเทียล
G	General	การกระจายเป็นแบบทั่วๆ ไป ไม่เจาะจง
D	Deterministic	ค่าคงที่

วงจรแปลงสัญญาณ

การทำงานของวงจรแปลงสัญญาณที่ใช้ในโครงการวิจัยนี้เป็นระบบที่ใช้งานผ่านระบบโทรศัพท์ ซึ่งรหัสที่ใช้ติดต่อระหว่างผู้โทรกับคอมพิวเตอร์เป็นสัญญาณความถี่ของสองความถี่ผสมกันเป็นความถี่เดียว ซึ่งมีชื่อทางเทคนิคว่า DTMF (DialTone Multi Frequency) และระบบเสียงพูด ซึ่งสำหรับคอมพิวเตอร์นั้น จะนำ DTMF ไปทำการประมวลผลและแสดงออกมาเป็นเสียง เสียงดังกล่าวจะเก็บอยู่ในรูปของ File ซึ่งอัดผ่านไมโครโฟนหรือด้วยระบบอื่น เพราะฉะนั้นคอมพิวเตอร์จึงต้องมีองค์ประกอบเพิ่มขึ้นในส่วนการรับ DTMF

และ Play File ดังกล่าว ดังนั้นวงจรแปลงสัญญาณจะทำหน้าที่ในการ Detect สัญญาณเสียงกระดิ่ง (Ringing) เมื่อมีการโทรเข้า และจะทำการยกหู (OFFHOOK) รับสายโทรศัพท์ นอกจากนี้ยังทำหน้าที่ในการถอดรหัสสัญญาณ DTMF เมื่อผู้โทรทำการเลือกรายการจากแป้นกด (Keypad) โทรศัพท์ และทำการ play voice file อีกด้วย

เนื่องจากระบบจะต้องมีการเปลี่ยนแปลงข้อมูลอยู่เสมอ ซึ่งอาจจะทำได้โดยการอัดเสียงผ่านทางไมโครโฟน หรือโดยการโทรศัพท์เข้ามาแล้วกดรหัสผ่าน (password) เพื่อทำการเปลี่ยนแปลงข้อมูล นอกจากนี้คอมพิวเตอร์จะต้องทำการสนทนาด้วยเสียงกับผู้ที่โทรเข้าได้ และต้องทำการโทรออกได้ด้วย ดังนั้นระบบจะต้องมีองค์ประกอบที่สำคัญอีกประการซึ่งทำหน้าที่ดังกล่าวคือ Sound card นั่นเอง

Sound Card

การ์ดเสียงทำหน้าที่ในการบันทึกและเล่นเสียง โดยพื้นฐานแล้ว การ์ดเสียงจะแตกต่างกันไปตามวิธีการที่สร้างเสียงขึ้นมา คุณภาพความละเอียดที่ใช้ และฟังก์ชันพิเศษที่เพิ่มมาบนการ์ด อย่างเช่น วงจรเสียงแบบ pass-through หรือซีดีรอมอินเตอร์เฟส

ในระบบ CT ที่ใช้ในโครงการนี้ต้องการความรวดเร็วในการทำงาน และมีประสิทธิภาพที่ดี ดังนั้นการเลือกการ์ดก็จำเป็นอย่างหนึ่ง เพราะโครงงานวิจัยและพัฒนานี้เป็นการติดต่อโดยใช้การบันทึกและเล่นเสียง จาก Voice File เป็นส่วนใหญ่ ทั้งยังติดต่อกับระบบเครือข่ายโทรศัพท์อีก ดังนั้นถ้าจะให้ดีจำเป็นต้องใช้การ์ดที่มีความสามารถในการเล่นเสียงและบันทึกเสียงได้อย่างมีประสิทธิภาพ เช่น การ์ดเสียงแบบ 16 บิต และระบบ Sound Blaster 16 เป็นตัวอย่างของการ์ดเสียงแบบ 16 บิต ที่ดีรุ่นหนึ่ง ที่เหมาะที่จะใช้กับระบบ CT (Computer Telephony)

ระบบโทรศัพท์

โครงสร้างเน็ตเวิร์กโทรศัพท์

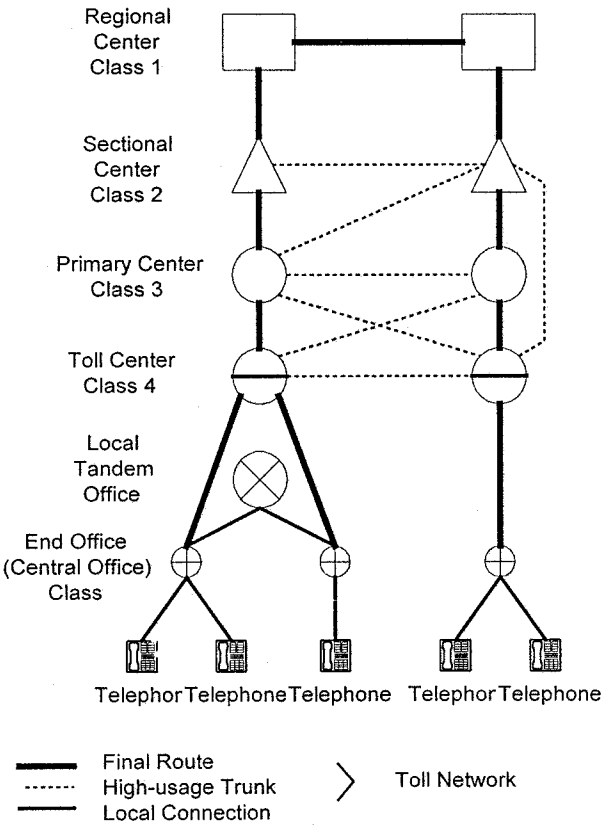
ระบบโทรศัพท์เป็นการสื่อสารชนิดหนึ่ง ระหว่างจุดต่อจุด เป็นการสื่อสารสองทางด้วยเสียงพูด และข้อมูล การติดต่อระหว่างโทรศัพท์ถึงโทรศัพท์ เกิดขึ้นที่สำนักงานกลางท้องถิ่นจะเป็นโทรศัพท์ภายในสำนักงานเดียวกัน จนถึงทรั้งไลน์(TRUNK LINE) ทำหน้าที่สวิทชิ่งในปัจจุบัน เพราะมีข้อดีหลายด้าน ซึ่งเป็นความได้เปรียบของสวิทชิ่งอิเลคทรอนิคที่ได้รับการเลือกใช้ในปัจจุบัน

ระบบโทรศัพท์ เริ่มจากลูปท้องถิ่น (LOCAL LOOP) ซึ่งประกอบด้วยลวด 2 เส้น ที่ต่อกับสำนักงานกลาง (CENTRAL OFFICE) ไปยังผู้ใช้โทรศัพท์ สำนักงานกลาง 1 แห่ง อาจมีชุมสายโทรศัพท์ (EXCHANGE) หลายแห่ง ซึ่งจะแทนด้วยหมายเลขโทรศัพท์ 3 ตัวแรก(จากทั้งหมด 7 ตัว) ซ้ำกันและสามารถรองรับโทรศัพท์ได้กี่เลขหมายขึ้นกับขนาดของชุมสาย เช่นจาก 0000 ถึง 9999 เป็นต้น

หน้าที่ของสำนักงานกลางเพื่อควบคุมลูปท้องถิ่น ให้โทรศัพท์ ได้เชื่อมต่อกันสำเร็จ สำนักงานกลางยังเฝ้าฟังสายโทรศัพท์ที่ต่อได้แล้ว ว่าได้วางหู (ON_HOOK) หรืออยู่ในสภาวะใช้งาน (OFFHOOK) เพื่อส่งโทรเรียกสัญญาณ กระดิ่งและโทนอื่นๆ ในระบบโทรศัพท์

หน้าที่เชื่อมโยงโทรศัพท์ของสำนักงานกลาง ภายในระบบทำหน้าที่ให้ภายในชุมสายหนึ่งเท่านั้น ถ้าการเรียกจำเป็นต้องไปอีกชุมสายหนึ่งสำนักงานกลางก็จะทำหน้าที่เชื่อมต่อ TRUNK เข้าด้วยกัน

โครงสร้างของเครือข่ายโทรศัพท์สามารถแสดงได้ดังบล็อกไดอะแกรมต่อไปนี้



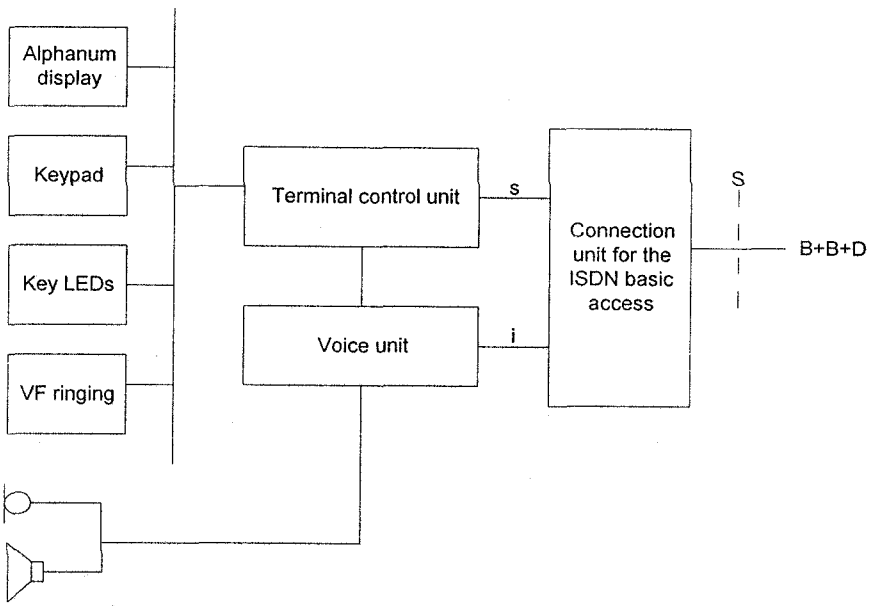
ภาพที่ 4 แสดงโครงสร้างเน็ตเวิร์คโทรศัพท์สาธารณะ

เครื่องโทรศัพท์

เป็นอุปกรณ์ที่ใช้สื่อสารจากจุดหนึ่งไปยังอีกจุดหนึ่ง และเป็นอุปกรณ์ปลายทางที่ให้บริการข่าวสารเพียงอย่างเดียว ซึ่งโครงสร้างพื้นฐานของโทรศัพท์จะแตกต่างกันไปตามบริษัทผู้ผลิต แต่จะมีส่วนประกอบที่สำคัญ คือ ALPHANUMERIC DISPLAY ซึ่งสามารถใช้แสดงเลขหมายเพื่อข่าวสารอื่นๆ ได้ ซึ่งในปัจจุบันการพัฒนาระบบเครือข่ายโทรศัพท์ในประเทศไทย จากเดิมที่เคยใช้ระบบอะนาลอก ซึ่งมีข้อกำหนดในการสื่อสาร

หลายด้าน และได้มีการพัฒนาเป็นการใช้ร่วมระบบดิจิตอล (ISDN) ทำให้สามารถพัฒนาใช้อุปกรณ์ปลายทางในการติดต่อสื่อสารได้หลายอย่างที่แตกต่างกันเช่น โทรศัพท์ Interactive data Communication, Electronic Mail, Slow scan TV, Videoconferencing รวมทั้งการพัฒนาการใช้คอมพิวเตอร์ให้สามารถใช้ร่วมกับระบบเครือข่ายโทรศัพท์ได้ด้วย

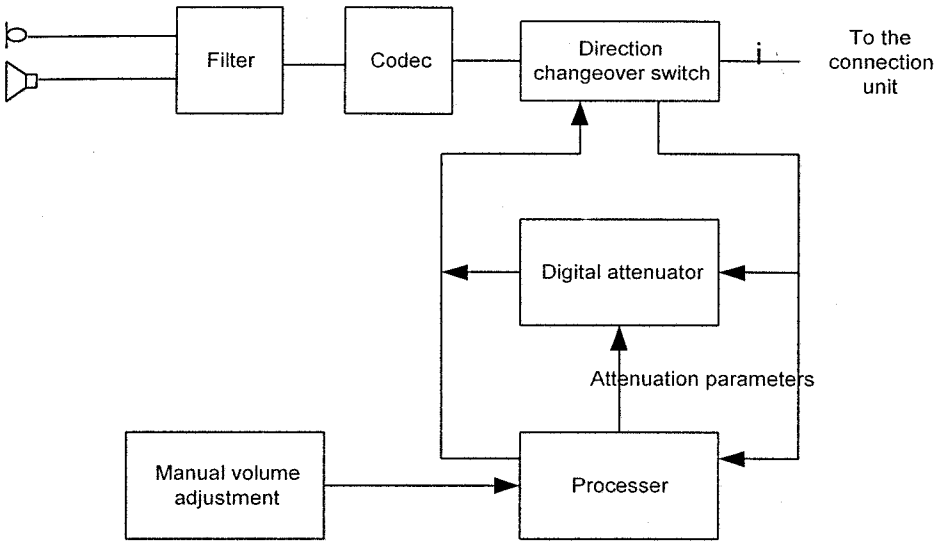
ด้วยทั่วไปเครื่องโทรศัพท์จะมีส่วนประกอบดังแสดงในภาพที่ 5



ภาพที่ 5 แสดงโครงสร้างพื้นฐานของโทรศัพท์ที่ใช้ในเน็ตเวิร์คร่วมดิจิตอล

- Keypad ประกอบด้วยปุ่มกดสำหรับการทำงานต่างๆ เพื่อสร้างสิ่งที่ต้องการและคีย์ที่ทำหน้าที่เพิ่มเติมในการให้บริการของโทรศัพท์สำหรับการบริการร่วมดิจิตอล
- Keyleds แสดงผลการใช้คีย์ต่างๆ
- VF RINGIG ทำหน้าที่แปลงสัญญาณเสียงเมื่อมีการเรียกเข้า
- VOICE UNIT ทำการแปลงสัญญาณเสียงจากอนาลอกให้เป็นดิจิตอล
- TERMINAL CONTROL หน้าที่ในการสแกนดูการทำงานของคีย์ต่างๆ และควบคุมส่วนของ ALPHANUMERIC DISPLAY ของระบบโทรศัพท์

สำหรับโทรศัพท์ที่ไม่มี HANDSET เรียกว่า HANDFREE TALKING จะมีองค์ประกอบต่างๆ ดังที่แสดง ด้วยบล็อกไดอะแกรมดังต่อไปนี้



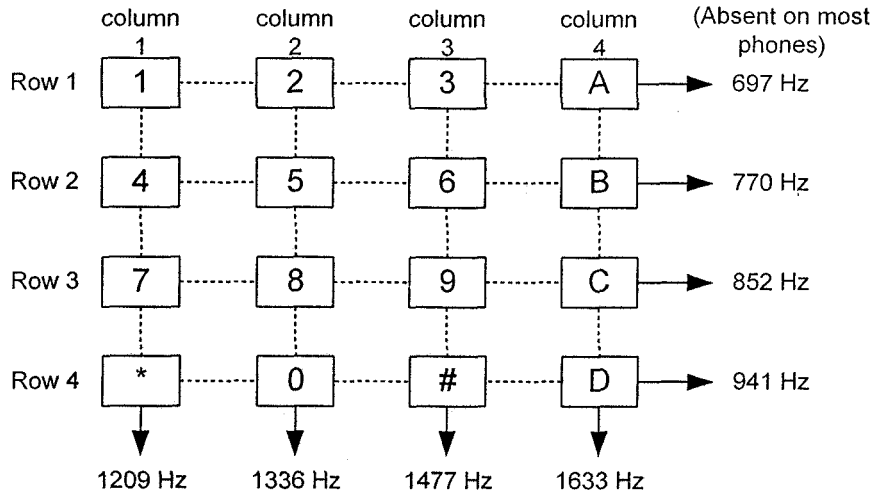
ภาพที่ 6 แสดงอุปกรณ์ HANDSFREE กับส่วนควบคุมสัญญาณเสียงดิจิตอล

- DIGITAL ATTANUATION ใช้สำหรับแก้ปัญหาการลดทอนเนื่องจากการเข้ารหัสดิจิตอลและการถอดรหัสดิจิตอลของเสียงพูดแบบ PCM โดยการเข้ารหัสและถอดรหัสเป็น CODE WORDS ซึ่งมาจากรหัสเริ่มแรกที่ถูกต้องกับค่าความคลาดเคลื่อนที่เกิดขึ้น จากการเข้ารหัสนั้น ขบวนการนี้มีความจำเป็นในการกำหนดค่าความผิดพลาดที่เกิดขึ้น โดยการเปรียบเทียบกับระดับก่อนในทิศทางการส่ง

เมื่อยกหูโทรศัพท์ขึ้นทำให้วงจรเปิดไม่มีกระแสไหล วงจรท้องถิ่น สมบูรณ์ครบกระแส 20 มิลลิแอมป์ไหลด้วยแรงเคลื่อน –48 โวลท์ จากแบตเตอร์ที่สำนักงานกลาง

ความแรงของระดับเสียงอยู่ในระดับ –10 ถึง 0 dBm มีความต้านทาน 600 โอห์ม (บางแห่งใช้ 900 โอห์ม) สัญญาณที่ผ่านหลายๆ ขั้นตอน เช่น เชื่อมโยงทรั้ง, สวิทต่างๆ จะมีระดับ –42 ถึง –4 dBm ความถี่ราว 3995 Hz และระดับ –16 ถึง –36 dBm หรือต่ำกว่าขั้นความถี่ แรงต้านจะอยู่ระหว่าง 200 ถึง 1,200 โอห์ม

โทนเรียกที่กดไปจะอยู่ในระหว่างความถี่ 770 ถึง 1,336 เรียกว่า ดีทีเอ็มเอฟ (DIUAL TONE MUTIFREQUENCY :DTMF) ซึ่งสามารถแสดงได้ดังภาพต่อไปนี้



ภาพที่ 7 แสดงโครงสร้างความถี่ DTMF ของโทรศัพท์

ตารางที่ 2 แสดงโครงสร้างความถี่ DTMF ของโทรศัพท์

Division factors for MC14403 DTMF generator IC (Courtesy of Motorola Inc.)

* f_{osc} = 3,579,545 Hz ; f_{actual} = OSC/N

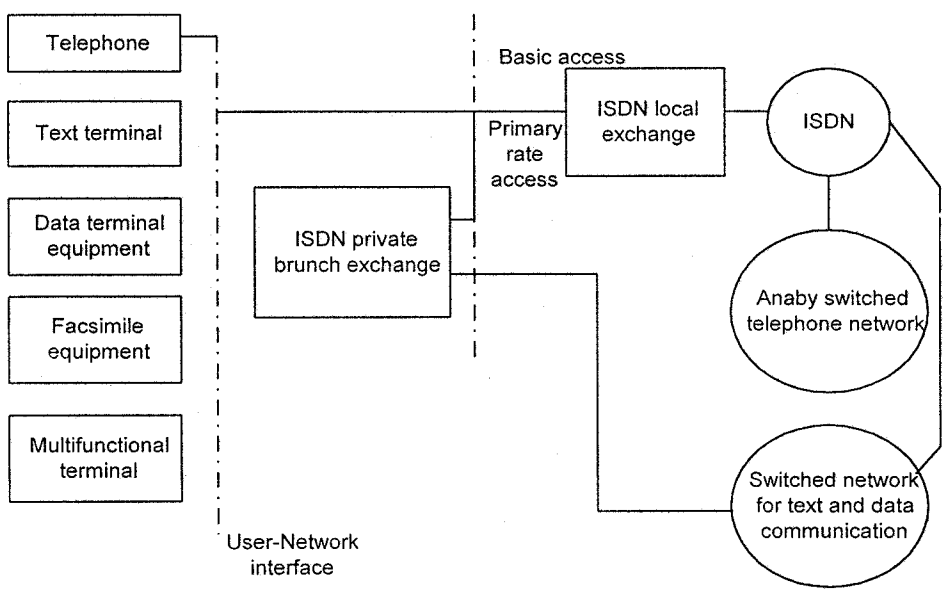
Input	Nominal (Hz)	Divider ratio (N)	Actual (Hz)	Deviation (%)
R1	697	5120	699.13	+0.3
R2	770	4672	766.17	-0.5
R3	852	4224	847.43	-0.5
R4	941	3776	947.97	+0.74
C1	1209	2944	1215.88	+0.57
C2	1336	2688	1331.68	-0.32
C3	1477	2432	1471.85	-0.35
C4	1633	2176	1645.01	+0.74

ปัจจุบัน การพัฒนาวงจรโทรศัพท์สามารถสร้างเป็นแบบ อิทิเกรดเทดเซอร์กิต (INTEGRATED CIRCUIT) ที่มีชื่อเรียกว่า CHIP ซึ่งสามารถรวมเอาระบบวงจรของเครื่องโทรศัพท์ใช้ใน IC ตัวเดียว และสามารถทำงานได้เท่ากับวงจรหลายๆ วงจรได้โดยใช้แรงดันเพียง 1.4 โวลท์

การอินเตอร์เฟส กับระบบเครือข่ายโทรศัพท์

ปัจจุบันการสื่อสารผ่านเครือข่ายโทรศัพท์ได้ก้าวหน้าไปมาก เพื่อเตรียมพร้อมสำหรับการสื่อสารได้หลายระบบข่าวสารทั้งการบริการที่ใช้เสียง และไม่ใช้เสียงที่ผู้ใช้บริการจะสามารถขอทำการแอกเซส (Access) กับระบบที่มีอยู่ในขอบเขตที่กำหนดการอินเตอร์เฟสกับเน็ตเวิร์คมาตรฐานของ CCITT ซึ่งระบบที่มีความสามารถดังกล่าว เรียกว่า ISDN (INTEGRATED SERVICE DIGITAL NETWORK) ที่นิยามโดย CCITT

เน็ตเวิร์ค ISDN สามารถใช้บริการเกี่ยวกับข้อมูลและโทรคมนาคมอื่นๆ ที่เป็นเน็ตเวิร์ค ดิจิตอล โดยอุปกรณ์ต่าง ๆ จะอยู่ในรูปแบบดิจิตอล เช่น ข่าวสารโทรศัพท์ คอมพิวเตอร์ PBX และอุปกรณ์อื่น ๆ โดยอุปกรณ์ข่าวสารทั้งหมดจะถูกสวิตช์และส่งผ่านเป็นกลุ่มบิทในเนตเวิร์คเดียวกันดังภาพแสดงต่อไปนี้



ภาพที่ 8 แสดงโครงสร้างพื้นฐานของอุปกรณ์ปลายทางที่ใช้ในเนตเวิร์คร่วมดิจิตอล

การเชื่อมต่อคอมพิวเตอร์เข้ากับระบบเครือข่ายโทรศัพท์เป็นการเชื่อมต่อเอาเทคโนโลยีด้านคอมพิวเตอร์และเครือข่ายโทรศัพท์เข้าด้วยกัน ซึ่งเรียกว่า COMPUTER TELEPHONY (CT)

ในความเป็นจริง CT เป็นการนำเอาเทคโนโลยีคอมพิวเตอร์ด้านความเร็ว (Speed) กำลัง (Power) ความสะดวกสบาย (Convenience) และความสามารถอื่น ๆ ของ PC มาใช้ร่วมกันกับระบบความสามารถของระบบโทรศัพท์ เพื่อที่จะทำให้เกิดประโยชน์ด้านการใช้งานแก่คนเราและอุปกรณ์การใช้งานด้วย

ความต้องการของระบบ CT : Computer Telephony

ระบบ CT นอกจากจะต้องประกอบด้วย PC และ Telephone line แล้ว CT ยังต้องการอุปกรณ์พิเศษที่สามารถทำการติดต่อสื่อสาร เช่น Hardware และ Software เพื่อทำการเชื่อมต่อระบบคอมพิวเตอร์เข้ากับระบบเครือข่ายโทรศัพท์ ให้สามารถทำงานร่วมกันในหลายๆ ด้านได้ ซึ่งอุปกรณ์เหล่านั้นประกอบด้วย

- แผงวงจรเชื่อมต่อระหว่าง PC กับโทรศัพท์ โดยทั่วไปอุปกรณ์นี้จะเป็น Modem หรือแผงวงจรก็ได้อุปกรณ์ที่ทำหน้าที่นี้อาจจะสามารถเชื่อมต่อกับ line โทรศัพท์ ได้โดยตรงหรือไม่ก็ได้ โดยอาจจะเป็นพอร์ตให้เสียบ line โทรศัพท์ บน PC เมื่อทำการเพิ่มแผงวงจรใน PC แล้ว และ Modem หรือแผงวงจรนี้จะต้องมี Multifunction พิเศษ ที่รวมอยู่ในตัวมันเองเช่น Sound Card, Fax, Modem และ ตัวควบคุม CD-ROM นอกจากนี้จะต้องมีฟังก์ชันพิเศษในการ interface line โทรศัพท์ เพื่อให้ PC หนึ่งเครื่องสามารถรองรับ Line โทรศัพท์ได้หลายๆ เบอร์
- Sound Card เพื่อที่จะเพิ่มความสามารถให้กับคอมพิวเตอร์ ให้สามารถโทรออกได้เหมือนกับโทรศัพท์ทั่วไป ซึ่ง Sound Card นี้เป็นตัว Common phrase ทำให้ผู้ใช้ได้ยินเสียงพูดเมื่อทำการติดต่อกับผู้โทรออก หรือเอาไว้ทดสอบโปรแกรม เมื่อไม่ต้องการโทรออก
- Software เพื่อทำหน้าที่ควบคุมการทำงานของ PC และ Modem ให้ทำการ Link Up ระหว่าง PC กับเครือข่ายโทรศัพท์ได้ รวมทั้งสามารถทำการโทรออก รับ Voice mail และ Call logging System ได้

ผลการออกแบบ

ระบบ CTI เป็นการนำเอาเทคโนโลยีด้านคอมพิวเตอร์ มาใช้ร่วมกันกับเทคโนโลยีการสื่อสารผ่านเครือข่ายโทรศัพท์ ซึ่งต้องมีอุปกรณ์พิเศษ หรือฟังก์ชันพิเศษ เพื่อทำหน้าที่ในการ Interface ระหว่างสองระบบเข้าด้วยกัน ซึ่งฟังก์ชันที่ว่านี้อาจจะเป็น Modem ที่อยู่นอกเครื่อง PC หรืออาจจะเป็นแผงวงจรที่สามารถประกอบเข้าไปในเครื่องคอมพิวเตอร์เลยก็ได้ ซึ่งในโครงงานวิจัยและพัฒนานี้ใช้แผงวงจรของบริษัท Dialogic รุ่น D41/D&D21/D ซึ่งสามารถรองรับคู่สายโทรศัพท์ได้ 4 คู่สายต่อหนึ่งแผงวงจร นอกจาก Hardware ที่ทำหน้าที่ในการ Interface แล้วยังจะต้องมี Software ที่สำหรับควบคุมการทำงานของ PC กับ การ์ดดังกล่าว รวมทั้งต้องมี Sound card เพื่อทำหน้าที่ในการบันทึกเสียงและเล่นเสียง เมื่อเราต้องการให้ PC ทำการสนทนาได้ตามปกติเหมือนกับการสนทนาจากเครื่องโทรศัพท์ถึงเครื่องโทรศัพท์โดยทั่วไป และทำให้ผู้ใช้สามารถได้ยินเสียงสนทนานั้นผ่านทาง Sound Card ได้ด้วย

บทที่ 3
วิธีดำเนินการวิจัย พัฒนาและปรับแต่ง

วิธีดำเนินการวิจัย และแผนงานวิจัย

ศึกษารายละเอียดที่เกี่ยวข้องกับโครงการ (Project Study and Analysis) กล่าวคือ ค้นคว้าเอกสารและศึกษาการทำงานของการสื่อสารผ่านเครือข่ายโทรศัพท์แบบอัตโนมัติซึ่งเป็นขั้นตอนของการพัฒนาการเชื่อมต่อระบบเครือข่ายข้อมูลคอมพิวเตอร์เข้ากับระบบสื่อสารผ่านเครือข่ายโทรศัพท์ โดยมีการทำงานเป็นแบบอัตโนมัติ โดยที่คอมพิวเตอร์เป็นตัวควบคุมการทำงานทั้งหมด

- ศึกษาวิชาการทางโทรศัพท์ (Telephony) และอุปกรณ์อิเล็กทรอนิกส์ ที่จะใช้ในระบบ ได้แก่

ก. วงจรเชื่อมต่อสัญญาณจาก DTMF (Dial Tone Multifrequency) เป็น Digital ซึ่งเป็นวงจรเชื่อมต่อระหว่างระบบการสื่อสารผ่านเครือข่ายโทรศัพท์

ข. หลักการการทำงานของระบบโทรศัพท์แบบอัตโนมัติ หรือที่มีชื่อทางเทคนิคว่าระบบ IVR (Interactive Voice Response) ซึ่งในโครงการนี้จะเป็นการพัฒนาโดยการนำเทคโนโลยีด้านระบบโทรศัพท์แบบอัตโนมัติมาใช้ร่วมกับระบบเพื่อให้ได้ตามวัตถุประสงค์ของโครงการ

ค. ศึกษาภาษาที่ใช้ในการเขียนโปรแกรม และออกแบบโครงสร้าง ขอบเขต เงื่อนไข และเขียน Software เพื่อควบคุมการทำงานของระบบ

- วิเคราะห์ความต้องการของระบบ และจัดทำข้อกำหนดอันเป็นเป้าหมายของระบบทั้งหมด
- วิเคราะห์และเปรียบเทียบเทคโนโลยีที่ได้ศึกษามา ที่ได้กล่าวมาข้างต้น ซึ่งบางตัวมีขายในตลาดอยู่แล้ว และมีราคาต่อคู่สายสูงมาก
- เลือกวิธีที่จะใช้ในการพัฒนาระบบ
- พัฒนาการสั่งงาน Hardwarc ให้มีการตอบรับการโทรเข้าโดยอัตโนมัติ และมีการกำหนดจำนวนของการพยายามโทรเข้า
- พัฒนาการสั่งงาน การทำรายการเลือกสำหรับผู้โทรเข้า เพื่อเลือกรายการที่จะขอใช้
- การพัฒนาการเชื่อมต่อกับเครือข่ายข้อมูลคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์สามารถเชื่อมต่อกับระบบเครือข่ายข้อมูลคอมพิวเตอร์ได้
- การทดสอบการทำงานของระบบ จะทำการทดสอบการทำงานตามขั้นตอนตามที่กล่าวมาข้างต้น

ความต้องการของระบบ

ระบบจำเป็นต้องมีการทำงานที่สามารถให้บริการงานในหลายๆ ด้านที่แตกต่างกันได้ ระบบจึงต้องมีการเชื่อมต่อข้อมูล และมีการเปลี่ยนแปลงข้อมูลที่เป็น Voice File เป็นต้น ดังนั้นระบบจะต้องทำงานดังต่อไปนี้

- ปฏิบัติการเปลี่ยนเสียงพูด สัญญาณเสียง การอัดข้อมูลเสียง และเปลี่ยนแปลงข้อมูลที่เป็น Voice file

- มีประสิทธิภาพในการส่งข้อมูลอย่างต่อเนื่อง ในขณะที่เกิดความหนาแน่นของข่าวสาร
- การเรียกอุปกรณ์ปลายทางและการทำงานอื่นซึ่งเป็นไปได้อย่างรวดเร็ว
- ให้ความผิดพลาดของข้อมูลน้อยที่สุด
- ระดับการสื่อสารมีความปลอดภัยสูงสุด

องค์ประกอบของระบบ

Hardware Component

1. เครื่องคอมพิวเตอร์ PC เพื่อทำหน้าที่ประมวลผล
2. แผงวงจร Voice Board ทำหน้าที่แปลงเสียงให้เป็นข้อมูลดิจิตอล และแปลงข้อมูลดิจิตอลจากแฟ้มข้อความกลับเป็นเสียงพูด
3. แผงวงจรเชื่อมต่อเพื่อรับ-ส่งข้อมูล สำหรับเชื่อมเข้ากับเครือข่ายคอมพิวเตอร์ของมหาวิทยาลัย เพื่อให้ระบบค้นหาข้อมูลในฐานข้อมูลของมหาวิทยาลัยได้
4. แผงวงจรเชื่อมต่อ (Interface Card) ทำหน้าที่ Interface ระหว่างคอมพิวเตอร์กับเครือข่ายโทรศัพท์ ให้สามารถ Link Up กันได้
5. สายรับ-ส่งข้อมูลและการเดินสายโทรศัพท์ เพื่อใช้ในการแลกเปลี่ยนและรับ-ส่งข้อมูลของระบบกับเครือข่ายข้อมูลคอมพิวเตอร์ของมหาวิทยาลัย
6. คู่สายโทรศัพท์ เพื่อติดต่อกับผู้โทรเข้ามาใช้บริการ
7. เครื่องรับโทรศัพท์ใช้ในการทดสอบระบบ

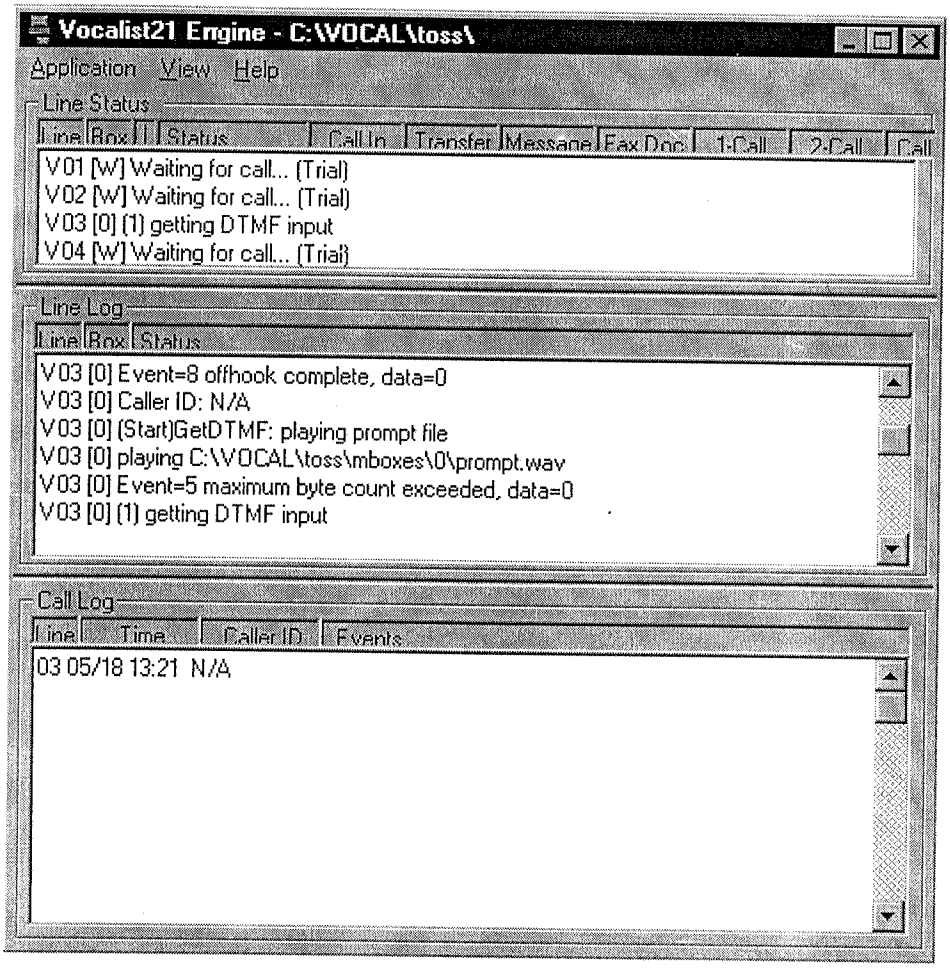
Software Component

1. ชุดโปรแกรมควบคุมการทำงานของระบบ เป็นชุดโปรแกรมที่ให้บริการระบบและควบคุมการทำงานของระบบทั้งหมด
2. แฟ้มข้อความเสียง เป็นข้อความเสียงที่บันทึกและจัดเก็บในรูปของข้อมูลดิจิตอล ซึ่งระบบจะใช้แปลงเป็นเสียงเพื่อติดต่อกับผู้ใช้บริการ
3. ฐานข้อมูล ได้แก่ ฐานข้อมูลของมหาวิทยาลัย เช่น ส่วนที่เป็นปฏิทินการศึกษา ผลการสอบ รายชื่อและรหัสนักศึกษา เป็นต้น

การพัฒนาเทคโนโลยี

เป็นการใช้เทคโนโลยีด้านคอมพิวเตอร์เป็นตัวควบคุม (Host Computer) การทำงานของระบบ ซึ่งคอมพิวเตอร์จะทำงานคล้ายกับเครื่องรับโทรศัพท์ และขณะเดียวกันก็ทำหน้าที่เป็นเจ้าหน้าที่คอยรับโทรศัพท์และตอบคำถาม รวมทั้งให้ข้อมูลและการประกาศข่าวแก่ผู้ที่สนใจด้วย ซึ่งคอมพิวเตอร์จะทำการตอบรับโดยอัตโนมัติเมื่อมีผู้โทรเข้า ซึ่งจะเป็นการสื่อสารและเผยแพร่ข่าวสารได้อย่างทั่วถึง แพร่หลาย

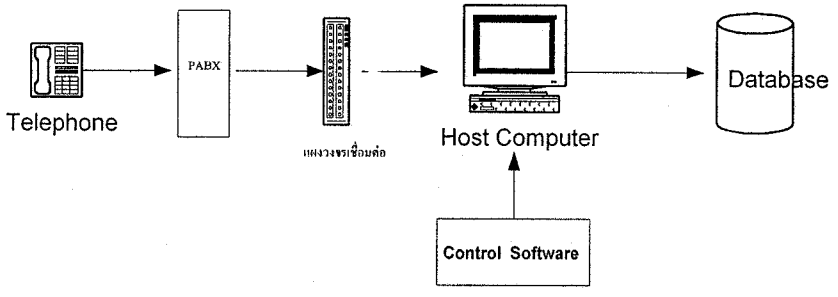
การใช้เทคโนโลยีด้านคอมพิวเตอร์เป็นตัวควบคุมการทำงานของระบบ จะต้องมี Software เป็นตัว



ภาพที่ 10 จอแสดงสถานะของโปรแกรม

ในส่วนของการบริการข่าวสารจะต้องเขียน Software กำหนดเป็น Menu สำหรับผู้โทรเข้ามาทำการเลือกรายการในการรับฟังข่าวสาร หรือในกรณีที่ของนิสิต นักศึกษา ที่จะขอทราบเกรด และผลสอบ ซึ่งในการเปลี่ยนแปลงรายการเลือก จะต้องสามารถทำได้ง่าย รวดเร็วและสะดวก การกำหนดความลึกของ Tree-struction จะต้องมีความลึกเพื่อให้สามารถกำหนดโครงสร้างของข้อมูลได้โดยง่าย ซึ่งขั้นตอนในการเขียน Software ดังกล่าวนี้ จะต้องมีการเขียนและทดสอบกับระบบไปด้วย ตัวอย่างของหน้าต่างโปรแกรมสามารถแสดงได้ดังนี้

ควบคุมการทำงานให้คอมพิวเตอร์และแผงวงจรเชื่อมต่อกันได้ พร้อมทั้งนำมาปรับปรุงให้เหมาะสมกับความต้องการของระบบในโครงการนี้ด้วย ซึ่ง software ที่โครงการวิจัยนี้ใช้จะเป็นการพัฒนาและปรับแต่ง software เดิมที่มีอยู่แล้ว โดยการทำงานดังที่กล่าวมาแล้วนั้นสามารถแสดงเป็นภาพได้ดังนี้



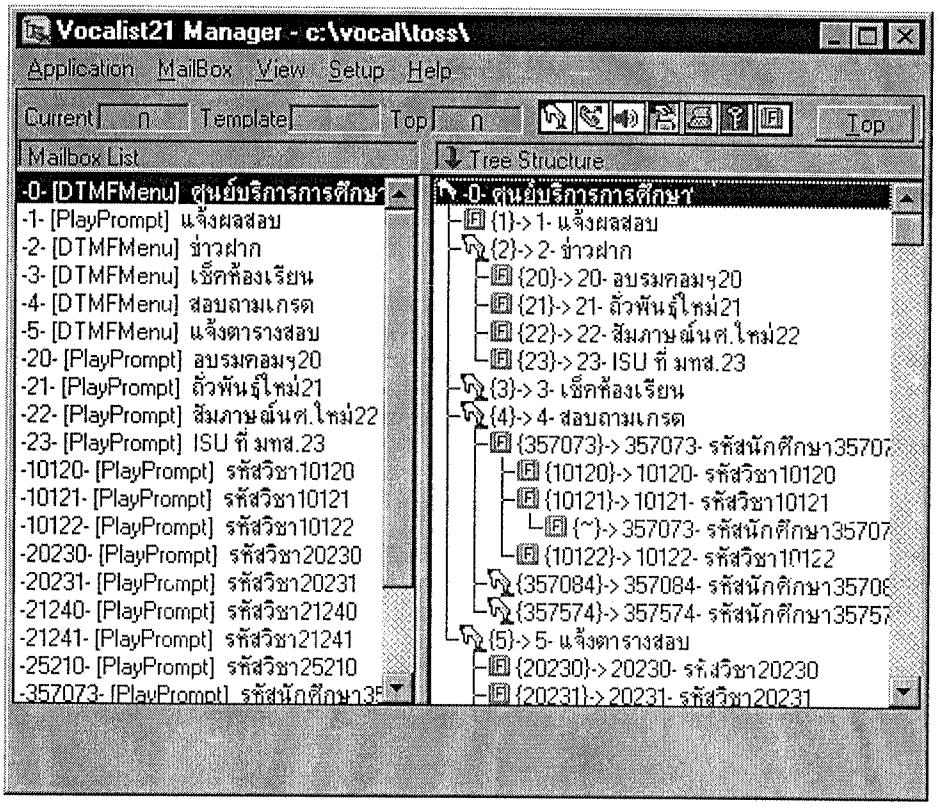
ภาพที่ 9 แสดงการทำงานของระบบ

การพัฒนาและปรับแต่ง Soft ware

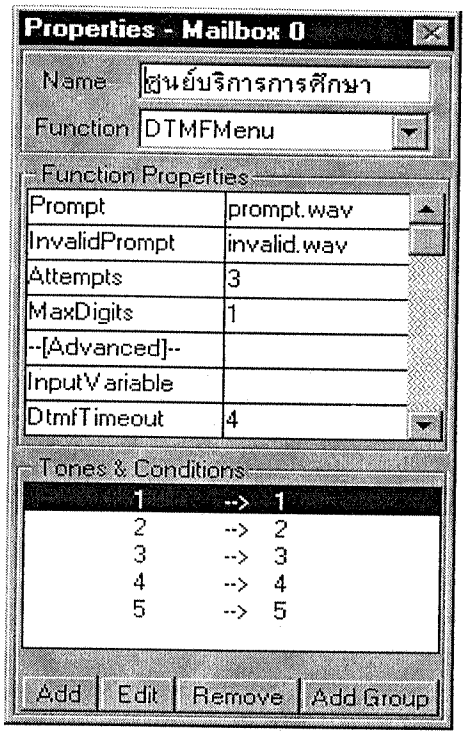
ในการพัฒนาและปรับแต่ง Software นั้น จะต้องเป็นไปตามขั้นตอนเพื่อจะได้ทราบ และวิเคราะห์ถึงข้อผิดพลาดที่เกิดขึ้นได้ เริ่มแรกโครงการวิจัยและพัฒนานี้ได้มีทางเลือกในการจัดการควบคุมระบบคือ เขียน Software ขึ้นใช้งานเองทั้งหมด ซึ่งต้องมีการทดสอบการทำงานไปทีละขั้นตอน แต่เมื่อวิเคราะห์และพิจารณาถึงระยะเวลาตามที่โครงการได้กำหนดไว้ จะทำให้ล่าช้า จึงได้นำเอา Software ที่มีขายอยู่แล้ว มาพัฒนาและปรับแต่งให้เหมาะสมกับการใช้งาน แต่ Softwaie ที่นำมาใช้เป็น Software ที่เป็น Freeware ไม่ได้ซื้อมาแต่อย่างใด จึงทำให้มีข้อจำกัดหลายด้านในการใช้งาน และปรับแต่ง เช่น อายุการทำงาน 30 วัน ไม่สามารถใช้ฟังก์ชันอื่น ๆ ที่จำเป็นได้ไม่มี Source Code ถึงกระนั้นก็ยังเป็น Software ที่มีประสิทธิภาพดี เหมาะสมกับจุดประสงค์กับการใช้งาน และพอใช้งานในการทดสอบบ้างบางส่วน แต่นั้นก็หมายความว่า Freeware ที่ได้มาไม่ได้สมบูรณ์แบบและพร้อมที่จะทำงานได้เลยทีเดียว ยังต้องปรับแต่งและแก้ไขส่วนของการ Setup และ Install เพื่อให้สามารถใช้งานได้ตามวัตถุประสงค์ของโครงการ โดยเฉพาะอย่างยิ่งการสร้างฟังก์ชันที่ควบคุม PC ให้สามารถทำงานได้เหมือนโทรศัพท์ เช่น Function Keypad, Character display, Screen การโทรออก การโอนสาย เป็นต้น

ตัวอย่างการทำงานของคอมพิวเตอร์ เช่น เมื่อมีการโทรเข้า จะต้องมีการแสดงเวลาของการโทรเข้ามาและแสดงสถานะของคู่สายว่าว่าง หรือไม่ว่าง ซึ่งตัวอย่างโปรแกรมสามารถแสดงได้ดังนี้

ภาพที่ 11 จอรับการกดเบอร์เพื่อหมุนเข้า



ภาพที่ 12 เมนูการเลือกรับฟังข่าวสาร



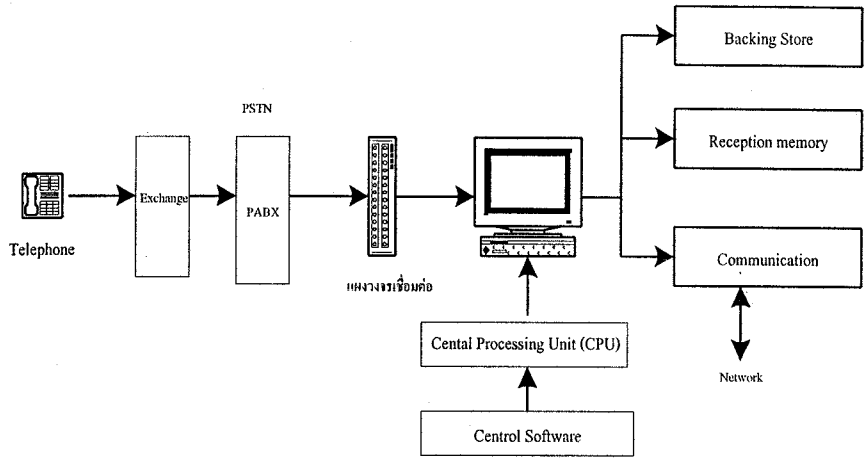
ภาพที่ 13 หน้าต่างการเติมเมนู

ขบวนการทำงานของระบบ

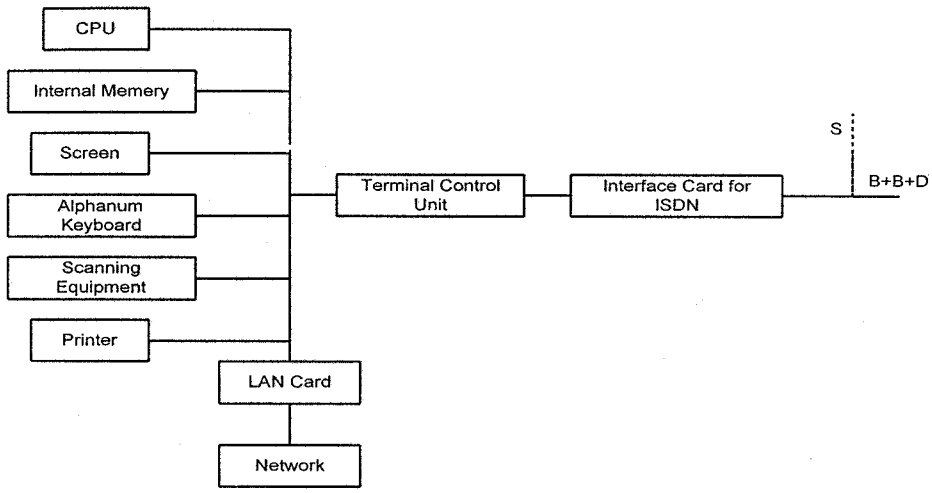
ขั้นตอนแรกก่อนที่จะเริ่มต้นของการทำงาน หรือมีการทดสอบขั้นตอนการทำงานได้นั้น จะต้องทำการ set ค่า configuration ของแผงวงจรเชื่อมต่อให้ได้ตามที่กำหนดไว้ของบริษัทผู้ผลิตแผงวงจร เพื่อให้ได้คุณสมบัติการใช้งานได้ตามที่ต้องการ โดยมีการ set ค่าต่างๆได้แก่

- การ set ค่า Hardware Interrupt Level (IRQ)
- การ set ค่า Base Memory Address Segment สำหรับแผงวงจรเชื่อมต่อ
- การ set ค่า Offset Address
- การกำหนดค่า Hardware Configuration อื่นๆ ด้วย

เมื่อทำการติดตั้งแผงวงจรเชื่อมต่อเข้าไปใน PC และทำการ Set ค่า Conficts เรียบร้อยแล้ว หลังจากนั้น จึงทำการพัฒนาและทดสอบแผงวงจรได้ ในหัวข้อนี้จะกล่าวถึงขบวนการทำงานของระบบ ซึ่งมีการทำงานดังนี้ เมื่อมีผู้โทรเข้า Software ก็จะไปสั่งงานให้แผงวงจรมีการทำงานตามที่ได้กำหนดตอนเขียน Software เช่นอนุญาตให้มีการโทรเข้าได้กี่ครั้ง ความเร็วของการยกหูรับสาย เมื่อมีการยกรับแล้ว ก็จะเป็นส่วนการทำงานของ PC ที่กำหนดการทำงานไว้ตามที่ได้กำหนด จนกระทั่งผู้โทรเข้าวางสาย องค์การโทรศัพท์ก็จะส่งสัญญาณมาที่เครื่อง PC ซึ่งทำหน้าคล้ายโทรศัพท์ว่าต้นสายได้มีการวางสายแล้ว PC ก็จะ Detect ได้แล้วจะทำงาน ON HOOK ทันที โดยไม่ต้องรอให้การสนทนาจบก่อน ขั้นตอนการทำงานที่กล่าวมาสามารถแสดงเป็นขั้นตอนตามบล็อกไดอะแกรมได้ดังต่อไปนี้



ภาพที่ 14 แสดงขบวนการทำงานของระบบ



ภาพที่ 15 แสดงผังการทำงานของระบบ

บทที่ 4
ผลการทดสอบและวิเคราะห์ข้อมูล

การทดสอบ

จากการทำการทดสอบและผลการทดสอบที่ได้จากการทดสอบระบบนั้น ทำให้ทราบว่าระบบสามารถทำงานได้อย่างมีประสิทธิภาพในระดับหนึ่งดังนี้ PC สามารถทำงานคล้ายเครื่องโทรศัพท์ได้ ขณะเดียวกันก็ทำงานคล้ายพนักงานรับโทรศัพท์ได้เหมือนกัน คอยตอบคำถาม และเผยแพร่ข่าวสารให้แก่ผู้ใช้บริการ โดยการทำงานเป็น Menu รายการเลือกสำหรับผู้โทร เข้าเลือกในการรับฟังข่าว นอกจากนี้นักศึกษาสามารถขอทราบผลเกรด และผลสอบได้รวมทั้งในอนาคตระบบยังสามารถพัฒนาให้นักศึกษาสามารถลงทะเบียนผ่านทางเครือข่ายระบบโทรศัพท์ได้อีกด้วย

ความสามารถของระบบ

ระบบจะสามารถให้บริการได้ตามที่กล่าวมาได้อย่างทั่วถึง ไม่จำกัดสถานที่ และเวลา เพียงแต่สถานที่นั้นมีโทรศัพท์ให้บริการ ก็สามารถใช้บริการได้ ซึ่งระบบสามารถให้บริการได้หลายอย่าง สำหรับมหาวิทยาลัย และบุคคลทั่วไป ได้แก่

- ระบบโอนสาย และเชื่อมต่อกับระบบวงจรอิเล็กทรอนิกส์อื่นได้
- รับสายโดยอัตโนมัติ และสามารถรับและบันทึกข้อความของผู้โทรเข้าได้
- แสดงรายการโดยอัตโนมัติเมื่อมีผู้โทรเข้า โดยจะแสดงเวลา และวันของการโทรเข้าและรายละเอียดอื่น ๆ ของการโทร
- สามารถเชื่อต่อกับระบบ FAX ได้
- สามารถรับและบันทึกข้อความเป็นลักษณะของ Voice Mail Box จากผู้โทรเข้า(Incomming call) รวมทั้งสามารถบันทึก digital call recording ได้
- สามารถบอกทวนรายการที่ผู้โทร ทำการเลือกเป็น Voice file ได้

ความลึกของ Tree-Struction และความจุข้อมูล

ระบบสามารถบันทึกข้อมูลได้อย่างไม่จำกัด จนกว่าจะเต็มพื้นที่ของหน่วยความจำหลัก (Harddisk) ส่วนความลึกของ Tree Struction ผู้กำหนดโครงสร้างสามารถทำได้อย่างอิสระ ไม่จำกัด เพราะสามารถกำหนดความลึกของ Tree Struction ได้อย่างไม่จำกัดจนกว่าจะเต็มพื้นที่ของหน่วยความจำหลัก (Harddisk)

การเปลี่ยนแปลงข้อมูลของระบบ

การเปลี่ยนแปลงข้อมูลสามารถทำได้โดยง่าย โดยสามารถทำได้โดยการใช้โทรศัพท์โดยมีรหัสผ่าน (Password) เพื่อที่จะทำการเปลี่ยนแปลงข้อมูล หรืออาจทำการบันทึกเป็น Voice file โดยใช้ไมโครโฟน จึงทำให้การเปลี่ยนแปลงข้อมูลเป็นไปได้อย่างรวดเร็วทันต่อเหตุการณ์

สิ่งที่ทำการทดสอบ

สิ่งที่ผู้วิจัยและพัฒนาได้ดำเนินการทดสอบคือ การทดสอบการสื่อสารผ่านเครือข่ายโทรศัพท์ จะทดสอบว่าระบบสามารถทำการสื่อสารผ่านเครือข่ายโทรศัพท์ ได้จริงตามทฤษฎี ความสามารถของการ์ดที่โครงการมีอยู่ได้จริงหรือไม่ และการทดสอบว่าระบบสามารถเชื่อมต่อกับเครือข่าย ข้อมูลคอมพิวเตอร์ได้หรือไม่ โดยผู้วิจัยและพัฒนาได้แบ่งการทดสอบออกเป็นส่วนๆ ดังนี้

1. การทดสอบว่าระบบสามารถทำการสื่อสารผ่านเครือข่ายโทรศัพท์ ได้จริงโดยมีการทดสอบดังนี้
   - ก. เมื่อมีการโทรเข้า ระบบสามารถทำงานการตอบรับ หรือยกหู (OFFHOOK) กับผู้โทรเข้ามาได้จริง
   - ข. เมื่อระบบทำการตอบรับ หรือ ยกหู ได้จริงแล้ว ระบบสามารถสนทนากับผู้โทร เข้าใช้บริการเป็น Voice file ได้จริง
   - ค. ระบบสามารถทำการ Play Voice file ได้จริง และตรงตามที่ผู้ใช้บริการเลือกรายการจาก Menu
   - ง. ระบบสามารถวางหู (ONHOOK) หรือทำการจบการสนทนาได้โดยทันทีเมื่อผู้โทรเข้า ทำการจบการสนทนาหรือ วางหู
   - จ. ระบบสามารถรับ Input เป็น Digital จากผู้โทรเข้า แล้วสร้างเป็น File ได้ และหลังจากสร้างเป็น Voice File สามารถทำการ Play Voice File นั้นกลับไป หรือสนทนากับผู้โทรเข้าได้
   - ฉ. ระบบสามารถสร้างเป็นรายการให้ผู้โทรใช้บริการได้เลือก Menu ได้
2. การทดสอบว่าระบบสามารถเชื่อมต่อกับระบบเครือข่ายฐานข้อมูลคอมพิวเตอร์ของมหาวิทยาลัยได้จริง การทดสอบโดยการทดลองทำการเปลี่ยนแปลงและเข้าไปเปลี่ยนข้อมูลระบบเครือข่ายระบบฐานข้อมูลคอมพิวเตอร์

เงื่อนไขการทดสอบ

เงื่อนไขการทดสอบว่าระบบสามารถทำการสื่อสารผ่านเครือข่ายโทรศัพท์ได้จริง มีเงื่อนไขการทดสอบดังนี้

1. จำนวนสัญญาณกระดิ่ง หรือ สัญญาณ Pulse ในการยกหูรับ หรือตอบรับของระบบเครือข่าย กล่าวคือ ในการตอบรับ ผู้โทรเข้า ระบบสามารถตอบรับได้เร็วทันทีโดยการรอรับสัญญาณ Pulse ไม่เกิน 4 Pulse เป็นอย่างมาก

2. ความลึกของจำนวนรายการ สำหรับการผู้โทรเข้ามาเลือกทำรายการสามารถกำหนดความลึกและความซับซ้อนได้ตามที่ต้องการ

3. ขั้นตอนและวิธีการเปลี่ยนแปลงข้อมูล

4. ความสะดวกแล่ายต่อการเปลี่ยนแปลงข้อมูล

5. ความจุของข้อมูลในระบบ

6. โทรศัพท์แต่ละ Channel จะ Contact กับโทรศัพท์ได้ 1 คู่สาย เท่านั้น หมายความว่า Card วงจรเชื่อมที่โครงการที่มีอยู่ 1 Card จะสามารถรองรับโทรศัพท์ ได้ 4 คู่สายในเวลาเดียวกัน

บทที่ 5
บทสรุป

สรุปผลการวิจัย

จากการพัฒนาและทดลองระบบดังที่กล่าวมา คือ ระบบสามารถทำงานได้ตามเป้าหมาย และจุดประสงค์ของโครงการ ซึ่งได้แก่ระบบสามารถรับสายโทรศัพท์ได้โดยอัตโนมัติ สามารถให้ข้อมูลข่าวสารแก่ผู้ที่สนใจ และนิสิตนักศึกษาได้ เช่น การประกาศข่าว การแจ้งตารางสอบ แจ้งผลสอบ การประชุมสัมมนา หรือการฝึกอบรบต่างๆ รวมทั้งข้อมูล ข่าวสารต่างๆ ที่ทางมหาวิทยาลัยมีความประสงค์ที่จะเผยแพร่ ซึ่งทำโดยการเก็บข้อมูลเป็นรูป Voice file ไว้ในคอมพิวเตอร์ซึ่งอาจทำได้โดยการบันทึกผ่านไมโครโฟน หรือผ่านทางโทรศัพท์ได้

การบริการต่างดังที่กล่าวมาสามารถให้บริการได้อย่างทั่วถึง ความถูกต้องของข้อมูลสูง ทันต่อเหตุการณ์ และข้อมูลที่เผยแพร่ออกไปจะข้อมูลล่าสุดที่ถูกบันทึกไว้

เนื่องจากมหาวิทยาลัยมีระบบการลงทะเบียนเป็นแบบระบบปิด กล่าวคือ มีการกำหนดจำนวนนักศึกษาที่ลงทะเบียนรายวิชานั้นๆ ดังนั้นระบบจะต้องมีการพัฒนา Software ไปอีกระดับหนึ่ง ในอนาคตโครงการวิจัยและพัฒนาจะสามารถพัฒนา ระบบให้นิสิตนักศึกษาสามารถลงทะเบียนผ่านเครือข่ายโทรศัพท์ได้ เพราะฉะนั้นการที่จะให้นักศึกษาสามารถลงทะเบียนผ่านทางเครือข่ายโทรศัพท์ได้ จึงอยู่ในระหว่างการพัฒนา ซึ่งในอนาคตอันใกล้นี้ น่าจะสัมฤทธิ์ผล

ข้อเสนอแนะ

1. การวิจัยแบบนี้ ต้องใช้เครื่องมือ และความรู้ทางวิศวกรรมศาสตร์อย่างมาก ทำให้เป็นอุปสรรคอย่างมากเวลาทำวิจัย ทั้งเครื่องมือ และบุคลากร ดังนั้น ควรจะกำหนดขอบเขตของโครงงานให้ดี มิเช่นนั้นอาจ จะทำไม่ได้ในเงินทุนและเวลาที่กำหนด

2. วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้องมากที่สุดคือ Voice Processing Card ซึ่งเป็นวงจร ที่มีราคาแพง และ หา Driver Software ยากมาก ดังนั้นเวลาซื้อต้องให้แน่ใจถึง Version และชนิดของการเชื่อมต่อ (Bus) แนะนำให้ใช้ ISA Bus จะหาเครื่องใช้ได้ง่าย แต่ช้ากว่ามาก

3. เนื่องจากคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่ ไม่ได้ใช้งานเต็มประสิทธิภาพอยู่แล้ว การนำเวลาคอมพิวเตอร์มาใช้แทนระบบโทรศัพท์จึง เป็นสิ่งที่ช่วยให้เราใช้ทรัพยากรของชาติอย่างเต็มที่ จึงควรจะเริ่มมีการนำเอาเทคโนโลยีด้านนี้มาใช้เพื่อประหยัดเวลา และค่าใช้จ่ายมากขึ้น

บรรณานุกรม

โกศล เพ็ชร์สุวรรณ และ ชิเงกิ โชจิ. (2534). เทคโนโลยีโทรคมนาคม. บริษัทสำนักพิมพ์ดวงกมล จำกัด: กรุงเทพมหานคร.

ฉัตรชัย สุมามาลย์. (2521). การสื่อสารข้อมูลคอมพิวเตอร์และระบบเครือข่าย. บริษัทด่านสุทธาการพิมพ์ จำกัด: กรุงเทพมหานคร.

สุริยัน ศรีสวัสดิ์กุล. (2539). ระบบสื่อสารข้อมูลคอมพิวเตอร์. บริษัทสยามสปอร์ต ซินดิเกท จำกัด: กรุงเทพมหานคร.

วิสันติ์ อาชาเดโชพล. ระบบโทรศัพท์ดิจิตอล. หจก. สำนักพิมพ์ฟิสิกส์เซ็นเตอร์: กรุงเทพมหานคร.

รัชนัย อินทุใส. (2537). ISDN. หจก.สำนักพิมพ์ฟิสิกส์เซ็เตอร์: กรุงเทพมหานคร.

วิวัฒน์ กิรานนท์. (2536). พื้นฐานการสื่อสารวิศวกรรมสถานแห่งประเทศไทยในพระบรมราชูปถัมภ์. จุฬาลงกรณ์มหาวิทยาลัย: กรุงเทพมหานคร.

ED Tittrl and Dawn Rader. (1996). Computer Telephony. Automating home offices and small businesses, United Stated: ACADEMIC PRESS, INC.

Matin . 1990. Telecommunication and The Computer. Prentice-Hall.

Tanenbaum, A. (1981). Computer Networks. Prentice-Hall.

Black, U. (1987). Computer Networks: Protocols, Standards and Interfaces. Inglewood Cliffis. N.J. Prentice-Hall, Inc.

ภาคผนวก ก

โปรแกรมที่ 1

```
Option Explicit
Sub ssfPlayNumber(nextstate$, n&)

Dim nextsubstate$, temp&, strip%, lineNo%

lineNo% = OleData(0)

 If substate$(lineNo%) = "" Then
  If n& >= 1000000000 Or n& < 0 Then
   Call vcfContinue(nextstate$)
   substate$(lineNo%) = ""
   Exit Sub
  ElseIf n& >= 1000000 Then
   substate$(lineNo) = "100,000,000"
  ElseIf n& >= 1000 Then
   substate$(lineNo%) = "100,000"
  Else
   substate$(lineNo%) = "100"
  End If
 End If

 Select Case substate$(lineNo%)

 ' Millions-----------------------------------------------------------------
 Case "100,000,000"
  If n& >= 100000000 Then
   Call vcfPlayFile(State$(lineNo%), "System", Right(Str(n& \ 100000000), 1) & ".wav")
   nextsubstate$ = "100MILLION"
  Else
```

```
    Call Jump(State$(lineNo%))
    nextsubstate$ = "10,000,000"
  End If
Case "100MILLION"
  Call vcfPlayFile(State$(lineNo%), "System", "hundred.wav")
  nextsubstate$ = "10,000,000"
Case "10,000,000"
  If n& Mod 100000000 >= 20000000 Then
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 10000000) Mod 10), 1) & "0.wav")
    nextsubstate$ = "1,000,000"
  Else
    Call Jump(State$(lineNo%))
    nextsubstate$ = "1,000,000"
  End If
Case "1,000,000"
  temp& = (n& \ 1000000) Mod 100
  If temp& >= 20 Then temp& = temp& Mod 10
  If temp& <> 0 Then
    If temp& > 9 Then strip% = 2 Else strip% = 1
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str(temp&), strip%) & ".wav")
  Else
    Call Jump(State$(lineNo%))
  End If
  nextsubstate$ = "MILLION"
Case "MILLION"
  Call vcfPlayFile(State$(lineNo%), "System", "million.wav")
  nextsubstate$ = "100,000"

'Thousands-----------------------------------------------------------------
Case "100,000"
  If n& Mod 1000000 >= 100000 Then
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 100000) Mod 10), 1) & ".wav")
    nextsubstate$ = "100THOUSAND"
```

```
 Else
  Call Jump(State$(lineNo%))
  nextsubstate$ = "10,000"
 End If
Case "100THOUSAND"
 Call vcfPlayFile(State$(lineNo%), "System", "hundred.wav")
 nextsubstate$ = "10,000"
Case "10,000"
 If n& Mod 100000 >= 20000 Then
  Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 10000) Mod 10), 1) & "0.wav")
  nextsubstate$ = "1,000"
 Else
  Call Jump(State$(lineNo%))
  nextsubstate$ = "1,000"
 End If
Case "1,000"
 temp& = (n& \ 1000) Mod 100
 If temp& >= 20 Then temp& = temp& Mod 10
 If temp& <> 0 Then
  If temp& > 9 Then strip% = 2 Else strip% = 1
  Call vcfPlayFile(State$(lineNo%), "System", Right(Str(temp&), strip%) & ".wav")
 Else
  Call Jump(State$(lineNo%))
 End If
 nextsubstate$ = "THOUSAND"
Case "THOUSAND"
 Call vcfPlayFile(State$(lineNo%), "System", "thousand.wav")
 nextsubstate$ = "100"

'Ones----------------------------------------------------------------
Case "100"
 If n& Mod 1000 >= 100 Then
  Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 100) Mod 10), 1) & ".wav")
```

```
    nextsubstate$ = "100HUNDRED"
   Else
    Call Jump(State$(lineNo%))
    nextsubstate$ = "10"
   End If
  Case "100HUNDRED"
   Call vcfPlayFile(State$(lineNo%), "System", "hundred.wav")
   nextsubstate$ = "10"
  Case "10"
   If n& Mod 100 >= 20 Then
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 10) Mod 10), 1) & "0.wav")
    nextsubstate$ = "1"
   Else
    Call Jump(State$(lineNo%))
    nextsubstate$ = "1"
   End If
  Case "1"
   temp& = n& Mod 100
   If temp& >= 20 Then temp& = temp& Mod 10
   If temp& <> 0 Then
    If temp& > 9 Then strip% = 2 Else strip% = 1
    Call vcfPlayFile(nextstate$, "System", Right(Str(temp&), strip%) & ".wav")
   Else
    Call vcfContinue(nextstate$)
   End If
   nextsubstate$ = ""
  End Select

  substate$(lineNo%) = nextsubstate$

End Sub

Sub ssfPlayCurrency(nextstate$, n&, c&)
```

```
Dim nextsubstate$, temp&, strip%, lineNo%

lineNo% = OleData(0)

If substate$(lineNo%) = "" Then
 If n& >= 1000000000 Or n& < 0 Then
  Call vcfContinue(nextstate$)
  substate$(lineNo%) = ""
  Exit Sub
 ElseIf n& >= 1000000 Then
  substate$(lineNo) = "100,000,000"
 ElseIf n& >= 1000 Then
  substate$(lineNo%) = "100,000"
 Else
  substate$(lineNo%) = "100"
 End If
End If

Select Case substate$(lineNo%)

' Millions-------------------------------------------------------------
Case "100,000,000"
 If n& >= 100000000 Then
  Call vcfPlayFile(State$(lineNo%), "System", Right(Str(n& \ 100000000), 1) & ".wav")
  nextsubstate$ = "100MILLION"
 Else
  Call Jump(State$(lineNo%))
  nextsubstate$ = "10,000,000"
 End If
Case "100MILLION"
 Call vcfPlayFile(State$(lineNo%), "System", "hundred.wav")
 nextsubstate$ = "10,000,000"
Case "10,000,000"
```

```
  If n& Mod 100000000 >= 20000000 Then
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 10000000) Mod 10), 1) & "0.wav")
    nextsubstate$ = "1,000,000"
  Else
    Call Jump(State$(lineNo%))
    nextsubstate$ = "1,000,000"
  End If
Case "1,000,000"
  temp& = (n& \ 1000000) Mod 100
  If temp& >= 20 Then temp& = temp& Mod 10
  If temp& <> 0 Then
    If temp& > 9 Then strip% = 2 Else strip% = 1
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str(temp&), strip%) & ".wav")
  Else
    Call Jump(State$(lineNo%))
  End If
  nextsubstate$ = "MILLION"
Case "MILLION"
  Call vcfPlayFile(State$(lineNo%), "System", "million.wav")
  nextsubstate$ = "100,000"

'Thousands----------------------------------------------------------------
Case "100,000"
  If n& Mod 1000000 >= 100000 Then
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 100000) Mod 10), 1) & ".wav")
    nextsubstate$ = "100THOUSAND"
  Else
    Call Jump(State$(lineNo%))
    nextsubstate$ = "10,000"
  End If
Case "100THOUSAND"
  Call vcfPlayFile(State$(lineNo%). "System", "hundred.wav")
  nextsubstate$ = "10,000"
```

```
Case "10,000"
 If n& Mod 100000 >= 20000 Then
  Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 10000) Mod 10), 1) & "0.wav")
  nextsubstate$ = "1,000"
 Else
  Call Jump(State$(lineNo%))
  nextsubstate$ = "1,000"
 End If
Case "1,000"
 temp& = (n& \ 1000) Mod 100
 If temp& >= 20 Then temp& = temp& Mod 10
 If temp& <> 0 Then
  If temp& > 9 Then strip% = 2 Else strip% = 1
  Call vcfPlayFile(State$(lineNo%), "System", Right(Str(temp&), strip%) & ".wav")
 Else
  Call Jump(State$(lineNo%))
 End If
 nextsubstate$ = "THOUSAND"
Case "THOUSAND"
 Call vcfPlayFile(State$(lineNo%), "System", "thousand.wav")
 nextsubstate$ = "100"

'Ones-------------------------------------------------------------
Case "100"
 If n& Mod 1000 >= 100 Then
  Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 100) Mod 10), 1) & ".wav")
  nextsubstate$ = "100HUNDRED"
 Else
  Call Jump(State$(lineNo%))
  nextsubstate$ = "10"
 End If
Case "100HUNDRED"
 Call vcfPlayFile(State$(lineNo%), "System", "hundred.wav")
```

```
  nextsubstate$ = "10"
Case "10"
  If n& Mod 100 >= 20 Then
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str((n& \ 10) Mod 10), 1) & "0.wav")
    nextsubstate$ = "1"
  Else
    Call Jump(State$(lineNo%))
    nextsubstate$ = "1"
  End If
Case "1"
  temp& = n& Mod 100
  If temp& >= 20 Then temp& = temp& Mod 10
  If temp& <> 0 Then
    If temp& > 9 Then strip% = 2 Else strip% = 1
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str(temp&), strip%) & ".wav")
  Else
    Call Jump(State$(lineNo%))
  End If
  nextsubstate$ = "DOLLARS"
Case "DOLLARS"
  If n& = 0 Then
    Call Jump(State$(lineNo%))
  ElseIf n& = 1 Then
    Call vcfPlayFile(State$(lineNo%), "System", "dollar.wav")
  Else
    Call vcfPlayFile(State$(lineNo%), "System", "dollars.wav")
  End If
  nextsubstate$ = "10CENTS"
Case "10CENTS"
  If c& Mod 100 >= 20 Then
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str((c& \ 10) Mod 10), 1) & "0.wav")
    nextsubstate$ = "1CENTS"
  Else
```

```
    Call Jump(State$(lineNo%))
    nextsubstate$ = "1CENTS"
   End If
  Case "1CENTS"
   temp& = c& Mod 100
   If temp& >= 20 Then temp& = temp& Mod 10
   If temp& <> 0 Then
    If temp& > 9 Then strip% = 2 Else strip% = 1
    Call vcfPlayFile(State$(lineNo%), "System", Right(Str(temp&), strip%) & ".wav")
   Else
    Call Jump(State$(lineNo%))
   End If
   nextsubstate$ = "CENTS"
  Case "CENTS"
   If c& = 0 Then
    Call vcfContinue(nextstate$)
   ElseIf c& = 1 Then
    Call vcfPlayFile(nextstate$, "System", "cent.wav")
   Else
    Call vcfPlayFile(nextstate$, "System", "cents.wav")
   End If
   nextsubstate$ = ""

  End Select

  substate$(lineNo%) = nextsubstate$

End Sub

Sub ssfPlayTime(nextstate$, MsgTime$)

  Dim nextsubstate$, temp&, strip%, lineNo%
```

```
lineNo% = OleData(0)

If substate$(lineNo%) = "" Then
 substate$(lineNo%) = "HOUR"
End If

Select Case substate$(lineNo%)

Case "HOUR"
 If Left(MsgTime$, 2) <= "12" Then
  If Left(MsgTime$, 2) = "00" Then
   Call vcfPlayFile(State$(lineNo%), "System", "12.wav")
  Else
   Call vcfPlayFile(State$(lineNo%), "System", Val(Left(MsgTime$, 2)) & ".wav")
  End If
 Else
  Call vcfPlayFile(State$(lineNo%), "System", (Val(Left(MsgTime$, 2)) - 12) & ".wav")
 End If
 nextsubstate$ = "MINUTE"

Case "MINUTE"
 If Mid(MsgTime$, 3, 2) >= "20" Then
  Call vcfPlayFile(State$(lineNo%), "System", Mid(MsgTime$, 3, 1) & "0.wav")
  If Right(MsgTime$, 1) = "0" Then nextsubstate$ = "AMPM" Else nextsubstate$ = "MINUTE1"
 ElseIf Mid(MsgTime$, 3, 2) >= "10" Then
  Call vcfPlayFile(State$(lineNo%), "System", Mid(MsgTime$, 3, 2) & ".wav")
  nextsubstate$ = "AMPM"
 ElseIf Mid(MsgTime$, 3, 2) = "00" Then
  Call Jump(State$(lineNo%))
  nextsubstate$ = "AMPM"
 Else
  Call vcfPlayFile(State$(lineNo%), "System", "o.wav")
  nextsubstate$ = "MINUTE1"
```

```
  End If
 Case "MINUTE1"
  Call vcfPlayFile(State$(lineNo%), "System", Right(MsgTime$, 1) & ".wav")
  nextsubstate$ = "AMPM"

 Case "AMPM"
  If Left(MsgTime$, 2) < "12" Then
   Call vcfPlayFile(nextstate$, "System", "am.wav")
  Else
   Call vcfPlayFile(nextstate$, "System", "pm.wav")
  End If
  nextsubstate$ = ""
 End Select

 substate$(lineNo%) = nextsubstate$

End Sub

Sub ssfPlayDate(nextstate$, sDate$)

 Dim nextsubstate$, lineNo%

 lineNo% = OleData(0)

 If substate$(lineNo%) = "" Then
  substate$(lineNo%) = "Month"
 End If

 Select Case substate$(lineNo%)

 Case "Month"

  Call vcfPlayFile(State$(lineNo%), "System", "month" & Left(sDate$, 2) & ".wav")
```

```
  nextsubstate$ = "Day"

 Case "Day"
  If Mid(sDate$, 3, 2) >= "20" Then
   If Right(sDate$, 1) = "0" Then
    Call vcfPlayFile(nextstate$, "System", Mid(sDate$, 3, 1) & "0.wav")
    nextsubstate$ = ""
   Else
    Call vcfPlayFile(State$(lineNo%), "System", Mid(sDate$, 3, 1) & "0.wav")
    nextsubstate$ = "Day1"
   End If
  ElseIf Mid(sDate$, 3, 2) >= "10" Then
   Call vcfPlayFile(nextstate$, "System", Mid(sDate$, 3, 2) & ".wav")
   nextsubstate$ = ""
  Else
   Call Jump(State$(lineNo%))
   nextsubstate$ = "Day1"
  End If

 Case "Day1"
  Call vcfPlayFile(nextstate$, "System", Mid(sDate$, 4, 1) & ".wav")
  nextsubstate$ = ""

 End Select

 substate$(lineNo%) = nextsubstate$

End Sub

Sub ssfPlayString(nextstate$, s$)

 Dim onchar%, lineNo%
```

```
lineNo% = OleData(0)

If substate$(lineNo%) = "" Then
  substate$(lineNo) = "1"
End If

onchar% = Val(substate$(lineNo%))

If Len(s$) > onchar% Then
  Call vcfPlayFile(State$(lineNo%), "System", Mid$(s$, onchar%, 1) & ".wav")
  substate$(lineNo%) = Str(onchar% + 1)
ElseIf Len(s$) = onchar% Then
  Call vcfPlayFile(nextstate$, "System", Mid$(s$, onchar%, 1) & ".wav")
  substate$(lineNo%) = ""
Else
  Call vcfContinue(nextstate$)
  substate$(lineNo%) = ""
End If

End Sub
```

ภาคผนวก ข

โปรแกรมที่ 2

```
Option Explicit

Sub vcfContinue(nextstate$)

  OleData(1) = "Continue"
  OleData(2) = ""
  OleData(3) = 0
  OleData(4) = 0
  OleData(5) = 0

  State$(OleData(0)) = nextstate$

End Sub

Sub Jump(nextstate$)

  State$(OleData(0)) = "JUMP:" & nextstate$

End Sub

Sub vcfDial(nextstate$, dialstring$)

  OleData(1) = "Dial"
  OleData(2) = dialstring$
  OleData(3) = 0
  OleData(4) = 0
  OleData(5) = 0

  State$(OleData(0)) = nextstate$
```

```
End Sub

Sub vcfDelay(nextstate$, delaytime&)

 OleData(1) = "Delay"
 OleData(2) = ""
 OleData(3) = 0
 OleData(4) = 0
 OleData(5) = delaytime&

 State$(OleData(0)) = nextstate$

End Sub

Sub vcfCallOut(nextstate$, dialstring$, rings%)

 OleData(1) = "CallOut"
 OleData(2) = dialstring$
 OleData(3) = rings%
 OleData(4) = 0
 OleData(5) = 0

 State$(OleData(0)) = nextstate$

End Sub

Sub vcfCallOutSC(nextstate$, dialstring$, rings%, SC%)

 OleData(1) = "CallOutSC"
 OleData(2) = dialstring$
 OleData(3) = rings%
 OleData(4) = SC%
 OleData(5) = 0
```

```
 State$(OleData(0)) = nextstate$

End Sub

Sub vcfFlushDigits(nextstate$)

 OleData(1) = "FlushDigits"
 OleData(2) = ""
 OleData(3) = 0
 OleData(4) = 0
 OleData(5) = 0

 State$(OleData(0)) = nextstate$

End Sub

Sub vcfPlayForward(nextstate$, where$, file$, forward&)

 If where$ = "System" Then
  OleData(2) = App.Path & "\sysvox\" & file$
 ElseIf where$ = "Current" Then
  OleData(2) = AppDir & "mboxes\" & OleData(1) & "\" & file$
 Else
  OleData(2) = file$
 End If

 If Language%(OleData(0)) <> 0 Then
  If isfile(Left(OleData(2), Len(OleData(2)) - 1) & Language%(OleData(0))) Then
   OleData(2) = Left(OleData(2), Len(OleData(2)) - 1) & Language%(OleData(0))
  End If
 End If

 OleData(1) = "PlayFile"
```

```
 OleData(3) = 0
 OleData(4) = 0
 If forward& = 0 Then forward& = 1
 OleData(5) = forward&

 State$(OleData(0)) = nextstate$

 FilePlaying$(OleData(0)) = OleData(2)

End Sub

Sub vcfPlaySeek(nextstate$, where$, file$, forward&)

 If where$ = "System" Then
  OleData(2) = App.Path & "\sysvox\" & file$
 ElseIf where$ = "Current" Then
  OleData(2) = AppDir & "mboxes\" & OleData(1) & "\" & file$
 Else
  OleData(2) = file$
 End If

 If Language%(OleData(0)) <> 0 Then
  If isfile(Left(OleData(2), Len(OleData(2)) - 1) & Language%(OleData(0))) Then
   OleData(2) = Left(OleData(2), Len(OleData(2)) - 1) & Language%(OleData(0))
  End If
 End If

 OleData(1) = "PlayFile"

 OleData(3) = -1
 OleData(4) = 0
 If forward& = 0 Ther. forward& = 1
 OleData(5) = forward&
```

```
State$(OleData(0)) = nextstate$

FilePlaying$(OleData(0)) = OleData(2)

End Sub

Sub vcfGetDigits(nextstate$, MaxDigits%, DigitsTimeout%)

OleData(1) = "GetDigits"
OleData(2) = ""
OleData(3) = MaxDigits%
OleData(4) = DigitsTimeout%
OleData(5) = 0   'maxtime&

State$(OleData(0)) = nextstate$

End Sub

Sub vcfGoto(Mailbox$)

OleData(1) = "Goto"
OleData(2) = Mailbox$
OleData(3) = 0
OleData(4) = 0
OleData(5) = 0

End Sub

Sub vcfHangup()

OleData(1) = "Hangup"
OleData(2) = ""
OleData(3) = 0
```

```
  OleData(4) = 0
  OleData(5) = 0

End Sub

Sub vcfReturn()

  OleData(1) = "Return"
  OleData(2) = ""
  OleData(3) = 0
  OleData(4) = 0
  OleData(5) = 0

End Sub

Sub vcfPickup(nextstate$)

  OleData(1) = "Pickup"
  OleData(2) = ""
  OleData(3) = 0
  OleData(4) = 0
  OleData(5) = 0

  State$(OleData(0)) = nextstate$

End Sub

Sub vcfPlayFile(nextstate$, where$, file$)

 If where$ = "System" Then
   OleData(2) = App.Path & "\sysvox\" & file$
 ElseIf where$ = "Current" Then
   OleData(2) = AppDir & "mboxes\" & OleData(1) & "\" & file$
```

```
Else
 OleData(2) = file$
End If

If Language%(OleData(0)) <> 0 Then
 If isfile(Left(OleData(2), Len(OleData(2)) - 1) & Language%(OleData(0))) Then
  OleData(2) = Left(OleData(2), Len(OleData(2)) - 1) & Language%(OleData(0))
 End If
End If

OleData(1) = "PlayFile"

OleData(3) = 0
OleData(4) = 0
OleData(5) = 0

State$(OleData(0)) = nextstate$

FilePlaying$(OleData(0)) = OleData(2)

End Sub

Sub vcfRecord(nextstate$, where$, file$, maxtime&)

 If where$ = "System" Then
  OleData(2) = App.Path & "\sysvox\" & file$
 ElseIf where$ = "Current" Then
  OleData(2) = AppDir & "mboxes\" & OleData(1) & "\" & file$
 Else
  OleData(2) = file$
 End If

 OleData(1) = "Record"
```

```
  OleData(3) = 0
  OleData(4) = 0
  OleData(5) = maxtime&

  State$(OleData(0)) = nextstate$

  FilePlaying$(OleData(0)) = OleData(2)

End Sub

Sub vcfRecordAppend(nextstate$, where$, file$, maxtime&)

  If where$ = "System" Then
    OleData(2) = App.Path & "\sysvox\" & file$
  ElseIf where$ = "Current" Then
    OleData(2) = AppDir & "mboxes\" & OleData(1) & "\" & file$
  Else
    OleData(2) = file$
  End If

  OleData(1) = "Record"
  OleData(3) = -1
  OleData(4) = 0
  OleData(5) = maxtime&

  State$(OleData(0)) = nextstate$

  FilePlaying$(OleData(0)) = OleData(2)
End Sub

Sub vcfSendFax(nextstate$, FaxInfo$)

  OleData(1) = "SendFax"
```

```
OleData(2) = FaxInfo$
OleData(3) = 0
OleData(4) = 0
OleData(5) = 0

State$(OleData(0)) = nextstate$

End Sub
48
56
48
48
```

ภาคผนวก ค

โปรแกรมที่ 3

```
Option Explicit
Declare Function GetPrivateProfileInt Lib "Kernel" (ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal nDefault As Integer, ByVal lpFileName As String) As Integer
Declare Function GetPrivateProfileString Lib "Kernel" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Integer, ByVal lpFileName As String) As Integer
Declare Function WritePrivateProfileString Lib "Kernel" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lplFileName As String) As Integer

Type QueueType
  Mailbox As String
  TimeAdded As Date
End Type
Type ParseType
  Parastr As String
  quoted As Boolean
End Type
Global Const Maxline = 74
Global AppDir$
Global OleData(0 To 5), OleLog$, OleCallLog$
Global State$(1 To 74), substate$(1 To 74)
Global FilePlaying$(1 To 74)
Global Language%(1 To 74)
Global RefreshSched As Boolean, RefreshMgr As Boolean
Sub Main()
End Sub
Sub CvtWaveFile(ByVal szSrcFil As String, ByVal szDstFil As String)
  Dim str_vox As String * 128, str_wav As String * 1
  Dim i&, ll As Long, ii As Integer
```

```
  Dim ss As String, FileLength&
  Dim tempfile$
'  MousePointer = 11
On Error GoTo errhandler
FileLength = FileLen(szSrcFil) ' Get length of file.
If isfile(szDstFil) Then Kill szDstFil
  Open szSrcFil For Binary Access Read As #1
tempfile$ = Left(szDstFil, Len(szDstFil) - 3) & "tmp"
  If isfile(tempfile$) Then Kill tempfile$
  Open tempfile$ For Binary Access Write As #2
ss = "RIFF"
  Put #2, , ss
  ll = FileLength + 50
  Put #2, , ll
  ss = "WAVEfmt "
  Put #2, , ss
  ll = 18
  Put #2, , ll
  ii = 7
  Put #2, , ii
  ii = 1
  Put #2, , ii
  ll = 8000
  Put #2, , ll
  Put #2, , ll
  ii = 1
  Put #2, , ii
  ii = 8
  Put #2, , ii
  ii = 0
  Put #2, , ii
  ss = "fact"
  Put #2, , ss
```

```
  ll = 4
  Put #2, , ll
  ll = FileLength
  Put #2, , ll
  ss = "data"
  Put #2, , ss
  ll = FileLength
  Put #2, , ll
For i = 1 To (FileLength / 128)
   Get #1, , str_vox
   Put #2, , str_vox
      DoEvents
  Next i
Get #1, , str_vox
For i = 1 To (FileLength Mod 128)
    str_wav = Mid(str_vox, i, 1)
    Put #2, , str_wav
  Next i
Close #1
  Close #2
If isfile(szSrcFil) Then Kill szSrcFil
Name tempfile$ As szDstFil
  If isfile(tempfile$) Then Kill tempfile$
Exit Sub
errhandler:
End Sub
Function isfile%(ByVal Filename$)
'Returns True if  a file exists
  isfile% = (Dir$(Filename$) <> "")
End Function
Function GetRecFileName$(Mailbox$)
Dim Filename$, i%, NowTime As Date
On Error Resume Next
```

```
 MkDir AppDir & "mboxes\" & Mailbox$ & "\msg"
 On Error GoTo 0
i = 0
 Do
   NowTime = DateAdd("n", i, Now)
   Filename$ = AppDir & "mboxes\" & Mailbox$ & "\msg\" & Format$(NowTime, "mmddhhnn") &
".raw"
   i = i + 1
 Loop While isfile%(Filename$) Or isfile%(Left(Filename$, Len(Filename$) - 3) & "wav") Or isfile%
(Left(Filename$, Len(Filename$) - 3) & "inf") Or isfile%(Left(Filename$, Len(Filename$) - 3) & "in_")
Or isfile%(Left(Filename$, Len(Filename$) - 3) & "i__") 'Loop until a new file is found
   GetRecFileName$ = Filename$
Open Filename$ For Binary Access Write As #1
 Close #1
End Function
Sub ParseString(ByVal s$, p() As ParseType)
Dim i&, ss$
i& = 1
 While s$ <> ""
   If InStr(s$, ",") <> 0 Then
     ss$ = Left(s$, InStr(s$, ",") - 1)
     s$ = Right(s$, Len(s$) - InStr(s$, ","))
   Else
     ss$ = s$
     s$ = ""
   End If
   ss$ = Trim(ss$)
   If Left(ss$, 1) = Chr(34) Then
     ss$ = Right(ss$, Len(ss$) - 1)
     p(i&).quoted = True
```

```
    Else
      p(i&).quoted = False
    End If
    If Right(ss$, 1) = Chr(34) Then ss$ = Left(ss$, Len(ss$) - 1)
    p(i&).Parastr = ss$
    i& = i& + 1
  Wend
End Sub
```

ภาคผนวก ง

โปรแกรมที่ 4

```
#Const ext = False

#Const jmr = False
#Const c21 = False
#Const wbt = False

Option Explicit

Public Sub RefreshManager()

 RefreshMgr = True

End Sub

Public Sub AllFunctions(FunctionList$)

 FunctionList =
"DTMFMenu,DoTransfer,PlayPrompt,SelectFax,SendFax,DoRemote,GetHours,CIDMenu,DIDMenu,DI
DMenu2,SetLanguage,GetInput,GetRecording,DoVerify,AddRecord,EditRecord,EvalField,SetCounter,D
oScript,Initialize,ForceHangup"

#If text Then    '#############################################################################
 FunctionList = FunctionList & ",PlayRanPrompt,PlayClassifiedAd,CreateClassifiedAd"
 FunctionList = FunctionList & ",GoBySequence"
 FunctionList = FunctionList & ",DoDateVerify,EvalDateTime,DoCallNumVerify,SchedAlarm"
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
End Sub

Public Sub GetFunction(FunctionName$, PropertyList$, FunctionHelp$, ConditionList$,
ConditionHelp$)

 Select Case FunctionName

 Case "DTMFMenu"
  PropertyList = "Prompt,InvalidPrompt,Attempts,MaxDigits,--[Advanced]--
,InputVariable,DtmfTimeout,LogInput,Check1stDigit,MinDigits,2ndInvalidPrompt,3rdInvalidPrompt"
  FunctionHelp = "DTMF Menu - present a menu of DTMF input choices for the caller."
  ConditionList = "DTMF,Invalid,NoInput,FaxTone,MinDigits"
  ConditionHelp = "Tones can be up to 20 digits. DTMF Menu conditions - (Invalid) invalid input after
max Attempts, (NoInput) no input on any attempt, (MinDigits) at least MinDigits digits received,
(FaxTone) incoming fax CNG tone detected. Note: Do not include the (NoInput) condition to treat 'no
input' like an invalid input."

 Case "CIDMenu"
  PropertyList = ""
  FunctionHelp = "Caller ID Menu - present a menu of Caller ID values."
  ConditionList = "DTMF,N/A,~"
  ConditionHelp = "Caller ID 'tones' should be 3, 4, 5, 6, 7 or 10 digits - Caller ID Menu conditions -
(N/A) caller ID not available, (~) caller ID not found in menu or otherwise."

 Case "ForceHangup"
  PropertyList = ""
  FunctionHelp = "Force Hangup - Force hangup, without exit call procedures."
  ConditionList = ""
  ConditionHelp = "Force Hangup - Do not add any conditions to this function."

 Case "DIDMenu"
  PropertyList = "Attempts,MaxDigits,DIDTimeout,--[Advanced]--,InputVariable,LogInput,MinDigits"
  FunctionHelp = "DID Menu - present a menu of DID values."
```

ConditionList = "DTMF,Invalid,NoInput,FaxTone,MinDigits"

ConditionHelp = "Tones can be up to 20 digits. DTMF Menu conditions - (Invalid) invalid input after max Attempts, (NoInput) no input on any attempt, (MinDigits) at least MinDigits digits received, (FaxTone) incoming fax CNG tone detected. Note: Do not include the (NoInput) condition to treat 'no input' like an invalid input."

Case "DIDMenu2"

PropertyList = "DIDSettings,Attempts,MaxDigits,DIDTimeout,--[Advanced]--,InputVariable,LogInput,MinDigits"

FunctionHelp = "DID Menu - present a menu of DID values."

ConditionList = "DTMF,Invalid,NoInput,FaxTone,MinDigits"

ConditionHelp = "Tones can be up to 20 digits. DTMF Menu conditions - (Invalid) invalid input after max Attempts, (NoInput) no input on any attempt, (MinDigits) at least MinDigits digits received, (FaxTone) incoming fax CNG tone detected. Note: Do not include the (NoInput) condition to treat 'no input' like an invalid input."

Case "GetHours"

PropertyList = ""

FunctionHelp = "Hours Menu - a menu of office hours."

ConditionList = "Open,Closed,Lunch,Holiday,~"

ConditionHelp = "Get Hours conditions - (Open) open hours, (Closed) closed hours, (Lunch) lunch hours, (Holiday) holiday, (~) otherwise. Note: only the first 4 conditions really exist. Use otherwise if you do not want to state every condition."

Case "DoTransfer"

PropertyList = "Prompt,PersonalPrompt,TransferSettings,--[Advanced]--,BusyPrompt,NoAnswerPrompt"

FunctionHelp = "Do Transfer - transfers the call and takes a message if transfer is not completed."

ConditionList = "~,Busy,NoAnswer"

ConditionHelp = "Do Transfer conditions - (~) go to mailbox after completing function, (Busy) busy, (NoAnswer) no answer. Note: Do not use Busy and NoAnswer if you want to continue to taking a message in the mailbox."

```
Case "DoRemote"
  PropertyList = ""
  FunctionHelp = "Do Remote - Remote control to hear messages and to set options."
  ConditionList = "~"
  ConditionHelp = "Do Remote conditions - (~) go to mailbox."

Case "GetInput"
  PropertyList = "Prompt,InvalidPrompt,InputVariable,Attempts,MaxDigits,MinDigits,Verify,--
[Advanced]--,DtmfTimeout,LogInput,2ndInvalidPrompt,3rdInvalidPrompt"
  FunctionHelp = "Get Input - get DTMF input from the caller and store in InputVariable."
  ConditionList = "Invalid,~"
  ConditionHelp = "Get Input conditions - (Invalid) invalid input after max Attempts, (~) valid input."

Case "GetRecording"
  PropertyList = "Prompt,Verify,MaxTime"
  FunctionHelp = "Get Recording - get caller to record to a voice file."
  ConditionList = "~"
  ConditionHelp = "Get Recording conditions -  (~) go to mailbox."

Case "PlayPrompt"
  PropertyList = "Prompt"
  FunctionHelp = "Play Prompt - plays the Prompt file."
  ConditionList = "~"
  ConditionHelp = "Play Prompt conditions - (~) go to mailbox."

Case "DoScript"
  PropertyList =
"Script,Prompt1,Prompt2,Prompt3,Prompt4,Prompt5,Prompt6,Prompt7,Prompt8,Prompt9"
  FunctionHelp = "Do Script - executes the Script commands."
  ConditionList = "~"
  ConditionHelp = "Do Script conditions - (~) go to mailbox."

Case "SetCounter"
```

```
  PropertyList = "CounterVariable,CounterAction,MaxCount"
  FunctionHelp = "Set Counter - resets or increments the CounterVariable until MaxCount."
  ConditionList = "MaxCount,~"
  ConditionHelp = "Set Counter conditions - (MaxCount) counter has reached MaxCount, (~) counter has
not reached MaxCount."

 Case "SetLanguage"
  PropertyList = "Language"
  FunctionHelp = "Set Language - set the file extension of this language.
  ConditionList = "~"
  ConditionHelp = "Set Language conditions - (~) go to mailbox."

 Case "DoVerify"
  PropertyList = "DataSource,FindField,FindVariable,MatchField,MatchVariable,InvalidPrompt"
  FunctionHelp = "Do Verify - verifies that a record contains FindVariable in FindField. Also can check
if MatchVariable matches MatchField."
  ConditionList = "Invalid,~"
  ConditionHelp = "Do Verify conditions - (Invalid) FindVariable not found or MatchVariable not valid,
(~) there is match."

 Case "AddRecord"
  PropertyList = "DataSource,Script,InvalidPrompt"
  FunctionHelp = "Add Record - adds record to the table specified by DataSource."
  ConditionList = "Invalid,~"
  ConditionHelp = "Add Record conditions - (Invalid) Record can not be added, (~) Record added or
goto mailbox if alone."

 Case "EditRecord"
  PropertyList = "DataSource,FindField,FindVariable,Script,InvalidPrompt"
  FunctionHelp = "Edit Record - edits record in the table specified by DataSource."
  ConditionList = "Invalid,~"
  ConditionHelp = "Edit Record conditions - (Invalid) Record can not be found, (~) Record edited or
goto mailbox if alone."
```

```
    Case "EvalField"
    PropertyList = "DataSource,FindField,FindVariable,MatchField,MatchValue,InvalidPrompt"
    FunctionHelp = "Evaluate Field - evaluate a field in a record in the table specified by DataSource."
    ConditionList = "Invalid,>,<,=,~"
    ConditionHelp = "Evaluate Field conditions - (Invalid) FindVariable not found, (>) Value of
MatchField greater than MatchValue, (<), (=), (~) Otherwise."

  Case "SelectFax"
    PropertyList = "Prompt,FaxFile"
    FunctionHelp = "Select Fax - selects FaxFile for fax-on-demand."
    ConditionList = "~,MaxFax"
    ConditionHelp = "Select Fax conditions - (~) function successful, (MaxFax) function successful, but
maximun limit of faxes reached."

  Case "SendFax"
    PropertyList = "FaxMethod,FaxCover,FaxFrom,MaxFax,Attempts"
    FunctionHelp = "Send Fax - sends off the selected faxes."
    ConditionList = "~,NoFax"
    ConditionHelp = "Select Fax conditions - (~) function successful, (NoFax) no fax selected."

' Case "OpenData"
'   ParaList = "Database,Table,DataSource"
' Case "SaveRecord"
'   ParaList = "CommandString,DataSource"
' Case "PlayInputs"
'   ParaList = "CommandString"

  Case "Initialize"
    PropertyList = "Script"
    FunctionHelp = "Initialize - Script is executed when program is started. There are no branch
conditions."
    ConditionList = ""
    ConditionHelp = "Initialize - Do not add any conditions to this function."
```

```
#If ext Then
'#########################################################################
  Case "PlayRanPrompt"
    PropertyList = "MaxCount"
    FunctionHelp = "Play Random Prompt - plays the Random Prompt file. (from 1.wav to
MaxCount.wav)"
    ConditionList = "~"
    ConditionHelp = "Play Random Prompt conditions - (~) go to mailbox."

  Case "PlayClassifiedAd"
    PropertyList = "PersonalPrompt,TransferSettings"
    FunctionHelp = "Play Classified Ad - plays the Classified Ad personal prompt and options."
    ConditionList = "Next"
    ConditionHelp = "Play Classified Ad conditions - (Next) next mailbox of category."

  Case "CreateClassifiedAd"
    PropertyList = "Prompt1,Prompt2,Prompt3,Prompt4,Prompt5"
    FunctionHelp = "Create Classified Ad - creates Classified Ad mailbox."
    ConditionList = "~"
    ConditionHelp = "Create Classified Ad conditions - (~) go to mailbox."

  Case "GoBySequence"
    PropertyList = "FirstBox,LastBox,StartBox"
    FunctionHelp = "Goes sequentially from FirstBox to LastBox, starting from StartBox."
    ConditionList = ""
    ConditionHelp = "Go By Sequence - Do not add any conditions to this function."

  Case "DoDateVerify"
    PropertyList = "DateVariable,TimeVariable,InvalidPrompt"
    FunctionHelp = "Do Date Verify - verifies that the date and/or time is valid."
    ConditionList = "Invalid,~"
    ConditionHelp = "Do Date Verify conditions - (Invalid) date and/or time is not valid, (~) valid."
```

```
 Case "EvalDateTime"
   PropertyList = "DateVariable,TimeVariable,AMPMVariable,InvalidPrompt"
   FunctionHelp = "Evaluate Date Time - evaluate date and time."
   ConditionList = "Invalid,>,~"
   ConditionHelp = "Evaluate Date Time conditions - (Invalid) date and time is invalid, (>) date and time
is within 100 days from now, (~) otherwise."

 Case "DoCallNumVerify"
   PropertyList = "CallNumVariable,InvalidPrompt"
   FunctionHelp = "Do Call Number Verify - verifies that the number to call is valid."
   ConditionList = "Forbidden,~"
   ConditionHelp = "Do Call Number Verify conditions - (Invalid) number to call forbidden - see
Konnects.ini, (~) valid."

#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

#If wbt Then
'##############################################################################
 Case "WTTransfer"
   PropertyList = "TransferSettings"
   FunctionHelp = "WT Transfer - transfers the call for WT."
   ConditionList = "Busy"
   ConditionHelp = "WT Transfer conditions - (Busy) line is busy."

 Case "1492Transfer"
   PropertyList = "TransferSettings"
   FunctionHelp = "1492 Transfer - transfers the call for 1492."
   ConditionList = "Busy"
   ConditionHelp = "1492 Transfer conditions - (Busy) line is busy."
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

```
#If jmr Then
'##########################################################################
 Case "JMRTransfer"
  PropertyList = "Prompt,TransferSettings"
  FunctionHelp = "JMR Transfer - transfers the call for JMR."
  ConditionList = "NoAns,Busy"
  ConditionHelp = "JMR Transfer conditions - (NoAns) no answer, (Busy) line is busy."
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

#If c21 Then
'##########################################################################
 Case "21stCapGetFax"
  PropertyList = ""
  FunctionHelp = "21stCap Get Fax - determines if record contains fax number."
  ConditionList = "Yes,No"
  ConditionHelp = "21stCap Get Fax conditions - (Yes) record has fax number, (No) record does not
have fax number."
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

 Case Else
  PropertyList =
"Attempts,MaxDigits,MinDigits,CounterVariable,CounterAction,MaxCount,InputVariable,DataSource,Fi
ndVariable,MatchVariable,Prompt,InvalidPrompt,2ndInvalidPrompt,3rdInvalidPrompt,FaxFile,Script"
  FunctionHelp = "No information on this function."
  ConditionList = ""
  ConditionHelp = "No information on this function."

 End Select

End Sub
```

```
Public Sub AllProperties(PropertyList$)

 PropertyList =
"Attempts,MaxDigits,MinDigits,CounterVariable,CounterAction,MaxCount,MaxTime,InputVariable,Dat
aSource,FindField,FindVariable,MatchField,MatchValue,MatchVariable,Prompt,PersonalPrompt,BusyPr
ompt,NoAnswerPrompt,InvalidPrompt,2ndInvalidPrompt,3rdInvalidPrompt,FaxFile,Script,FaxMethod,Fa
xCover,MaxFax,TransferSettings,Language,DtmfTimeout,DIDTimeout,FaxFrom,Verify,Check1stDigit,L
ogInput,DIDSettings,Prompt1,Prompt2,Prompt3,Prompt4,Prompt5,Prompt6,Prompt7,Prompt8,Prompt9,F
irstBox,LastBox,StartBox,DateVariable,TimeVariable,AMPMVariable,CallNumVariable"

End Sub

Public Sub GetProperty(PropName$, PropType$, PropInfo$, PropHelp$, PropDataType$,
PropDataSize%)

 Select Case PropName

 Case "Attempts"
  PropType = "Combo"
  PropInfo = "1,2,3,4,5,6,7,8,9,10"
  PropHelp = "Attempt - Number of attempts to get DTMF or input."

  PropDataType = "Integer"

 Case "MaxDigits"
  PropType = "Combo"
  PropInfo = "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21"
  PropHelp = "Max Digits - Maximum number of DTMF's to wait for."

  PropDataType = "Integer"

 Case "MinDigits"
  PropType = "Combo"
```

```
    PropInfo = "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21"
    PropHelp = "Min Digits - Minimum required DTMF's for input."

    PropDataType = "Integer"

  Case "DtmfTimeout"
    PropType = "Combo"
    PropInfo = "0,1,2,3,4,5,6,7,8,9,10,12"
    PropHelp = "Dtmf Timeout - Seconds to wait for DTMF input."

    PropDataType = "Integer"

  Case "DIDTimeout"
    PropType = "Combo"
    PropInfo = "0,100,200,300,400,500,600,700,800,900,1000,2000"
    PropHelp = "Dtmf Timeout - Milliseconds to wait for DTMF input."

    PropDataType = "Integer"

' Case "1stDigitTimeout"
'   PropType = "Combo"
'   PropInfo = "0,100,200,300,400,500,600,700,800,900,1000,2000"
'   PropHelp = "1stDigitTimeout - Milliseconds to wait for the first digit of DTMF input."

'   PropDataType = "Integer"

  Case "CounterVariable"
    PropType = "Combo"
    PropInfo = "Counter0,Counter1,Counter2,Counter3,Counter4,Counter5"
    PropHelp = "Counter Variable - Select a counter variable."

    PropDataType = "Text"
    PropDataSize = 20
```

```
Case "CounterAction"
PropType = "Combo"
PropInfo = "Reset,Increment"
PropHelp = "Counter Action - Set an action for this counter."

PropDataType = "Text"
PropDataSize = 20

Case "MaxCount"
PropType = "Combo"
PropInfo = "1,2,3,4,5,6,7,8,9,10"
PropHelp = "Max Count - Maximum count to set the condition of MaxCount."

PropDataType = "Integer"

Case "MaxTime"
PropType = "Text"
PropInfo = ""
PropHelp = "Max Time - Maximum seconds to perform function."

PropDataType = "Integer"

Case "InputVariable"
PropType = "Combo"
PropInfo = "Input0,Input1,Input2,Input3,Input4,Input5,Input6,Input7,Input8,Input9"
PropHelp = "Input Variable - Select a variable to store the input."

PropDataType = "Text"
PropDataSize = 20

Case "DataSource"
PropType = "Combo"
PropInfo = "Data0,Data1,Data2,Data3,Data4,Data5,Data6,Data7,Data8,Data9"
```

```
  PropHelp = "Data Source - Table specified by the CreateDynaset script command in the Initialize
mailbox."

  PropDataType = "Text"
  PropDataSize = 20

 Case "FindField"
  PropType = "Text"
  PropInfo = ""
  PropHelp = "Find Field - Field in table to search."

  PropDataType = "Text"
  PropDataSize = 25

 Case "FindVariable"
  PropType = "Combo"
  PropInfo = "Input0,Input1,Input2,Input3,Input4,Input5,Input6,Input7,Input8,Input9,CallerID$"
  PropHelp = "Find Variable - Value to search. Can be an Input Variable or a fixed value."

  PropDataType = "Text"
  PropDataSize = 20

 Case "MatchField"
  PropType = "Text"
  PropInfo = ""
  PropHelp = "Match Field - Field in table to match."

  PropDataType = "Text"
  PropDataSize = 25

 Case "MatchValue"
  PropType = "Text"
  PropInfo = ""
```

```
PropHelp = "Match Value - Value to Match against."

PropDataType = "Text"
PropDataSize = 25

Case "MatchVariable"
PropType = "Combo"
PropInfo = "Input0,Input1,Input2,Input3,Input4,Input5,Input6,Input7,Input8,Input9,CallerID$"
PropHelp = "Match Variable - Value to match. Can be an Input Variable or a fixed value."

PropDataType = "Text"
PropDataSize = 20

Case "Prompt"
PropType = "VoiceFile"
PropInfo = "prompt.wav"
PropHelp = "Prompt - The main prompt of the mailbox."

PropDataType = "Text"
PropDataSize = 50

Case "Prompt1"
PropType = "VoiceFile"
PropInfo = "prompt1.wav"
PropHelp = "Prompt1 - Prompt of the mailbox."

PropDataType = "Text"
PropDataSize = 50

Case "Prompt2"
PropType = "VoiceFile"
PropInfo = "prompt2.wav"
PropHelp = "Prompt2 - Prompt of the mailbox."
```

```
    PropDataType = "Text"
    PropDataSize = 50

  Case "Prompt3"
    PropType = "VoiceFile"
    PropInfo = "prompt3.wav"
    PropHelp = "Prompt3 - Prompt of the mailbox."

    PropDataType = "Text"
    PropDataSize = 50

  Case "Prompt4"
    PropType = "VoiceFile"
    PropInfo = "prompt4.wav"
    PropHelp = "Prompt4 - Prompt of the mailbox."

    PropDataType = "Text"
    PropDataSize = 50

  Case "Prompt5"
    PropType = "VoiceFile"
    PropInfo = "prompt5.wav"
    PropHelp = "Prompt5 - Prompt of the mailbox."

    PropDataType = "Text"
    PropDataSize = 50

  Case "Prompt6"
    PropType = "VoiceFile"
    PropInfo = "prompt6.wav"
    PropHelp = "Prompt6 - Prompt of the mailbox."

    PropDataType = "Text"
```

```
  PropDataSize = 50

Case "Prompt7"
  PropType = "VoiceFile"
  PropInfo = "prompt7.wav"
  PropHelp = "Prompt7 - Prompt of the mailbox."

  PropDataType = "Text"
  PropDataSize = 50

Case "Prompt8"
  PropType = "VoiceFile"
  PropInfo = "prompt8.wav"
  PropHelp = "Prompt8 - Prompt of the mailbox."

  PropDataType = "Text"
  PropDataSize = 50

Case "Prompt9"
  PropType = "VoiceFile"
  PropInfo = "prompt9.wav"
  PropHelp = "Prompt9 - Prompt of the mailbox."

  PropDataType = "Text"
  PropDataSize = 50

Case "PersonalPrompt"
  PropType = "VoiceFile"
  PropInfo = "personal.wav"
  PropHelp = "PersonalPrompt - Prompt containing your personal greeting."

  PropDataType = "Text"
  PropDataSize = 50
```

```
  Case "BusyPrompt"
  PropType = "VoiceFile"
  PropInfo = "busy.wav"
  PropHelp = "BusyPrompt - Prompt containing your busy personal greeting."

  PropDataType = "Text"
  PropDataSize = 50

Case "NoAnswerPrompt"
  PropType = "VoiceFile"
  PropInfo = "NoAnswer.wav"
  PropHelp = "NoAnswerPrompt - Prompt containing your no answer personal greeting."

  PropDataType = "Text"
  PropDataSize = 50

Case "InvalidPrompt"
  PropType = "VoiceFile"
  PropInfo = "invalid.wav"
  PropHelp = "Invalid Prompt - Prompt to play when the input is invalid."

  PropDataType = "Text"
  PropDataSize = 50

Case "2ndInvalidPrompt"
  PropType = "VoiceFile"
  PropInfo = "invalid2.wav"
  PropHelp = "2nd Invalid Prompt - Prompt to play on the 2nd invalid input."

  PropDataType = "Text"
  PropDataSize = 50

Case "3rdInvalidPrompt"
```

```
  PropType = "VoiceFile"
  PropInfo = "invalid3.wav"
  PropHelp = "3rd Invalid Prompt - Prompt to play on the 3rd invalid input."

  PropDataType = "Text"
  PropDataSize = 50

 Case "FaxFile"
  PropType = "FaxFile"
  PropInfo = ""
  PropHelp = "Fax File - Fax file for fax-on-demand."

  PropDataType = "Text"
  PropDataSize = 50

 Case "TransferSettings"
  PropType = "IniFile"
  PropInfo = "transfer.ini"
  PropHelp = "Transfer Settings File - Initialization file for transfer settings."

  PropDataType = "Text"
  PropDataSize = 20

 Case "DIDSettings"
  PropType = "IniFile"
  PropInfo = "did.ini"
  PropHelp = "DID Settings File - Initialization file for DID settings."

  PropDataType = "Text"
  PropDataSize = 20

 Case "FaxCover"
  PropType = "FaxFile"
```

```
  PropInfo = ""
  PropHelp = "Fax Cover - Fax cover for 2-call fax-on-demand."

  PropDataType = "Text"
  PropDataSize = 50

Case "FaxFrom"
  PropType = "Text"
  PropInfo = ""
  PropHelp = "Fax From - Sender name to appear at the top of fax."

  PropDataType = "Text"
  PropDataSize = 50

Case "MaxFax"
  PropType = "Combo"
  PropInfo = "1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19,20,21"
  PropHelp = "Max Fax - Maximum number of fax documents than can be selected."

  PropDataType = "Integer"

Case "FaxMethod"
  PropType = "Combo"
  PropInfo = "1CallOnly,2CallOnly,Both"
  PropHelp = "Fax Method - Select allowable Fax-on-Demand methods."

  PropDataType = "Text"
  PropDataSize = 20

Case "Script"
  PropType = "TextBox"
  PropInfo = ""
  PropHelp = "Script - Script with commands to execute."
```

```
    PropDataType = "Memo"

Case "Language"
    PropType = "Combo"
    PropInfo = "WAV,WA1,WA2,WA3,WA4,WA5,WA6,WA7,WA8,WA9"
    PropHelp = "Language - Select an extension for this language."

    PropDataType = "Text"
    PropDataSize = 20

Case "Verify"
    PropType = "Combo"
    PropInfo = "Yes,No"
    PropHelp = "Verify - Select whether to verify input."

    PropDataType = "Text"
    PropDataSize = 20

Case "Check1stDigit"
    PropType = "Combo"
    PropInfo = "Yes,No"
    PropHelp = "Check1stDigit - Select whether to check if first digit is in Tones & Conditions."

    PropDataType = "Text"
    PropDataSize = 20

Case "LogInput"
    PropType = "Combo"
    PropInfo = "Yes,No"
    PropHelp = "LogInput - Select whether to log caller input into the Call Log."

    PropDataType = "Text"
    PropDataSize = 20
```

```
  Case "FirstBox"
  PropType = "Text"
  PropInfo = ""
  PropHelp = "First Box - First Mailbox."

  PropDataType = "Text"
  PropDataSize = 20

 Case "LastBox"
  PropType = "Text"
  PropInfo = ""
  PropHelp = "Last Box - Last Mailbox."

  PropDataType = "Text"
  PropDataSize = 20

 Case "StartBox"
  PropType = "Text"
  PropInfo = ""
  PropHelp = "Start Box - Starting Mailbox."

  PropDataType = "Text"
  PropDataSize = 20

 Case "DateVariable"
  PropType = "Combo"
  PropInfo = "Input0,Input1,Input2,Input3,Input4,Input5,Input6,Input7,Input8,Input9"
  PropHelp = "Date Variable - Select a variable to store the date."

  PropDataType = "Text"
  PropDataSize = 20

 Case "TimeVariable"
```

```
    PropType = "Combo"
    PropInfo = "Input0,Input1,Input2,Input3,Input4,Input5,Input6,Input7,Input8,Input9"
    PropHelp = "Time Variable - Select a variable to store the time."

    PropDataType = "Text"
    PropDataSize = 20

  Case "AMPMVariable"
    PropType = "Combo"
    PropInfo = "Input0,Input1,Input2,Input3,Input4,Input5,Input6,Input7,Input8,Input9"
    PropHelp = "AMPM Variable - Select a variable to store 1 for AM and 2 for PM"

    PropDataType = "Text"
    PropDataSize = 20

  Case "CallNumVariable"
    PropType = "Combo"
    PropInfo = "Input0,Input1,Input2,Input3,Input4,Input5,Input6,Input7,Input8,Input9"
    PropHelp = "Time Variable - Select a variable to store the time."

    PropDataType = "Text"
    PropDataSize = 20

  Case Else
    PropType = ""
    PropInfo = ""
    PropHelp = "No information on this property."

    PropDataType = ""

  End Select

End Sub
```

ภาคผนวก จ

โปรแกรมที่ 5 Main Program

```
#Const ext = False
#Const jmr = False
#Const c21 = False
#Const wbt = False

Option Explicit

    Dim VMailDB As Database     'The MS Access database variable
    Dim Mailboxes As Dynaset    'The mailbox dynaset
    Dim Tonetables As Dynaset
    Dim Messages As Dynaset
    Dim ScheduleDB As Database
    'Dim Schedule As Dynaset
    'Dim SchedLog As Dynaset
    Dim Schedule As Recordset
    Dim SchedLog As Recordset
    Dim db(0 To 9) As Database
    'Dim ds(0 To 9) As Dynaset
    Dim ds(0 To 9) As Recordset
    Dim ds_string$(0 To 9)
    Dim ds_dbi%(0 To 9)
    Dim CurrentFunction$(1 To 74)
    Dim PreviousFunction$(1 To 74)
    Dim DIDinfo$(1 To 74)
    Dim counter%(1 To 74), counter2%(1 To 74)
    Dim temp$(1 To 74), temp1$(1 To 74), temp2$(1 To 74)
    Dim FaxInfo$(1 To 74), FaxDocs%(1 To 74), FaxLines%(1 To 10), NumFax%
    Dim Inp$(1 To 74, 0 To 9)
    Dim ScheduleID&(1 To 74)
    Dim VoiceLines%
```

```
        Dim Queue(1 To 74) As QueueType
        Dim lineTime(1 To 74) As Date
        Dim RecFileName$(1 To 74)
        Dim SeekPos&(1 To 74)
        Dim MaxFaxDoc%

        Public Sub VserverSetup(ApplicationDir$, VLines%, FLines%(), NFax%)

        Dim s$, ss$, sss$, func$, dbi%, dsi%, res%, retstr As String * 255, i%

        VoiceLines% = VLines%
        AppDir$ = ApplicationDir$
        NumFax% = NFax%
For i = 1 To 10
        FaxLines%(i) = FLines%(i)
Next i

        ' Set VMailDB = OpenDatabase(AppDir & "vmail.mdb")
         Set VMailDB = Workspaces(0).OpenDatabase(AppDir & "vmail.mdb")
        Set Mailboxes = VMailDB.OpenRecordset("MailBoxes", dbOpenDynaset)
         Set Tonetables = VMailDB.OpenRecordset("Tonetables", dbOpenDynaset)
        ' Set Messages = VMailDB.CreateDynaset("Messages")
        RefreshMgr = False

        Set ScheduleDB = Workspaces(0).OpenDatabase(AppDir & "schedule.mdb")
         Set Schedule = ScheduleDB.OpenRecordset("Schedule", dbOpenDynaset) ' order by
TimeScheduled")
         Set SchedLog = ScheduleDB.OpenRecordset("SchedLog", dbOpenDynaset)
        RefreshSched = False

 Schedule.FindFirst "ScheduleID <> Null"
 While Not Schedule.NoMatch
```

```
        If Schedule("Status") <> "Pending" And Schedule("Status") <> "Deleted" And Schedule
("Status") <> "Edited" Then
    Schedule.Edit
    Schedule("Status") = "Pending"
    Schedule.Update
   End If
   Schedule.FindNext "ScheduleID <> Null"
  Wend
  FreeLocks


        Mailboxes.FindFirst "Function = 'Initialize'"
  If Not Mailboxes.NoMatch Then
        s$ = "" & Mailboxes("Script")
   While s$ <> ""
        If InStr(s$, Chr(10)) <> 0 Then
        ss$ = Left(s$, InStr(s$, Chr(10)) - 1)
        s$ = Right(s$, Len(s$) - InStr(s$, Chr(10)))
    Else
        s$ = s$
        s$ = ""
    End If
     If InStr(ss$, "(") <> 0 And InStr(ss$, ")") <> 0 Then
        func$ = Left(ss$, InStr(ss$, "(") - 1)
        ss$ = Left(ss$, InStr(ss$, ")") - 1)
        ss$ = Right(ss$, Len(ss$) - InStr(ss$, "("))

Select Case func$
      Case "OpenDatabase"
       dbi% = Val(Mid(ss$, InStr(ss$, ",") - 1, 1))
       ss$ = Right(ss$, Len(ss$) - InStr(ss$, ","))
       sss$ = Left(ss$, InStr(ss$, ",") - 1)
      Set db(dbi%) = Workspaces(0).OpenDatabase(sss$, False, False, Right(ss$, Len(ss$) - InStr(ss$, ",")))
      Case "CreateDynaset"
```

```
        dbi% = Val(Mid(ss$, InStr(ss$, ",") - 1, 1))
        ss$ = Right(ss$, Len(ss$) - InStr(ss$, ","))
        dsi% = Val(Mid(ss$, InStr(ss$, ",") - 1, 1))
        ds_string$(dsi%) = Right(ss$, Len(ss$) - InStr(ss$, ","))
        ds_dbi%(dsi) = dbi%
        Set ds(dsi%) = db(dbi%).OpenRecordset(Right(ss$, Len(ss$) - InStr(ss$, ",")), dbOpenDynaset)

    '   ds(dsi%).CacheSize = 0

'@@c2102@@

      Case Else

      End Select

    End If
   Wend
  End If

  Mailboxes.FindFirst "Function = 'SendFax'"
  MaxFaxDoc% = 20
  If Not Mailboxes.NoMatch Then
   If Not IsNull(Mailboxes("MaxFax")) Then MaxFaxDoc% = Mailboxes("MaxFax")
  End If

End Sub

Public Sub OleLink(Subject As String, Data(), Log As String, CallLog As String)

  Dim i%, lineNo%, Mailbox$, digits$, callstate%, CallerID$
  Dim res%, retstr As String * 255
  Dim mStateS, s$
```

```
For i = 0 To 4
 OleData(i) = Data(i)
Next i

lineNo% = OleData(0)
Mailbox$ = OleData(1)
digits$ = OleData(2)
callstate% = OleData(3)
CallerID$ = OleData(4)

FilePlaying$(lineNo%) = ""

If RefreshMgr = True Then
 Set Mailboxes = VMailDB.OpenRecordset("MailBoxes", dbOpenDynaset)
 Set Tonetables = VMailDB.OpenRecordset("Tonetables", dbOpenDynaset)
 RefreshMgr = False
End If

If Subject = "Start" Then
 PreviousFunction$(lineNo%) = CurrentFunction$(lineNo%)
 Mailboxes.FindFirst "MailboxNumber='" & Mailbox$ & "'"
 If Not Mailboxes.NoMatch Then
  If IsNull(Mailboxes("Function")) Then
   CurrentFunction$(lineNo%) = "NullFunction"
  Else
   CurrentFunction$(lineNo%) = Mailboxes("Function")
  End If
 Else
  CurrentFunction$(lineNo%) = "NoMailbox"
 End If
 State$(lineNo%) = ""
 substate$(lineNo%) = ""
ElseIf Subject = "Schedule" Or Subject = "ScheduleFax" Or Subject = "Exit" Or Subject = "Reset" Then
```

```
    PreviousFunction$(lineNo%) = CurrentFunction$(lineNo%)
    CurrentFunction$(lineNo%) = Subject
    State$(lineNo%) = ""
    substate$(lineNo%) = ""
  End If

  If State$(lineNo%) = "" Then
    Log = "(Start)"
  Else
    Log = "(" & State$(lineNo%) & ") "
  End If
  OleLog = ""
  OleCallLog = ""

JumpLabel:

  Select Case CurrentFunction$(lineNo%)
  Case "DTMFMenu"
    fDTMFMenu
  Case "CIDMenu"
    fCIDMenu
  Case "DIDMenu"
    fDIDMenu
  Case "DIDMenu2"
    fDIDMenu2
  Case "DoTransfer"
'   Debug.Print state$(lineNo%)
    fDoTransfer
  Case "DoRemote"
    fDoRemote
  Case "SelectFax"
    fSelectFax
  Case "GetHours"
```

```
    fGetHours
  Case "GetInput"
    fGetInput
  Case "GetRecording"
    fGetRecording
  Case "DoVerify"
    fDoVerify
  Case "AddRecord"
    fAddRecord
  Case "EditRecord"
    fEditRecord
  Case "EvalField"
    fEvalField
  Case "DoScript"
    fDoScript
  Case "PlayPrompt"
    fPlayPrompt
  Case "SetCounter"
    fSetCounter
  Case "SetLanguage"
    fSetLanguage
  Case "SendFax"
    fSendFax
  Case "Exit"
    ExitCall
  Case "Schedule"
    ScheduleCall
  Case "ScheduleFax"
    ScheduleFax

#If ext Then
'#########################################################################
  Case "PlayRanPrompt"
```

```
  extPlayRanPrompt
 Case "PlayClassifiedAd"
  extPlayClassifiedAd
 Case "CreateClassifiedAd"
  extCreateClassifiedAd
 Case "GoBySequence"
  extGoBySequence
 Case "EvalDateTime"
  extEvalDateTime
 Case "DoCallNumVerify"
  extDoCallNumVerify
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

 Case "Reset"
'  OleLog = "Reset line..."
'  Queue(lineNo%).Mailbox = ""
'  FaxInfo$(lineNo%) = ""
'  FaxDocs%(lineNo%) = 0
'  Language%(lineNo%) = 0
'  For i = 0 To 9
'    Inp(lineNo%, i) = ""
'  Next i
  If isfile(App.Path & "\sysvox\temp" & lineNo% & ".raw") Then
   On Error Resume Next
   Kill App.Path & "\sysvox\temp" & lineNo% & ".raw"
  End If
'-----------
  If RecFileName$(lineNo%) <> "" Then
   If isfile(RecFileName(lineNo%)) Then
    OleCallLog = "Length " & FileLen(RecFileName$(lineNo%)) \ 8000 & " sec."
    If FileLen(RecFileName$(lineNo%)) < GetPrivateProfileInt("General", "MinMessageLength", 4,
App.Path & "\konnects.ini") * 8000& Then
```

```
        On Error Resume Next
        Kill RecFileName(lineNo%)
      Else
        Call CvtWaveFile(RecFileName$(lineNo%), Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "wav")
        If Not isfile(Left(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf") Then
          res% = WritePrivateProfileString("General", "PhoneNumber", "", Left(RecFileName$(lineNo%),
Len(RecFileName$(lineNo%)) - 3) & "inf")
          res% = WritePrivateProfileString("General", "CallerID", CallerID$, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf")
          res% = GetPrivateProfileString("General", "State", "", retstr$, Len(retstr$) - 1, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 12) & "status")
          mState$ = Str(Val(Left(retstr$, res%)) + 1)
          If mState$ > "30000" Then mState$ = "0"
          res% = WritePrivateProfileString("General", "State", mState$, Left(RecFileName$(lineNo%), Len
(RecFileName$(lineNo%)) - 12) & "status")
        End If

        res% = GetPrivateProfileString("Paging", "EnablePaging", "No", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
        If UCase(Left(retstr$, res%)) = "YES" Then
          If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then
            Schedule.AddNew
            Schedule("TimeScheduled") = DateAdd("s", 10, Now)
            Schedule("TimeInserted") = Now
            Schedule("Status") = "Pending"   'status$
            Schedule("Action") = "Page"
            res% = GetPrivateProfileString("Paging", "PhoneNumber", "", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
            Schedule("PhoneNumber") = Left(retstr$, res%)
            res% = GetPrivateProfileString("Paging", "PagerString", "", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
```

```
        s$ = Left(retstr$, res%)
        res% = GetPrivateProfileString("General", "PhoneNumber", "", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf")
        Schedule("PagerString") = s$ & "," & Left(retstr$, res%) & "#"
        Schedule("Attempts") = 3
        Schedule("Name") = ""
        Schedule("OwnerMailbox") = Mailbox$
        Schedule("AttemptsMade") = 0
        Schedule.Update
       End If
      End If

      res% = GetPrivateProfileString("Notification", "EnableNotification", "No", retstr$, Len(retstr$) - 1,
Left(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
      If UCase(Left(retstr$, res%)) = "YES" Then
       If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then
        Schedule.AddNew
        Schedule("TimeScheduled") = DateAdd("n", 5, Now)
        Schedule("TimeInserted") = Now
        Schedule("Status") = "Pending"   'status$
        Schedule("Action") = "Notify"
        res% = GetPrivateProfileString("Notification", "PhoneNumber", "", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
        Schedule("PhoneNumber") = Left(retstr$, res%)
        Schedule("Attempts") = 2
        Schedule("Name") = ""
        Schedule("OwnerMailbox") = Mailbox$
        Schedule("AttemptsMade") = 0
        Schedule.Update
       End If
      End If
```

```
res% = GetPrivateProfileString("Message Lamp", "EnableMessageLamp", "No", retstr$, Len
(retstr$) - 1, Left(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
If UCase(Left(retstr$, res%)) = "YES" Then
 If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then

 Schedule.FindFirst "OwnerMailbox = '" & Mailbox$ & "'  and Action = 'Lamp' and Status =
'Pending'"
 If Schedule.NoMatch Then Schedule.AddNew Else Schedule.Edit

 Schedule("TimeScheduled") = DateAdd("s", 10, Now)
 Schedule("TimeInserted") = Now
 Schedule("Status") = "Pending"   'status$
 Schedule("Action") = "Lamp"
 res% = GetPrivateProfileString("Message Lamp", "DialString", "", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
 Schedule("PhoneNumber") = Left(retstr$, res%)
 Schedule("Attempts") = GetPrivateProfileInt("Message Lamp", "Attempts", 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 16) & "transfer.ini")
 Schedule("Name") = ""
 Schedule("OwnerMailbox") = Mailbox$
 Schedule("AttemptsMade") = 0

 Schedule.Update
 End If
 End If

 End If
End If

 RecFileName$(lineNo%) = ""
End If
```

```
@@c2103@@

'@@jmr02@@

  OleLog = "Reset line..."
  Queue(lineNo%).Mailbox = ""
  FaxInfo$(lineNo%) = ""
  FaxDocs%(lineNo%) = 0
  Language%(lineNo%) = 0
  For i = 0 To 9
   Inp(lineNo%, i) = ""
  Next i

 Case "ForceHangup"
  OleLog = "force the hangup"
  Call vcfHangup

 Case "NoMailbox"
  OleLog = "mailbox does not exist, going to Hangup"
  Call vcfGoto("Hangup")

 Case "NullFunction"
  OleLog = "function not defined, going to Hangup"
  Call vcfGoto("Hangup")


#If jmr Then
'#############################################################################
 Case "JMRTransfer"
  fJMRTransfer
```

```
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
#If c21 Then
'################################################################################
 Case "21stCapGetFax"
  f21stCapGetFax
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
#If wbt Then
'################################################################################
 Case "WTTransfer"
  fWTTransfer
 Case "1492Transfer"
  f1492Transfer
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE



 Case Else
  OleLog = "function not programmed,"
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  If Not Tonetables.NoMatch Then
   Mailbox$ = Tonetables("Assignment")
   OleLog = OleLog & " going to " & Mailbox$
   Call vcfGoto(Mailbox$)
  Else
   OleLog = OleLog & " going to Hangup"
   Call vcfGoto("Hangup")
  End If
 End Select

 If Left(State$(lineNo%), 5) = "JUMP:" Then
```

```
  State$(lineNo%) = Right(State$(lineNo%), Len(State$(lineNo%)) - 5)
  GoTo JumpLabel
 End If

 If OleLog <> "" Then Log = Log & OleLog Else Log = ""
 CallLog = OleCallLog

 For i = 0 To 5
  Data(i) = OleData(i)
 Next i

End Sub

Private Sub ScheduleCall()

 Dim lineNo%, Mailbox$, digits$, callstate%, status$
 Dim HaveSchedule As Boolean
 Static Count%
 Dim res%, retstr As String * 255
 Static MBox$
 Dim Msg$, i%, PhoneNumber$, s$

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)
 callstate% = OleData(3)

 If State$(lineNo%) = "" Then
  State$(lineNo%) = "FindSchedule"
  counter%(lineNo%) = 0
 End If

 Select Case State$(lineNo%)
```

```
Case "FindSchedule"

HaveSchedule = False

If RefreshSched = True Then
 Set Schedule = ScheduleDB.OpenRecordset("Schedule", dbOpenDynaset) ' order by TimeScheduled"
 RefreshSched = False
Else
 Count% = Count% + 1
 If Count% > 300 Then
  Count% = 0
  Set Schedule = ScheduleDB.OpenRecordset("Schedule", dbOpenDynaset)
  Set SchedLog = ScheduleDB.OpenRecordset("SchedLog", dbOpenDynaset)
 End If
End If

Schedule.FindFirst "Action <> 'Fax'"  "'ScheduleID <> Null"
While Not Schedule.NoMatch And Not HaveSchedule
 If Schedule("Status") = "Deleted" Then
  SchedLog.AddNew
  CopySchedule
  SchedLog("Status") = "Deleted"
  SchedLog.Update
  Schedule.Delete
 ElseIf Schedule("Status") = "Edited" Then
  Schedule Delete
 ElseIf Schedule("TimeScheduled") < Now And Schedule("Status") = "Pending" Then
  HaveSchedule = True
 End If
 If Not HaveSchedule Then Schedule.FindNext "Action <> 'Fax'"
Wend

If HaveSchedule Then
```

```
    ScheduleID&(lineNo%) = Schedule("ScheduleID")
    If Schedule("Action") = "Lamp" Then

     OleLog = "schedule " & ScheduleID&(lineNo%) & " found, going off hook for message lamp "
     Call vcfPickup("DialLamp")

    Else
     res% = GetPrivateProfileString("General", "PageWithCPM", "", retstr$, Len(retstr$) - 1, App.Path &
"\konnects.ini")
     If Schedule("Action") = "Page" And UCase(Left(retstr$, res%)) = "NO" Then
      OleLog = "schedule " & ScheduleID&(lineNo%) & " found, going off hook for paging "
      Call vcfPickup("DialPager")


     Else
      res% = GetPrivateProfileString("General", "DialingPrefix", "", retstr$, Len(retstr$) - 1, App.Path &
"\konnects.ini")
      OleLog = "schedule " & ScheduleID&(lineNo%) & " found, calling " & Left(retstr$, res%) &
Schedule("PhoneNumber")
      Call vcfCallOut("Goto", Left(retstr$, res%) & Schedule("PhoneNumber"), 4)
     End If
    End If
    Schedule.Edit
    Schedule("Status") = "calling"
    Schedule("AttemptsMade") = Schedule("AttemptsMade") + 1
    Schedule.Update
   Else
    OleLog = "" 'OleLog & "No schedule found, returning..."
    Call vcfReturn

'@@c2104@@

   End If
```

```
'@@c2105@@

 Case "DialLamp"
   Schedule.FindFirst "ScheduleID=" & ScheduleID&(lineNo%)

   OleLog = "dialing for message lamp " & Schedule("PhoneNumber")
   Call vcfDial("Goto", Schedule("PhoneNumber") & ",,,")

 Case "DialPager"
   Schedule.FindFirst "ScheduleID=" & ScheduleID&(lineNo%)
   res% = GetPrivateProfileString("General", "DialingPrefix", "", retstr$, Len(retstr$) - 1, App.Path &
"\konnects.ini")
   OleLog = "dialing " & Left(retstr$, res%) & Schedule("PhoneNumber") & ",,,," & Schedule
("PagerString")
   Call vcfDial("Goto", Left(retstr$, res%) & Schedule("PhoneNumber") & ",,,," & Schedule
("PagerString"))


 Case "Goto"
   If callstate% = 7 Or callstate% = 8 Or callstate% = 9 Then
    Select Case callstate%
    Case 7
     OleCallLog = "Busy. "
     OleLog = "line busy,"
     status$ = "busy"
    Case 8
     OleCallLog = "No answer. "
     OleLog = "no answer,"
     status$ = "no answer"
    Case 9
     OleLog = "no dialtone or no ring/busy tone,"
     OleCallLog = "No dialtone. "
     status$ = "no tone"
```

```
    End Select

    Call vcfHangup

    Schedule.FindFirst "ScheduleID=" & ScheduleID&(lineNo%)
    If Not Schedule.NoMatch Then
     If Schedule("AttemptsMade") < Schedule("Attempts") Then
      OleLog = OleLog & " schedule " & ScheduleID&(lineNo%) & " is rescheduled..."
      Schedule.Edit
      If Schedule("Action") = "Notify" Then
       Schedule("TimeScheduled") = DateAdd("n", 60, Now)
      Else
       Schedule("TimeScheduled") = DateAdd("n", 2, Now)
      End If
      Schedule("Status") = "Pending" 'status$
      Schedule.Update
     Else
      OleLog = OleLog & " max callout attempts reached, deleting schedule " & ScheduleID&(lineNo%)
'used status log (max callout attempts)

      If Schedule("Action") <> "Notify" And Schedule("Action") <> "Page" And Schedule("Action") <>
"Lamp" Then
       SchedLog.AddNew
       CopySchedule
       SchedLog("Status") = status$
       SchedLog.Update
      End If

      Schedule.Delete
     End If
    End If
   Else
    Schedule.FindFirst "ScheduleID=" & ScheduleID&(lineNo%)
```

```
If Not Schedule.NoMatch Then

 If Schedule("Action") = "GotoBox" Then
  OleCallLog = "Answered, going to mailbox " & Schedule("GotoMailbox") & "."
  OleLog = "callout answered, going to mailbox " & Schedule("GotoMailbox") & " and moving
schedule to log..."  'used by status log (callout answered)
  Call vcfGoto(Schedule("GotoMailbox"))
  SchedLog.AddNew
  CopySchedule
  SchedLog("Status") = "Answered"
  SchedLog.Update
  Schedule.Delete
 ElseIf Schedule("Action") = "Notify" Then
  OleCallLog = "Answered, going to remote mailbox for notification."
  OleLog = "callout answered, going to remote mailbox for notification"  'used by status log (callout
answered)
  Mailboxes.FindFirst "Function = 'DoRemote'"
  If Not Mailboxes.NoMatch Then
   Call vcfGoto(Mailboxes("MailboxNumber"))
  Else
   OleLog = OleLog & ", but remote mailbox not found."
   Call vcfHangup
  End If
  Schedule.Delete
 ElseIf Schedule("Action") = "Page" Then
  res% = GetPrivateProfileString("General", "PageWithCPM", "", retstr$, Len(retstr$) - 1, App.Path
& "\konnects.ini")
  If UCase(Left(retstr$, res%)) = "NO" Then
'   OleCallLog = "Page answered."
   OleLog = "callout answered, hanging up for paging"  'used by status log (callout answered)
   Call vcfHangup
  Else
   OleCallLog = "Answered, send pager string '" & Schedule("PagerString") & "'."
```

```
        OleLog = "callout answered, dialing pager string '" & Schedule("PagerString") & "'"  'used by
status log (callout answered)
        Call vcfDial("Hangup", Schedule("PagerString"))
       End If
       Schedule.Delete
      ElseIf Schedule("Action") = "Lamp" Then
'       OleCallLog = "Lamp string answered."
       OleLog = "callout answered, hanging up for lamp string,"  'used by status log (callout answered)
       Call vcfHangup

       If Schedule("AttemptsMade") < Schedule("Attempts") Then
        OleLog = OleLog & " schedule " & ScheduleID&(lineNo%) & " is rescheduled..."
        Schedule.Edit
        Schedule("TimeScheduled") = DateAdd("n", 2, Now)
        Schedule("Status") = "Pending" 'status$
        Schedule.Update
       Else
        OleLog = OleLog & " max callout attempts reached, deleting schedule " & ScheduleID&
(lineNo%) 'used status log (max callout attempts)
        Schedule.Delete
       End If
      Else
       OleLog = "no such action, '" & Schedule("Action") & "'"
       Call vcfHangup
       SchedLog.AddNew
       CopySchedule
       SchedLog("Status") = "Answered"
       SchedLog.Update
       Schedule.Delete
      End If

     Else
```

```
      OleLog = "schedule " & ScheduleID&(lineNo%) & " callout answered, but schedule deleted..."
'used by status log (callout answered)
      Call vcfHangup
     End If
    End If

  Case "Hangup"
    OleLog = "schedule completed, hanging up"
    Call vcfHangup

  Case Else
    OleLog = "invalid state, going to Hangup"
    Call vcfHangup

  End Select

  FreeLocks

End Sub

Private Sub ScheduleFax()

  Dim lineNo%, Mailbox$, digits$, callstate%, status$
  Dim HaveSchedule As Boolean
  Dim res%, retstr As String * 255
  Dim i%, PhoneNumber$, s$, prefix$, Forbidden As Boolean
  Dim FaxInf$

  lineNo% = OleData(0)
  Mailbox$ = OleData(1)
  digits$ = OleData(2)
  callstate% = OleData(3)
```

```
If State$(lineNo%) = "" Then
 State$(lineNo%) = "FindSchedule"
 counter%(lineNo%) = 0
End If

Select Case State$(lineNo%)

Case "FindSchedule"
 HaveSchedule = False

 If RefreshSched = True Then
  Set Schedule = ScheduleDB.OpenRecordset("Schedule", dbOpenDynaset) ' order by TimeScheduled")
  RefreshSched = False
 End If

 Schedule.FindFirst "Action = 'Fax'"  '"ScheduleID <> Null"
 While Not Schedule.NoMatch And Not HaveSchedule
  If Schedule("Status") = "Deleted" Then
   SchedLog.AddNew
   CopySchedule
   SchedLog("Status") = "Deleted"
   SchedLog.Update
   Schedule.Delete
  ElseIf Schedule("Status") = "Edited" Then
   Schedule.Delete
  ElseIf Schedule("TimeScheduled") < Now And Schedule("Status") = "Pending" Then 'And Schedule
("Action") = "Fax" Then
   HaveSchedule = True
  End If
  If Not HaveSchedule Then Schedule.FindNext "Action = 'Fax'"
 Wend

 If HaveSchedule Then
```

```
s$ = Schedule("PhoneNumber")
    PhoneNumber$ = ""
    For i% = 1 To Len(s$)
     If Mid(s$, i%, 1) >= "0" And Mid(s$, i%, 1) <= "9" Then
      PhoneNumber$ = PhoneNumber$ & Mid(s$, i%, 1)
     End If
    Next i%
    res% = GetPrivateProfileString("General", "LongDistPrefix", "", retstr$, Len(retstr$) - 1, App.Path &
"\konnects.ini")
    If Left(retstr$, res%) <> "" Then
     If InStr(PhoneNumber$, Left(retstr$, res%)) = 1 Then
      PhoneNumber$ = Right(PhoneNumber$, Len(PhoneNumber$) - res%)
     End If
    End If

    res% = GetPrivateProfileString("General", "ForbiddenPrefixes", "", retstr$, Len(retstr$) - 1, App.Path
& "\konnects.ini")
    s$ = Left(retstr$, res%)
    Forbidden = False
    While InStr(s$, ",") <> 0
     If InStr(PhoneNumber$, Left(s$, InStr(s$, ",") - 1)) = 1 Then Forbidden = True
     s$ = Right(s$, Len(s$) - InStr(s$, ","))
    Wend
    If s$ <> "" Then If InStr(PhoneNumber$, s$) = 1 Then Forbidden = True

    If Forbidden = True Then
     OleLog = "schedule " & Schedule("ScheduleID") & " deleted for containing forbidden prefix"

     SchedLog.AddNew
     CopySchedule
     SchedLog("Status") = "fax failed-forbidden #"
     SchedLog.Update
     Schedule.Delete
```

```
Call vcfReturn

    Else

      ScheduleID&(lineNo%) = Schedule("ScheduleID")
      OleLog = "schedule " & ScheduleID&(lineNo%) & " found, faxing to " & Schedule
("PhoneNumber")

      res% = GetPrivateProfileString("General", "LocalAreaCode", "", retstr$, Len(retstr$) - 1, App.Path
& "\konnects.ini")
      If Left(retstr$, res%) <> "" Then
        If InStr(PhoneNumber$, Left(retstr$, res%)) = 1 Then
          PhoneNumber$ = Right(PhoneNumber$, Len(PhoneNumber$) - res%)
        End If
      End If

      If Len(PhoneNumber$) = GetPrivateProfileInt("General", "LongDistDigits", 10, App.Path &
"\konnects.ini") Then
        res% = GetPrivateProfileString("General", "LongDistPrefix", "", retstr$, Len(retstr$) - 1, App.Path
& "\konnects.ini")
        PhoneNumber$ = Left(retstr$, res%) & PhoneNumber$
      End If

      res% = GetPrivateProfileString("General", "DialingPrefix", "", retstr$, Len(retstr$) - 1, App.Path &
"\konnects.ini")
      PhoneNumber$ = Left(retstr$, res%) & PhoneNumber$

      FaxInf$ = PhoneNumber$ & ";Person at Ext. " & Schedule("Extension") & ";" & Schedule
("FaxFrom") & ";" & Schedule("FaxFile")
      If Not IsNull(Schedule("Name")) Then
        If Schedule("Name") <> "" Then
          FaxInf$ = PhoneNumber$ & ";" & Schedule("Name") & ";" & Schedule("FaxFrom") & ";" &
Schedule("FaxFile")
```

```
        End If
      End If
      Call vcfSendFax("GetResult", FaxInf$)
      Schedule.Edit
      Schedule("Status") = "faxing"
      Schedule("AttemptsMade") = Schedule("AttemptsMade") + 1
      Schedule.Update
    End If
  Else
    OleLog = ""  'OleLog & "no fax schedule found, returning..."
    Call vcfReturn
  End If

 Case "GetResult"
  If callstate% <> 100 Then
    Schedule.FindFirst "ScheduleID=" & ScheduleID&(lineNo%)
    If Not Schedule.NoMatch Then
      If Schedule("AttemptsMade") < Schedule("Attempts") Then
        OleI og = "fax attempt failed (" & callstate% & "), schedule " & ScheduleID&(lineNo%) & " is
rescheduled..."
        Schedule.Edit
        Schedule("TimeScheduled") = DateAdd("n", 2, Now)
        Schedule("Status") = "Pending" 'status$
        Schedule.Update
      Else
        OleLog = "fax failed (" & callstate% & "), max attempts reached, deleting schedule " &
ScheduleID&(lineNo%)    'used by status log
        SchedLog.AddNew
        CopySchedule
        SchedLog("Status") = "fax failed-" & callstate%
        SchedLog.Update
        Schedule.Delete
      End If
```

```
    End If
   Else
    Schedule.FindFirst "ScheduleID=" & ScheduleID&(lineNo%)
    If Not Schedule.NoMatch Then
     OleLog = "fax completed, moving schedule to log..."    'used by status log
     SchedLog.AddNew
     CopySchedule
     SchedLog("Status") = "fax completed-" & callstate%
     SchedLog.Update
     Schedule.Delete
    Else
     OleLog = "schedule " & ScheduleID&(lineNo%) & " fax completed, but schedule deleted..."   'used
by status log
    End If
   End If

   Call vcfReturn

  Case Else
   OleLog = "invalid state, going to Hangup"
   Call vcfHangup

  End Select

  FreeLocks

End Sub

Private Sub ExitCall()

  Dim lineNo%, Mailbox$, digits$, status$

  lineNo% = OleData(0)
```

```
Mailbox$ = OleData(1)
digits$ = OleData(2)
status$ = OleData(3)

If State$(lineNo%) = "" Then
 State$(lineNo%) = "CheckFaxes"
 counter%(lineNo%) = 0
End If

Select Case State$(lineNo%)

Case "CheckFaxes"
 If FaxInfo$(lineNo%) <> "" Then
  Mailboxes.FindFirst "Function = 'SendFax'"
  If Not Mailboxes.NoMatch Then
   OleLog = "Exit: faxes selected, going to sendfax box"
   Call vcfGoto(Mailboxes("MailboxNumber"))
  Else
   OleLog = "Exit: faxes selected, but there is no sendfax box, going to goodbye"
   Call vcfContinue("GoodBye")
  End If
 Else
  OleLog = "Exit: no fax selected, going to goodbye"
  Call vcfContinue("GoodBye")
 End If

Case "GoodBye"
 OleLog = "playing file 'good bye...'"
 Call vcfPlayFile("Exit", "System", "goodbye.wav")

Case "Exit"
 OleLog = "Exit: finished, hanging up"
 Call vcfHangup
```

```
End Select

End Sub

Private Sub fSendFax()

 Dim lineNo%, Mailbox$, digits$, callstate%
 Dim i%, HasFaxPort As Boolean

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)
 callstate% = OleData(3)

If State$(lineNo%) = "" Then
  State$(lineNo%) = "Start"
  counter%(lineNo%) = 0
End If

Select Case State$(lineNo%)

Case "Start"
  If FaxInfo$(lineNo%) = "" Then
    OleLog = "Prompt: playing no fax prompt"
'   Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
    Call vcfPlayFile("NoFax", "System", "nofax.wav")
'   OleLog = "Exit: no fax selected, going to goodbye"
'   Call vcfContinue("GoodBye")
  Else
    HasFaxPort = False
    For i = 1 To NumFax%
      If FaxLines(i) = lineNo% Then
        HasFaxPort = True
```

```
      Exit For
     End If
    Next i

    Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"

    If Mailboxes("FaxMethod") = "1CallOnly" Then 'FaxMethod$ = "1" Then
     If HasFaxPort Then
      OleLog = "Exit: faxes were selected, going to 1-call method"
      Call vcfContinue("1CallInstructions")
     Else
      OleLog = "Exit: faxes were selected, going to 2-call method"
      Call vcfContinue("GetFaxNum1")
     End If
    ElseIf Mailboxes("FaxMethod") = "2CallOnly" Then
     OleLog = "Exit: faxes were selected, going to 2-call method"
     Call vcfContinue("GetFaxNum1")
    Else 'FaxMethod% = 3
     If HasFaxPort Then
      OleLog = "Exit: faxes were selected, allow caller to choose fax method"
      Call vcfContinue("ChooseFaxMethod1")
     Else
      OleLog = "Exit: faxes were selected, going to 2-call method"
      Call vcfContinue("GetFaxNum1")
     End If
    End If
   End If

  Case "NoFax"
   OleLog = "finished prompt,"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'NoFax'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
```

```
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If
'-------choose between 1 and 2-call methods----------------------------------------------------

  Case "ChooseFaxMethod1"
   OleLog = "playing file 'if you are calling from a fax...'"
   Call vcfPlayFile("ChooseFaxMethod2", "System", "1or2call.wav")

  Case "ChooseFaxMethod2"
   OleLog = "getting fax method DTMF"
   Call vcfGetDigits("ChooseFaxMethod3", 1, 3)

  Case "ChooseFaxMethod3"
   OleLog = "digits " & digits$ & " received, "
   If digits$ = "1" Then
    OleLog = OleLog & "going to 1-call method"
    counter%(lineNo%) = 0
    Call vcfContinue("1CallInstructions")
   ElseIf digits$ = "2" Then
    OleLog = OleLog & "going to 2-call method"
    counter%(lineNo%) = 0
    Call vcfContinue("GetFaxNum1")
   Else
    OleLog = OleLog & "playing file 'invalid input...'"
    counter(lineNo%) = counter(lineNo%) + 1
    If counter(lineNo%) = 2 Then
     counter%(lineNo%) = 0
     Call vcfContinue("GetFaxNum1")
    Else
```

```
    Call vcfPlayFile("ChooseFaxMethod1", "System", "invalid.wav")
   End If
  End If

'-------2-call Fax-on-demand-----------------------------------------------------------

 Case "GetFaxNum1"
  OleLog = "playing file 'please enter fax #...'"
  Call vcfPlayFile("GetFaxNum2", "System", "enterfax.wav")

 Case "GetFaxNum2"
  OleLog = "getting fax number DTMF"
  Call vcfGetDigits("GetFaxNum3", 15, 3)

 Case "GetFaxNum3"
  OleLog = "digits " & digits$ & " received, "
  If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
  If digits$ <> "" Then
   OleLog = OleLog & "playing file 'you have entered...'"
   temp$(lineNo%) = digits$
   Call vcfPlayFile("GetFaxNum4' , "System", "youenter.wav")
  Else
   OleLog = OleLog & "playing file 'invalid input...'"
   counter(lineNo%) = counter(lineNo%) + 1
   If counter(lineNo%) = 2 Then
    Call vcfPlayFile("Finished", "System", "invalid.wav")
   Else
    Call vcfPlayFile("GetFaxNum1", "System", "invalid.wav")
   End If
  End If

 Case "GetFaxNum4"
  OleLog = "playing fax number " & temp$(lineNo%) & " for verification"
```

```
 Call ssfPlayString("GetFaxNum5", temp$(lineNo%))

Case "GetFaxNum5"
 OleLog = "play file '1 if correct, 2 to enter again...'"
 Call vcfPlayFile("GetFaxNum6", "System", "correct.wav")

Case "GetFaxNum6"
 OleLog = "getting response..."
 Call vcfGetDigits("GetFaxNum7", 1, 3)

Case "GetFaxNum7"
 If digits$ = "1" Then
   OleLog = "fax number correct"
'   FaxInfo$(lineNo%) = temp$(lineNo%) & ";" & FaxInfo$(lineNo%)
   counter%(lineNo%) = 0
   Call vcfContinue("GetExtNum1")
 Else
   OleLog = "get fax number again..."
   Call vcfContinue("GetFaxNum1")
 End If

Case "GetExtNum1"
 OleLog = "playing file 'please enter extension #...'"
 Call vcfPlayFile("GetExtNum2", "System", "enterext.wav")

Case "GetExtNum2"
 OleLog = "getting extension number DTMF"
 Call vcfGetDigits("GetExtNum3", 15, 3)

Case "GetExtNum3"
 OleLog = "digits " & digits$ & " received, "
 If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
 OleLog = OleLog & "putting fax in 2 call queue"
```

```
'   If digits$ <> "" Then
'     OleLog = OleLog & "putting fax in 2 call queue"
'     FaxInfo$(lineNo%) = digits$ & ";" & FaxInfo$(lineNo%)
'     Call vcfContinue("SendFax2Call")
'   Else
'     OleLog = OleLog & "playing file 'invalid input...'"
'     counter(lineNo%) = counter(lineNo%) + 1
'     If counter(lineNo%) = 2 Then
'       Call vcfPlayFile("GoodBye", "System", "invalid.wav")
'     Else
'       Call vcfPlayFile("GetExtNum1", "System", "invalid.wav")
'     End If
'   End If

    Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"

    Schedule.AddNew
'   Schedule("TimeScheduled") = Now
    Schedule("TimeScheduled") = DateAdd("s", 10, Now)
    Schedule("TimeInserted") = Now
    Schedule("Status") = "Pending"   'status$
    Schedule("Action") = "Fax"
    Schedule("PhoneNumber") = temp$(lineNo%)
    Schedule("Extension") = digits$
    Schedule("FaxFrom") = Mailboxes("FaxFrom")
    Schedule("Attempts") = Mailboxes("Attempts")
    Schedule("Name") = ""
    Schedule("OwnerMailbox") = "2-Call"
    If Not IsNull(Mailboxes("FaxCover")) Then
      If Mailboxes("FaxCover") <> "" Then
        FaxInfo$(lineNo%) = AppDir & "mboxes\" & Mailbox$ & "\" & Mailboxes("FaxCover") & FaxInfo
(lineNo%)
      End If
```

```
  End If

  If Left(FaxInfo$(lineNo%), 1) = ";" Then
   FaxInfo$(lineNo%) = Right(FaxInfo$(lineNo%), Len(FaxInfo$(lineNo%)) - 1)
  End If
  Schedule("FaxFile") = FaxInfo$(lineNo%)
  Schedule("AttemptsMade") = 0
  Schedule.Update
  Call vcfContinue("FaxScheduled")

 Case "FaxScheduled"
  OleLog = "playing file 'your fax has been scheduled...'"
  Call vcfPlayFile("Finished", "System", "faxsched.wav")

'-------1-call Fax-on-demand-------------------------------------------------------

 Case "1CallInstructions"
  OleLog = "playing 1-call FOD instructions'"
  Call vcfPlayFile("SendFax1Call", "System", "prsstart.wav")

 Case "SendFax1Call"
  OleLog = "sending 1-call fax" & ";" & FaxInfo$(lineNo%)
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  If Not IsNull(Mailboxes("FaxCover")) Then
   If Mailboxes("FaxCover") <> "" Then
    FaxInfo$(lineNo%) = ";" & AppDir & "mboxes\" & Mailbox$ & "\" & Mailboxes("FaxCover") &
FaxInfo(lineNo%)
   End If
  End If
  Call vcfSendFax("Get1CallResult", ";Here's your fax request.;" & Mailboxes("FaxFrom") & FaxInfo$
(lineNo%))

 Case "Get1CallResult"
```

```
SchedLog.AddNew

SchedLog("ScheduleID") = 0
SchedLog("OwnerMailbox") = "1-Call"
SchedLog("TimeInserted") = Now
SchedLog("TimeScheduled") = Now
SchedLog("Action") = "Fax"
SchedLog("PhoneNumber") = ""
SchedLog("GotoMailbox") = ""
SchedLog("Attempts") = 1
SchedLog("PagerString") = ""
SchedLog("Prompt") = ""
' SchedLog("GetConfirmation") = Schedule("GetConfirmation")
' SchedLog("GetReply") = Schedule("GetReply")
SchedLog("Name") = ""
SchedLog("Extension") = ""
SchedLog("FaxFrom") = ""
SchedLog("FaxFile") = Right(FaxInfo$(lineNo%), Len(FaxInfo$(lineNo%)) - 1)
SchedLog("AttemptsMade") = 1

If callstate% = 100 Then
  OleLog = "1-call faxing completed, going to exit"    'used by status log
  SchedLog("Status") = "fax completed-" & callstate%
Else
  OleLog = "1-call faxing failed (" & callstate% & "), going to exit"    'used by status log
  SchedLog("Status") = "fax failed-" & callstate%
End If

SchedLog.Update

Call vcfContinue("Finished")

'-------Exiting-----------------------------------------------------------------
```

```
' Case "GoodBye"
'   OleLog = "playing file 'good bye...'"
'   Call vcfPlayFile("Exit", "System", "goodbye.wav")

' Case "Exit"
'   OleLog = "Exit: finished, hanging up"
'   Call vcfHangup

  Case "Finished"
    FaxInfo$(lineNo%) = ""
    FaxDocs%(lineNo%) = 0
    OleLog = "finished SendFax,"
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
    If Not Tonetables.NoMatch Then
      Mailbox$ = Tonetables("Assignment")
      OleLog = OleLog & " going to " & Mailbox$
      Call vcfGoto(Mailbox$)
    Else
      OleLog = OleLog & " going to Hangup"
      Call vcfGoto("Hangup")
    End If

  Case Else
    OleLog = "invalid state, going to Hangup"
    Call vcfGoto("Hangup")

  End Select

End Sub

Private Sub fSelectFax()

  Dim doc$, lineNo%, Mailbox$
```

```
lineNo% = OleData(0)
 Mailbox$ = OleData(1)

 If State$(lineNo%) = "" Then
  State$(lineNo%) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  OleLog = "playing prompt file"
  If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.wav") Then
   Call vcfPlayFile("1", "Current", "prompt.wav")
  Else
   Call Jump("1")
  End If

'  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
'  If Not IsNull(Mailboxes("Prompt")) Then
'   Call vcfPlayFile("1", "Current", Mailboxes("Prompt"))
'  Else
'   Call Jump("1")
'  End If

 Case "1"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"

  If Not IsNull(Mailboxes("FaxFile")) Then
   doc$ = Mailboxes("FaxFile")
   If InStr(FaxInfo$(lineNo%), AppDir & "mboxes\" & Mailbox$ & "\" & doc$) = 0 Then
    FaxInfo$(lineNo%) = FaxInfo$(lineNo%) & ";" & AppDir & "mboxes\" & Mailbox$ & "\" & doc$
    OleLog = "selected fax document " & doc$ & " added, " 'used by status log
    FaxDocs(lineNo%) = FaxDocs(lineNo%) + 1
```

```
    Call vcfContinue("10")
   Else
    OleLog = "playing 'document already selected...' file"
    Call vcfPlayFile("10", "System", "alrorder.wav")
   End If
  Else
   Call Jump("10")
  End If

 Case "10"
  If FaxDocs(lineNo%) >= MaxFaxDoc% Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'MaxFax'"
   OleLog = "max fax documents selected,"
   If Tonetables.NoMatch Then Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and
Tone = '~'"
  Else
   OleLog = "document selected,"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  End If

  If Not Tonetables.NoMatch Then
   Mailbox$ = Tonetables("Assignment")
   OleLog = OleLog & " going to " & Mailbox$
   Call vcfGoto(Mailbox$)
  Else
   OleLog = OleLog & " going to Hangup"
   Call vcfGoto("Hangup")
  End If

 End Select

End Sub
```

```
Private Sub fDTMFMenu()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim dialstring$, MaxDigits%, DtmfTimeout%
' Dim FirstDigitTimeout%

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  temp$(lineNo%) = ""
  OleLog = "GetDTMF: playing prompt file"
  If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.wav") Then
   Call vcfPlayFile("1", "Current", "prompt.wav")
  Else
   Call Jump("1")
  End If

 Case "1"
  Mailboxes.FindFirst "MailBoxNumber=" & Mailbox$ & ""
  DtmfTimeout% = 3
  If Not IsNull(Mailboxes("DtmfTimeout")) Then DtmfTimeout% = Mailboxes("DtmfTimeout")
'  FirstDigitTimeout% = 0
'  If Not IsNull(Mailboxes("1stDigitTimeout")) Then FirstDigitTimeout% =
Mailboxes("1stDigitTimeout")
```

```
'  If (Mailboxes("Check1stDigit") = "Yes" Or FirstDigitTimeout% <> 0) And Mailboxes("MaxDigits")
<> 1 Then
   If (Mailboxes("Check1stDigit") = "Yes") And Mailboxes("MaxDigits") <> 1 Then
'    If FirstDigitTimeout% <> 0 Then DtmfTimeout = -1 * FirstDigitTimeout%
     OleLog = "getting first digit of DTMF input"
     Call vcfGetDigits("20", 1, DtmfTimeout%)
   Else
     OleLog = "getting DTMF input"
     MaxDigits% = 30
     If Not IsNull(Mailboxes("MaxDigits")) Then MaxDigits% = Mailboxes("MaxDigits")
     If MaxDigits% = 0 Then MaxDigits% = 30
'    If MaxDigits% = 1 And FirstDigitTimeout% <> 0 Then DtmfTimeout = -1 * FirstDigitTimeout%
     Call vcfGetDigits("2", MaxDigits%, DtmfTimeout%)
   End If

 Case "20"
   If digits$ = "F" Then
     OleLog = "Fax CNG tone detected, "
   Else
     OleLog = "first digit " & digits$ & " received, "
   End If

   temp$(lineNo%) = digits$

   If digits$ = "F" Then
     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='FaxTone'"
   ElseIf digits$ <> "" Then
     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & digits & "'"
   Else
     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='NoInput'"
   End If

   If Not Tonetables.NoMatch Then
```

```
    Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
    If Mailboxes("InputVariable") <> "" Then
     If Left(Mailboxes("InputVariable"), 2) = ">>" Then
      Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right(Mailboxes
("InputVariable"), 1))) & digits
     Else
      Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
     End If
    End If

    If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" & digits$
& "' received. "

    Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
    OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes("Name")
    Call vcfGoto(Tonetables("Assignment"))
    Exit Sub
   End If

   If digits$ = "F" Then
    OleLog = "Fax CNG tone detected, going to Hangup"
    Call vcfHangup
    Exit Sub
   End If

   If digits$ = "#" Or digits$ = "" Then
    counter(lineNo%) = counter(lineNo%) + 1
    Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
    OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")

    If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"

    If counter(lineNo%) = 1 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
```

```
    Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
   ElseIf counter(lineNo%) = 2 And isfile(AppDir & "mboxes\" & MailboxS & "\invalid2.wav") Then
    Call vcfPlayFile(nextstate$, "Current", "invalid2.wav")
   ElseIf counter(lineNo%) = 3 And isfile(AppDir & "mboxes\" & MailboxS & "\invalid3.wav") Then
    Call vcfPlayFile(nextstate$, "Current", "invalid3.wav")
   ElseIf isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
    Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
   Else
    Call vcfPlayFile(nextstate$, "System", "invalid.wav")
   End If
   Exit Sub
  End If

  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  DtmfTimeout% = 3
  If Not IsNull(Mailboxes("DtmfTimeout")) Then DtmfTimeout% = Mailboxes("DtmfTimeout")
  OleLog = OleLog & "getting rest of DTMF input"
  MaxDigits% = 30
  If Not IsNull(Mailboxes("MaxDigits")) Then MaxDigits% = Mailboxes("MaxDigits") - 1
  If MaxDigits% = 0 Then MaxDigits% = 30
  Call vcfGetDigits("2", MaxDigits%, DtmfTimeout%)

 Case "2"
  digits$ = temp$(lineNo%) & digits$

  If InStr(digits$, "F") Then
   OleLog = "Fax CNG tone detected, "
  Else
   OleLog = "digits " & digits$ & " received, "
  End If

  If Right(digits, 1) = "#" And digits <> "#" Then digits = Left(digits, Len(digits) - 1)
```

```
If InStr(digits$, "F") Then
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='FaxTone'"
ElseIf digits$ <> "" Then
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & digits & "'"
Else
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='NoInput'"
End If

If Not Tonetables.NoMatch Then
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 If Mailboxes("InputVariable") <> "" Then
  If Left(Mailboxes("InputVariable"), 2) = ">>" Then
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right(Mailboxes
("InputVariable"), 1))) & digits
  Else
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
  End If
 End If

 If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" & digits$
& "' received. "

 Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
 OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes("Name")
 Call vcfGoto(Tonetables("Assignment"))
 Exit Sub
End If

If InStr(digits$, "F") Then
 OleLog = "Fax CNG tone detected, going to Hangup"
 Call vcfHangup
 Exit Sub
End If
```

```
Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  If Not IsNull(Mailboxes("MinDigits")) Then
   If Mailboxes("MinDigits") > 0 And Len(digits$) >= Mailboxes("MinDigits") Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='MinDigits'"
    If Not Tonetables.NoMatch Then
     If Mailboxes("InputVariable") <> "" Then
      If Left(Mailboxes("InputVariable"), 2) = ">>" Then
       Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right
(Mailboxes("InputVariable"), 1))) & digits
      Else
       Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
      End If
     End If

     If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" &
digits$ & "' received. "

     Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
     OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes
("Name")
     Call vcfGoto(Tonetables("Assignment"))
     Exit Sub
    End If
   End If
  End If

  counter(lineNo%) = counter(lineNo%) + 1
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")

  If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"

  If counter(lineNo%) = 1 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
```

```
    Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
   ElseIf counter(lineNo%) = 2 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid2.wav") Then
    Call vcfPlayFile(nextstate$, "Current", "invalid2.wav")
   ElseIf counter(lineNo%) = 3 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid3.wav") Then
    Call vcfPlayFile(nextstate$, "Current", "invalid3.wav")
   ElseIf isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
    Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
   Else
    Call vcfPlayFile(nextstate$, "System", "invalid.wav")
   End If

  Case "10"
   OleLog = "can not get DTMF input after " & counter(lineNo%) & " attempt(s),"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='Invalid'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If

  End Select

End Sub

Private Sub fDTMFMenuOld()

  Dim lineNo%, Mailbox$, digits$, nextstate$
  Dim dialstring$, MaxDigits%, DtmfTimeout%

  lineNo% = OleData(0)
```

```
Mailbox$ = OleData(1)
digits$ = OleData(2)

If State$(lineNo%) = "" Then
 counter%(lineNo%) = 0
 State$(lineNo) = "0"
End If

Select Case State$(lineNo%)

Case "0"
 OleLog = "GetDTMF: playing prompt file"
' Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.wav") Then
  Call vcfPlayFile("1", "Current", "prompt.wav")
 Else
  Call Jump("1")
 End If

Case "1"
 OleLog = "getting DTMF input"
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 MaxDigits% = 30
 If Not IsNull(Mailboxes("MaxDigits")) Then MaxDigits% = Mailboxes("MaxDigits")
 If MaxDigits% = 0 Then MaxDigits% = 30
 DtmfTimeout% = 3
 If Not IsNull(Mailboxes("DtmfTimeout")) Then DtmfTimeout% = Mailboxes("DtmfTimeout")
 Call vcfGetDigits("2", MaxDigits%, DtmfTimeout%)

Case "2"
 If InStr(digits$, "F") Then
  OleLog = "Fax CNG tone detected, "
 Else
```

```
 OleLog = "digits " & digits$ & " received, "
End If

If Right(digits, 1) = "#" And digits <> "#" Then digits = Left(digits, Len(digits) - 1)

If InStr(digits$, "F") Then
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='FaxTone'"
ElseIf digits$ <> "" Then
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & digits & "'"
Else
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='NoInput'"
End If

If Not Tonetables.NoMatch Then
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 If Mailboxes("InputVariable") <> "" Then
  Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
 End If

 Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
 OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes("Nam
 Call vcfGoto(Tonetables("Assignment"))
 Exit Sub
End If

If InStr(digits$, "F") Then
 OleLog = "Fax CNG tone detected, going to Hangup"
 Call vcfHangup
 Exit Sub
End If

counter(lineNo%) = counter(lineNo%) + 1
Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
```

```
OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")

If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"

If counter(lineNo%) = 1 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
ElseIf counter(lineNo%) = 2 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid2.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid2.wav")
ElseIf counter(lineNo%) = 3 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid3.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid3.wav")
ElseIf isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
Else
 Call vcfPlayFile(nextstate$, "System", "invalid.wav")
End If

Case "10"
 OleLog = "can not get DTMF input after " & counter(lineNo%) & " attempt(s),"
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='Invalid'"
 If Not Tonetables.NoMatch Then
  Mailbox$ = Tonetables("Assignment")
  OleLog = OleLog & " going to " & Mailbox$
  Call vcfGoto(Mailbox$)
 Else
  OleLog = OleLog & " going to Hangup"
  Call vcfGoto("Hangup")
 End If

End Select

End Sub

Private Sub fCIDMenu()
```

```
Dim lineNo%, Mailbox$, digits$, nextstate$, CallerID$
 Dim dialstring$, MaxDigits%, DtmfTimeout%

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)
 CallerID$ = OleData(4)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"

  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & CallerID$ & "'"
  If Tonetables.NoMatch Then
   CallerID$ = Left(CallerID$, 7)
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & CallerID$ & "'"
  End If
  If Tonetables.NoMatch Then
   CallerID$ = Left(CallerID$, 6)
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & CallerID$ & "'"
  End If
  If Tonetables.NoMatch Then
   CallerID$ = Left(CallerID$, 5)
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & CallerID$ & "'"
  End If
  If Tonetables.NoMatch Then
   CallerID$ = Left(CallerID$, 4)
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & CallerID$ & "'"
```

```
    End If
    If Tonetables.NoMatch Then
     CallerID$ = Left(CallerID$, 3)
     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & CallerID$ & "'"
    End If

    If Not Tonetables.NoMatch Then
     Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
     OleLog = "Caller ID " & CallerID$ & " received, going to mailbox " & Tonetables("Assignment") & "
- " & Mailboxes("Name")
     Call vcfGoto(Tonetables("Assignment"))
    Else
     OleLog = "no match for Caller ID,"
     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='~'"
     If Not Tonetables.NoMatch Then
      Mailbox$ = Tonetables("Assignment")
      OleLog = OleLog & " going to " & Mailbox$
      Call vcfGoto(Mailbox$)
     Else
      OleLog = OleLog & " going to Hangup"
      Call vcfGoto("Hangup")
     End If
    End If

  End Select

End Sub

Private Sub fDIDMenu()

  Dim lineNo%, Mailbox$, digits$, nextstate$
  Dim dialstring$, MaxDigits%, DIDTimeout%
' Dim FirstDigitTimeout%
```

```
lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  temp$(lineNo%) = ""
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  DIDTimeout% = 500
  If Not IsNull(Mailboxes("DIDTimeout")) Then DIDTimeout% = Mailboxes("DIDTimeout")
  If Mailboxes("MaxDigits") = 1 Then
   OleLog = "getting one DTMF input"
   Call vcfGetDigits("2", 1, -1 * DIDTimeout%)
  Else
   OleLog = "getting first digit of DTMF input"
   Call vcfGetDigits("20", 1, -1 * DIDTimeout%)
  End If

 Case "20"
  If digits$ = "F" Then
   OleLog = "Fax CNG tone detected, "
  Else
   OleLog = "first digit " & digits$ & " received, "
  End If

  temp$(lineNo%) = digits$
```

```
If digits$ = "F" Then
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='FaxTone'"
ElseIf digits$ <> "" Then
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & digits & "'"
Else
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='NoInput'"
End If

If Not Tonetables.NoMatch Then
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 If Mailboxes("InputVariable") <> "" Then
  If Left(Mailboxes("InputVariable"), 2) = ">>" Then
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right(Mailboxes
("InputVariable"), 1))) & digits
  Else
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
  End If
 End If

 If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" & digits$
& "' received. "

 Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
 OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes("Name")
 Call vcfGoto(Tonetables("Assignment"))
 Exit Sub
End If

If digits$ = "F" Then
 OleLog = "Fax CNG tone detected, going to Hangup"
 Call vcfHangup
 Exit Sub
End If
```

```
If digits$ = "#" Or digits$ = "" Then
    counter(lineNo%) = counter(lineNo%) + 1
    Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
    OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")
    If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"
    Call vcfContinue(nextstate$)
    Exit Sub
  End If

  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  OleLog = OleLog & "getting rest of DID DTMF input"
  MaxDigits% = 30
  If Not IsNull(Mailboxes("MaxDigits")) Then MaxDigits% = Mailboxes("MaxDigits") - 1
  If MaxDigits% = 0 Then MaxDigits% = 30
  Call vcfGetDigits("2", MaxDigits%, 1)

Case "2"
  digits$ = temp$(lineNo%) & digits$

  If InStr(digits$, "F") Then
   OleLog = "Fax CNG tone detected, "
  Else
   OleLog = "digits " & digits$ & " received, "
  End If

  If Right(digits, 1) = "#" And digits <> "#" Then digits = Left(digits, Len(digits) - 1)

  If InStr(digits$, "F") Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='FaxTone'"
  ElseIf digits$ <> "" Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & digits & "'"
  Else
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='NoInput'"
```

```
End If

If Not Tonetables.NoMatch Then
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 If Mailboxes("InputVariable") <> "" Then
  If Left(Mailboxes("InputVariable"), 2) = ">>" Then
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right(Mailboxes
("InputVariable"), 1))) & digits
  Else
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
  End If
 End If

 If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" & digits$
& "' received. "

 Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
 OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes("Name")
 Call vcfGoto(Tonetables("Assignment"))
 Exit Sub
End If

If InStr(digits$, "F") Then
 OleLog = "Fax CNG tone detected, going to Hangup"
 Call vcfHangup
 Exit Sub
End If

Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
If Not IsNull(Mailboxes("MinDigits")) Then
 If Mailboxes("MinDigits") > 0 And Len(digits$) >= Mailboxes("MinDigits") Then
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='MinDigits'"
  If Not Tonetables.NoMatch Then
```

```
        If Mailboxes("InputVariable") <> "" Then
         If Left(Mailboxes("InputVariable"), 2) = ">>" Then
          Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right
(Mailboxes("InputVariable"), 1))) & digits
         Else
          Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
         End If
        End If

        If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" &
digits$ & "' received. "

        Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
        OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes
("Name")
        Call vcfGoto(Tonetables("Assignment"))
        Exit Sub
       End If
      End If
     End If

     counter(lineNo%) = counter(lineNo%) + 1
     Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
     OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")
     If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"
     Call vcfContinue(nextstate$)

   Case "10"
     OleLog = "can not get DTMF input after " & counter(lineNo%) & " attempt(s),"
     Tonetables.FindFirst "MailBoxNumber= ' & Mailbox$ & "'" & "and Tone='Invalid'"
     If Not Tonetables.NoMatch Then
      Mailbox$ = Tonetables("Assignment")
      OleLog = OleLog & " going to " & Mailbox$
```

```
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If

 End Select

End Sub

Private Sub fDIDMenu2()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim dialstring$, MaxDigits%, DIDTimeout%
 Dim res%, retstr As String * 255
 Dim DIDextension$, DIDCP$

' Dim FirstDigitTimeout%

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
  DIDinfo$(lineNo%) = ""
 End If

 Select Case State$(lineNo%)

 Case "0"
  temp$(lineNo%) = ""
```

```
Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
DIDTimeout% = 500
If Not IsNull(Mailboxes("DIDTimeout")) Then DIDTimeout% = Mailboxes("DIDTimeout")
If Mailboxes("MaxDigits") = 1 Then
  OleLog = "getting one DTMF input"
  Call vcfGetDigits("2", 1, -1 * DIDTimeout%)
Else
  OleLog = "getting first digit of DTMF input"
  Call vcfGetDigits("20", 1, -1 * DIDTimeout%)
End If

Case "20"
If digits$ = "F" Then
  OleLog = "Fax CNG tone detected, "
Else
  OleLog = "first digit " & digits$ & " received, "
End If

temp$(lineNo%) = digits$

If digits$ = "F" Then
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='FaxTone'"
ElseIf digits$ <> "" Then
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & digits & "'"
Else
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='NoInput'"
End If

If Not Tonetables.NoMatch Then
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  If Mailboxes("InputVariable") <> "" Then
    If Left(Mailboxes("InputVariable"), 2) = ">>" Then
```

```
      Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right(Mailboxes
("InputVariable"), 1))) & digits
     Else
      Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
     End If
    End If

    If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" & digits$
& "' received. "

    Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
    OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes("Name")
    Call vcfGoto(Tonetables("Assignment"))
    Exit Sub
   End If

   If digits$ = "F" Then
    OleLog = "Fax CNG tone detected, going to Hangup"
    Call vcfHangup
    Exit Sub
   End If

   If digits$ = "#" Or digits$ = "" Then
    counter(lineNo%) = counter(lineNo%) + 1
    Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
    OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")
    If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"
    Call vcfContinue(nextstate$)
    Exit Sub
   End If

   Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
   OleLog = OleLog & "getting rest of DID DTMF input"
```

```
  MaxDigits% = 30
  If Not IsNull(Mailboxes("MaxDigits")) Then MaxDigits% = Mailboxes("MaxDigits") - 1
  If MaxDigits% = 0 Then MaxDigits% = 30
  Call vcfGetDigits("2", MaxDigits%, 1)

 Case "2"
  digits$ = temp$(lineNo%) & digits$

  If InStr(digits$, "F") Then
   OleLog = "Fax CNG tone detected, "
  Else
   OleLog = "digits " & digits$ & " received, "
  End If

  If Right(digits, 1) = "#" And digits <> "#" Then digits = Left(digits, Len(digits) - 1)

  DIDCP$ = Mid(digits, GetPrivateProfileInt("General", "CPStart", 1, AppDir & "mboxes\" & Mailbox$
& "\did.ini"), GetPrivateProfileInt("General", "CPLength", 3, AppDir & "mboxes\" & Mailbox$ &
"\did.ini"))

  res% = GetPrivateProfileString("General", "BusyStrings", "", retstr$, Len(retstr$) - 1, AppDir &
"mboxes\" & Mailbox$ & "\did.ini")

  If InStr(Left(retstr$, res%), DIDCP$) <> 0 Then
   DIDinfo$(lineNo%) = "Busy"
  Else
   res% = GetPrivateProfileString("General", "NoAnswerStrings", "", retstr$, Len(retstr$) - 1, AppDir &
"mboxes\" & Mailbox$ & "\did.ini")
   If InStr(Left(retstr$, res%), DIDCP$) <> 0 Then DIDinfo$(lineNo%) = "NoAnswer"
  End If
```

```
DIDextension$ = Mid(digits, GetPrivateProfileInt("General", "ExtensionStart", 7, AppDir & "mboxes\" & Mailbox$ & "\did.ini"), GetPrivateProfileInt("General", "ExtensionLength", 3, AppDir & "mboxes\" & Mailbox$ & "\did.ini"))

If InStr(digits$, "F") Then
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='FaxTone'"
ElseIf digits$ <> "" Then
'  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & digits & "'"
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='" & DIDextension$ & "'"
Else
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='NoInput'"
End If

If Not Tonetables.NoMatch Then
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 If Mailboxes("InputVariable") <> "" Then
  If Left(Mailboxes("InputVariable"), 2) = ">>" Then
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) & digits
  Else
   Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
  End If
 End If

 If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" & digits$ & "' received. "

 Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
 OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes("Name")
 Call vcfGoto(Tonetables("Assignment"))
 Exit Sub
End If
```

```
If InStr(digits$, "F") Then
 OleLog = "Fax CNG tone detected, going to Hangup"
 Call vcfHangup
 Exit Sub
End If

Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
If Not IsNull(Mailboxes("MinDigits")) Then
 If Mailboxes("MinDigits") > 0 And Len(digits$) >= Mailboxes("MinDigits") Then
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='MinDigits'"
  If Not Tonetables.NoMatch Then
   If Mailboxes("InputVariable") <> "" Then
    If Left(Mailboxes("InputVariable"), 2) = ">>" Then
     Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right
(Mailboxes("InputVariable"), 1))) & digits
    Else
     Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
    End If
   End If

   If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Menu digits '" &
digits$ & "' received. "

   Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
   OleLog = OleLog & "going to mailbox " & Tonetables("Assignment") & " - " & Mailboxes
("Name")
   Call vcfGoto(Tonetables("Assignment"))
   Exit Sub
  End If
 End If
End If

counter(lineNo%) = counter(lineNo%) + 1
```

```
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")
  If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"
  Call vcfContinue(nextstate$)

 Case "10"
  OleLog = "can not get DTMF input after " & counter(lineNo%) & " attempt(s),"
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='Invalid'"
  If Not Tonetables.NoMatch Then
   Mailbox$ = Tonetables("Assignment")
   OleLog = OleLog & " going to " & Mailbox$
   Call vcfGoto(Mailbox$)
  Else
   OleLog = OleLog & " going to Hangup"
   Call vcfGoto("Hangup")
  End If

 End Select

End Sub

Private Sub fGetInput()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim MaxDigits%, DtmfTimeout%

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
```

```
End If

Select Case State$(lineNo%)

Case "0"
  OleLog = "Input: playing prompt file"
  If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.wav") Then
    Call vcfPlayFile("1", "Current", "prompt.wav")
  Else
    Call Jump("1")
  End If

Case "1"
  OleLog = "getting DTMF input"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  DtmfTimeout% = 3
  If Not IsNull(Mailboxes("DtmfTimeout")) Then DtmfTimeout% = Mailboxes("DtmfTimeout")
  MaxDigits% = 30
  If Not IsNull(Mailboxes("MaxDigits")) Then MaxDigits% = Mailboxes("MaxDigits")
  If MaxDigits% = 0 Then MaxDigits% = 30
  Call vcfGetDigits("2", MaxDigits%, DtmfTimeout%)

Case "2"
  OleLog = "digits " & digits$ & " received, "
  If Right(digits, 1) = "#" Then digits = Left(digits, Len(digits) - 1)

  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"

  If Mailboxes("LogInput") = "Yes" Then OleCallLog = "(" & Mailbox$ & ") Input digits '" & digits$ &
"' received. "

  If Len(digits$) < Mailboxes("MinDigits") Then
    counter(lineNo%) = counter(lineNo%) + 1
```

```
OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of " & Mailboxes("Attempts")

If counter(lineNo%) = Mailboxes("Attempts") Then nextstate$ = "10" Else nextstate$ = "0"

If counter(lineNo%) = 1 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
ElseIf counter(lineNo%) = 2 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid2.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid2.wav")
ElseIf counter(lineNo%) = 3 And isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid3.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid3.wav")
ElseIf isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
 Call vcfPlayFile(nextstate$, "Current", "invalid.wav")
Else
 Call vcfPlayFile(nextstate$, "System", "invalid.wav")
End If
Else
If Mailboxes("InputVariable") <> "" Then
 If Left(Mailboxes("InputVariable"), 2) = ">>" Then
  Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = Inp$(lineNo%, Val(Right(Mailboxes
("InputVariable"), 1))) & digits
 Else
  Inp$(lineNo%, Val(Right(Mailboxes("InputVariable"), 1))) = digits
 End If
End If
If Mailboxes("Verify") = "Yes" Then
 OleLog = OleLog & "assigning to " & Mailboxes("InputVariable") & "and playing file 'you have
entered..."
 temp$(lineNo%) = digits$
 Call vcfPlayFile("3", "System", "youenter.wav")
Else
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
 If Not Tonetables.NoMatch Then
```

```
        OleLog = OleLog & "assigning to " & Mailboxes("InputVariable") & " and going to " & Tonetables
("Assignment")
        Call vcfGoto(Tonetables("Assignment"))
       Else
        OleLog = OleLog & "assigning to " & Mailboxes("InputVariable") & " and going to hangup"
        Call vcfGoto("Hangup")
       End If
      End If
     End If

    Case "3"
     OleLog = "playing caller input " & temp$(lineNo%) & " for verification"
     Call ssfPlayString("4", temp$(lineNo%))

    Case "4"
     OleLog = "playing file '1 if correct, 2 to enter again..."
     Call vcfPlayFile("5", "System", "correct.wav")

    Case "5"
     OleLog = "getting response..."
     Call vcfGetDigits("6", 1, 3)

    Case "6"
     If digits$ = "1" Or digits$ = "" Then
      Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
      If Not Tonetables.NoMatch Then
       OleLog = "input verified, going to " & Tonetables("Assignment")
       Call vcfGoto(Tonetables("Assignment"))
      Else
       OleLog = "input verified, going to hangup"
       Call vcfGoto("Hangup")
      End If
     Else
```

```
    OleLog = "get input again..."
    counter%(lineNo%) = 0
    Call vcfContinue("0")
   End If

  Case "10"
   OleLog = "can not get DTMF input after " & counter(lineNo%) & " attempt(s),"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Invalid'"
   If Tonetables.NoMatch Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   End If
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If

  End Select

 End Sub

Private Sub fGetRecording()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim MaxDigits%, maxtime&

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)
```

```
If State$(lineNo%) = "" Then
 counter%(lineNo%) = 0
 State$(lineNo) = "0"
End If

Select Case State$(lineNo%)

Case "0"
 OleLog = "GetRecording: playing prompt file"
 If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.wav") Then
  Call vcfPlayFile("FlushBeforeRecording", "Current", "prompt.wav")
 Else
  Call Jump("FlushBeforeRecording")
 End If

Case "FlushBeforeRecording"
 OleLog = "flushing digits..."
 Call vcfFlushDigits("RecordMessage")

Case "RecordMessage"
 If isfile(App.Path & "\sysvox\temp" & lineNo% & ".raw") Then
  SeekPos&(lineNo%) = FileLen(App.Path & "\sysvox\temp" & lineNo% & ".raw")
 Else
  SeekPos&(lineNo%) = 0
 End If
 OleLog = "recording message"   'used by status log
 OleCallLog = "(" & Mailbox$ & ") Get Recording. "
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 maxtime& = 90
 If Not IsNull(Mailboxes("MaxTime")) Then maxtime& = Mailboxes("MaxTime")
 If maxtime& < 1 Then maxtime& = 90
 Call vcfRecordAppend("FlushAfterRecording", "System", "temp" & lineNo% & ".raw", maxtime&)
```

```
Case "FlushAfterRecording"
  OleLog = "flushing digits..."
  Call vcfFlushDigits("PlayRecording1")

Case "PlayRecording1"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  If Mailboxes("Verify") = "Yes" Then
    OleLog = "playing file 'you have entered...'"
    Call vcfPlayFile("PlayRecording2", "System", "youenter.wav")
  Else
    Call Jump("End")
'   OleLog = OleLog & "play 'message saved' file"
'   Call vcfPlayFile("End", "System", "recsave.wav")
  End If

Case "PlayRecording2"
  OleLog = OleLog & "play recorded message segment"
  Call vcfPlaySeek("PlayRecordOptions", "System", "temp" & lineNo% & ".raw", SeekPos&(lineNo%))

'---------------------------- Record Options
Case "PlayRecordOptions"
  OleLog = "'1 if correct, 2 to enter again...'"
  Call vcfPlayFile("GetDTMFRecordOptions", "System", "correct.wav")

Case "GetDTMFRecordOptions"
  OleLog = "getting digits..."
  Call vcfGetDigits("CheckDTMFRecordOptions", 1, 3)

Case "CheckDTMFRecordOptions"
  OleLog = "digits '" & digits$ & "' received..."
  Select Case digits$
  Case "1"
    Call Jump("End")
```

```
'     OleLog = OleLog & "play 'message saved' file"
'     Call vcfPlayFile("End", "System", "recsave.wav")
'     OleLog = OleLog & "play recorded message"
'     Call vcfPlayFile("PlayRecordOptions", "System", "temp" & lineNo% & ".raw")
    Case "2"
     Call Jump("0")
    Case Else
     Call Jump("End")
'     OleLog = OleLog & "play 'message saved' file"
'     Call vcfPlayFile("End", "System", "recsave.wav")
   End Select

  Case "End"
   OleLog = "recording finished,"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If

  End Select

 End Sub

Private Sub fDoVerify()

 Dim lineNo%, Mailbox$, v$
 Dim d%
 Dim Match As Boolean
```

```
lineNo% = OleData(0)
 Mailbox$ = OleData(1)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  d% = Right(Mailboxes("DataSource"), 1)
  If Left(Mailboxes("FindVariable"), 5) = "Input" Then
   If InStr(Mailboxes("FindVariable"), "*") <> 0 Then
    v$ = Left(Inp$(lineNo%, Val(Mid(Mailboxes("FindVariable"), 6, 1))), Val(Mid(Mailboxes
("FindVariable"), 8, 1)))
'    MsgBox ":" & v$ & ":"
   Else
    v$ = Inp$(lineNo%, Right(Mailboxes("FindVariable"), 1))
   End If
  ElseIf Left(Mailboxes("FindVariable"), 8) = "CallerID" Then
   v$ = OleData(4)  'CallerID variable"
  Else
   v$ = "" & Mailboxes("FindVariable")
  End If


  ds(d%).Edit
  ds(d%).Update

 If ds(d%).Restartable = True Then
  ds(d%).Requery
```

```
End If


 ds(d%).FindFirst Mailboxes("FindField") & " = '" & v$ & "'"
 If Not ds(d%).NoMatch Then
  If IsNull(Mailboxes("MatchField")) Or Mailboxes("MatchField") = "" Then
   Match = True
  Else
   If Left(Mailboxes("MatchVariable"), 5) = "Input" Then
    If InStr(Mailboxes("MatchVariable"), "*") <> 0 Then
     v$ = Left(Inp$(lineNo%, Val(Mid(Mailboxes("MatchVariable"), 6, 1))), Val(Mid(Mailboxes
("MatchVariable"), 8, 1)))
    Else
     v$ = Inp$(lineNo%, Right(Mailboxes("MatchVariable"), 1))
    End If
   Else
    v$ = "" & Mailboxes("MatchVariable")
   End If
   If ds(d%)(Mailboxes("MatchField")) = v$ Then Match = True Else Match = False
  End If
  If Match = True Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = "DoVerify: match valid, going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = "DoVerify: match valid, going to Hangup"
    Call vcfGoto("Hangup")
   End If
  Else
'   Call vcfPlayFile("10", "Current", "invalid.wav")
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
```

```
      OleLog = "DoVerify: match invalid, playing invalid file"
      Call vcfPlayFile("10", "Current", "invalid.wav")
     Else
      OleLog = "DoVerify: match invalid"
      Call vcfContinue("10")
     End If
    End If
   Else
   ' Call vcfPlayFile("10", "Current", "invalid.wav")
    If isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
     OleLog = "DoVerify: variable not found, playing invalid file"
     Call vcfPlayFile("10", "Current", "invalid.wav")
    Else
     OleLog = "DoVerify: variable not found"
     Call vcfContinue("10")
    End If
   End If

Case "10"
  OleLog = "unable to verify,"
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Invalid'"
  If Tonetables.NoMatch Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  End If
  If Not Tonetables.NoMatch Then
   Mailbox$ = Tonetables("Assignment")
   OleLog = OleLog & " going to " & Mailbox$
   Call vcfGoto(Mailbox$)
  Else
   OleLog = OleLog & " going to Hangup"
   Call vcfGoto("Hangup")
  End If
```

```
 End Select

End Sub

Private Sub fAddRecord()

 Dim lineNo%, Mailbox$
 Dim d%, s$, ss$, f$, v$, fvalue$

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)

 If State$(lineNo%) = "" Then
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  Mailboxes.FindFirst "MailBoxNumber ='" & Mailbox$ & "'"
  d% = Right(Mailboxes("DataSource"), 1)
  s$ = Mailboxes("Script")

  ds(d%).AddNew
  While s$ <> ""
   If InStr(s$, Chr(10)) <> 0 Then
    ss$ = Left(s$, InStr(s$, Chr(10)) - 1)
    s$ = Right(s$, Len(s$) - InStr(s$, Chr(10)))
   Else
    ss$ = s$:  s$ = ""
   End If

   v$ = Right(ss$, Len(ss$) - InStr(ss$, "="))
```

```
    If InStr(v$, Chr(13)) <> 0 Then v$ = Left(v$, Len(v$) - 1)
    f$ = Left(ss$, InStr(ss$, "=") - 1)

    If UCase(Left(v$, 5)) = "INPUT" Then
      fvalue$ = Inp$(lineNo%, Val(Right(v$, 1)))
    Else
      fvalue$ = v$
    End If

    If ds(d%).Fields(f$).Type = 10 Then
      ds(d%)(f$) = fvalue$
    ElseIf ds(d%).Fields(f$).Type >= 3 And ds(d%).Fields(f$).Type <= 5 Then
      ds(d%)(f$) = Val(fvalue$)
    End If
  Wend

  On Error Resume Next
  ds(d%).Update

ds(d%).Close
Set ds(d%) = db(ds_dbi%(d%)).OpenRecordset(ds_string$(d%), dbOpenDynaset)

  If Err.Number = 0 Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
    If Not Tonetables.NoMatch Then
      Mailbox$ = Tonetables("Assignment")
      OleLog = "record added, going to " & Mailbox$
      Call vcfGoto(Mailbox$)
    Else
      OleLog = "record added, going to Hangup"
      Call vcfGoto("Hangup")
    End If
  Else
```

```
    OleLog = "can not add record, playing invalid file"
    Call vcfPlayFile("10", "Current", "invalid.wav")
   End If

   On Error GoTo 0

  Case "10"
   OleLog = "unable to add record,"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Invalid'"
   If Tonetables.NoMatch Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   End If
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If

  End Select

End Sub

Private Sub fEditRecord()

  Dim lineNo%, Mailbox$
  Dim d%, s$, ss$, f$, v$, fvalue$

  lineNo% = OleData(0)
  Mailbox$ = OleData(1)
```

```
If State$(lineNo%) = "" Then
 State$(lineNo) = "0"
End If

Select Case State$(lineNo%)

Case "0"
 Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
 d% = Right(Mailboxes("DataSource"), 1)
 s$ = Mailboxes("Script")

 If Left(Mailboxes("FindVariable"), 5) = "Input" Then
  v$ = Inp$(lineNo%, Right(Mailboxes("FindVariable"), 1))
 Else
  v$ = "" & Mailboxes("FindVariable")
 End If

 ds(d%).FindFirst Mailboxes("FindField") & " = '" & v$ & "'"

 If Not ds(d%).NoMatch Then
  ds(d%).Edit
  While s$ <> ""
   If InStr(s$, Chr(10)) <> 0 Then
    ss$ = Left(s$, InStr(s$, Chr(10)) - 1)
    s$ = Right(s$, Len(s$) - InStr(s$, Chr(10)))
   Else
    ss$ = s$:  s$ = ""
   End If

   v$ = Right(ss$, Len(ss$) - InStr(ss$, "="))
   If InStr(v$, Chr(13)) <> 0 Then v$ = Left(v$, Len(v$) - 1)
   f$ = Left(ss$, InStr(ss$, "=") - 1)
```

```
    If UCase(Left(v$, 5)) = "INPUT" Then
      fvalue$ = Inp$(lineNo%, Val(Right(v$, 1)))
    Else
      fvalue$ = v$
    End If

    If ds(d%).Fields(f$).Type = 10 Then
      ds(d%)(f$) = fvalue$
    ElseIf ds(d%).Fields(f$).Type >= 3 And ds(d%).Fields(f$).Type <= 5 Then
      ds(d%)(f$) = Val(fvalue$)
    End If
  Wend

  On Error Resume Next
  ds(d%).Update

ds(d%).Close
Set ds(d%) = db(ds_dbi%(d%)).OpenRecordset(ds_string$(d%), dbOpenDynaset)

  If Err.Number = 0 Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
    If Not Tonetables.NoMatch Then
      Mailbox$ = Tonetables("Assignment")
      OleLog = "record added, going to " & Mailbox$
      Call vcfGoto(Mailbox$)
    Else
      OleLog = "record added, going to Hangup"
      Call vcfGoto("Hangup")
    End If
  Else
    OleLog = "can not edit record, playing invalid file"
    Call vcfPlayFile("10", "Current", "invalid.wav")
  End If
```

```
On Error GoTo 0
   Else
    OleLog = "can not edit record, playing invalid file"
    Call vcfPlayFile("10", "Current", "invalid.wav")
   End If

  Case "10"
   OleLog = "unable to edit record,"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Invalid'"
   If Tonetables.NoMatch Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   End If
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If

  End Select

End Sub

Private Sub fEvalField()

  Dim lineNo%, Mailbox$, v$
  Dim d%
  Dim Match$

  lineNo% = OleData(0)
  Mailbox$ = OleData(1)
```

```
If State$(lineNo%) = "" Then
   counter%(lineNo%) = 0
   State$(lineNo) = "0"
  End If

  Select Case State$(lineNo%)

  Case "0"
   Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
   d% = Right(Mailboxes("DataSource"), 1)

   If Left(Mailboxes("FindVariable"), 5) = "Input" Then

    If InStr(Mailboxes("FindVariable"), "*") <> 0 Then    'for example Input1*6 means the first 6 digits
of Input1
      v$ = Left(Inp$(lineNo%, Val(Mid(Mailboxes("FindVariable"), 6, 1))), Val(Mid(Mailboxes
("FindVariable"), 8, 1)))
     Else
      v$ = Inp$(lineNo%, Right(Mailboxes("FindVariable"), 1))
     End If

   ElseIf Left(Mailboxes("FindVariable"), 8) = "CallerID" Then
    v$ = OleData(4)   'CallerID variable"
   Else
    v$ = "" & Mailboxes("FindVariable")
   End If

   ds(d%).Edit
   ds(d%).Update

   If ds(d%).Restartable = True Then
    ds(d%).Requery
   End If
```

```
ds(d%).FindFirst Mailboxes("FindField") & " = '" & v$ & "'"
  If Not ds(d%).NoMatch Then

    If ds(d%)(Mailboxes("MatchField")) < Val(Mailboxes("MatchValue")) Then
     Match = "<"
    ElseIf ds(d%)(Mailboxes("MatchField")) = Val(Mailboxes("MatchValue")) Then
     Match = "="
    Else
     Match = ">"
    End If

    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "' and Tone ='" & Match & "'"
    If Tonetables.NoMatch Then Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and
Tone = '~'"

    If Not Tonetables.NoMatch Then
     Mailbox$ = Tonetables("Assignment")
     OleLog = "EvalField: match " & Match & ", going to " & Mailbox$
     Call vcfGoto(Mailbox$)
    Else
     OleLog = "EvalField: match " & Match & ", going to Hangup"
     Call vcfGoto("Hangup")
    End If

  Else

    If isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
     OleLog = "EvalField: FindVariable not found, playing invalid file"
     Call vcfPlayFile("10", "Current", "invalid.wav")
    Else
     OleLog = "DoVerify: FindVariable not found"
     Call vcfContinue("10")
    End If
```

```
End If

 Case "10"
  OleLog = "unable to find FindVariable,"
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Invalid'"
  If Tonetables.NoMatch Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  End If
  If Not Tonetables.NoMatch Then
   Mailbox$ = Tonetables("Assignment")
   OleLog = OleLog & " going to " & Mailbox$
   Call vcfGoto(Mailbox$)
  Else
   OleLog = OleLog & " going to Hangup"
   Call vcfGoto("Hangup")
  End If

 End Select

End Sub

Private Sub fPlayPrompt()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim dialstring$, MaxDigits%

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
```

```
End If

Select Case State$(lineNo%)

Case "0"
 OleLog = "playing prompt file"

 If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.wav") Then
  Call vcfPlayFile("10", "Current", "prompt.wav")
 Else
  Call Jump("10")
 End If


' Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
' Call vcfPlayFile("10", "Current", Mailboxes("Prompt"))

Case "10"
 OleLog = "finished prompt,"
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
 If Not Tonetables.NoMatch Then
  Mailbox$ = Tonetables("Assignment")
  OleLog = OleLog & " going to " & Mailbox$
  Call vcfGoto(Mailbox$)
 Else
  OleLog = OleLog & " going to Hangup"
  Call vcfGoto("Hangup")
 End If

End Select

End Sub
```

```
Private Sub fSetCounter()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim dialstring$, MaxDigits%, c%
 Static Count%(1 To 74, 0 To 5)

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  c% = Val(Right(Mailboxes("CounterVariable"), 1))
  If Mailboxes("CounterAction") = "Reset" Then
   OleLog = "resetting Counter" & c%
   Count(lineNo%, c%) = 0
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  Else
   Count(lineNo%, c%) = Count(lineNo%, c%) + 1
   If Count(lineNo%, c%) < Mailboxes("MaxCount") Then
    OleLog = "incrementing Counter" & c%
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   Else
    OleLog = "incrementing Counter" & c% & " to its max,"
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'MaxCount'"
    If Tonetables.NoMatch Then
```

```
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   End If
  End If
 End If

 If Not Tonetables.NoMatch Then
  Mailbox$ = Tonetables("Assignment")
  OleLog = OleLog & " going to " & Mailbox$
  Call vcfGoto(Mailbox$)
 Else
  OleLog = OleLog & " going to Hangup"
  Call vcfGoto("Hangup")
 End If

End Select

End Sub

Private Sub fSetLanguage()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim dialstring$, MaxDigits%, c%

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)
```

```
Case "0"
   Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
   If Not IsNull(Mailboxes("Language")) Then
    If UCase(Mailboxes("Language")) <> "WAV" Then
     Language%(lineNo%) = Val(Right(Mailboxes("Language"), 1))
    End If
   End If

   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If

 End Select

End Sub

Private Sub fGetHours()

 Dim lineNo%, Mailbox$, digits$, nextstate$
 Dim res%, retstr As String * 255
 Dim dow$, s$

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

'   Debug.Print Time
```

```
res% = GetPrivateProfileString("General", "Holidays", "", retstr$, Len(retstr$) - 1, AppDir & "vmail.ini")
 If InStr(Left(retstr$, res%), Format(Now, "mm/dd")) Then
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Holiday'"
  OleLog = "Holiday found,"
 Else
  dow$ = Format(Now, "dddd")
  res% = GetPrivateProfileString("General", dow$ & "Hours", "Off", retstr$, Len(retstr$) - 1, AppDir &
"vmail.ini")
  s$ = Left(retstr$, res%)
'  Debug.Print s$ & ":"
  On Error Resume Next
  If s$ = "Off" Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Closed'"
   OleLog = "Closed hours,"
  ElseIf Len(s$) > 12 Then
   If Time < CDate(Mid(s$, 1, 5)) Or Time > CDate(Mid(s$, 19, 5)) Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Closed'"
    OleLog = "Closed hours,"
   ElseIf Time > CDate(Mid(s$, 7, 5)) And Time < CDate(Mid(s$, 13, 5)) Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Lunch'"
    OleLog = "Lunch hours,"
   Else
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Open'"
    OleLog = "Open hours,"
   End If
  Else
   If Time < CDate(Mid(s$, 1, 5)) Or Time > CDate(Mid(s$, 7, 5)) Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Closed'"
    OleLog = "Closed hours,"
   Else
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Open'"
    OleLog = "Open hours,"
   End If
```

```
    End If
    On Error GoTo 0
  End If

  If Tonetables.NoMatch Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  End If

'   If Time >= #8:00:00 AM# And Time <= #5:00:00 PM# Then
'     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Open'"
'   Else
'     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Closed'"
'   End If

  If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
  Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
  End If

End Sub

Private Sub fDoScript()

  Dim lineNo%, Mailbox$, digits$, nextstate$
  Dim s$, ss$, i%, func$, dsi%, dsi2%, f$, fvalue$
  Dim res%, retstr As String * 255
  Dim mState$
  Dim FindField$, FindVariable$
  Dim p(1 To 10) As ParseType
```

```
lineNo% = OleData(0)
 MailboxS = OleData(1)
 digits$ = OleData(2)

 If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  s$ = Mailboxes("Script")
  i = 0
  While s$ <> "" And i <= counter%(lineNo%)
   If InStr(s$, Chr(10)) <> 0 Then
    ss$ = Left(s$, InStr(s$, Chr(10)) - 1)
    s$ = Right(s$, Len(s$) - InStr(s$, Chr(10)))
   Else
    ssS = s$
    s$ = ""
   End If
   i = i + 1
  Wend

  If i > counter%(lineNo%) Then
   If InStr(ss$, "(") <> 0 And InStr(ss$, ")") <> 0 Then
    func$ = Left(ss$, InStr(ss$, "(") - 1)
    ss$ = Left(ss$, InStr(ss$, ")") - 1)
    ss$ = Right(ss$, Len(ss$) - InStr(ss$, "("))
   Else
    f$ = ""
```

```
End If

Select Case func$

Case "Tester"
 OleLog = "tester dialing " & Left(Format(Now, "ss"), 1)
 Call vcfDial("0", Left(Format(Now, "ss"), 1))

Case "PlayFile"
 OleLog = "playing file for PlayFile script function"
 Call vcfPlayFile("0", "Current", ss$)

Case "PlaySystemFile"
 OleLog = "playing system file for PlayFile script function"
 Call vcfPlayFile("0", "System", ss$)

Case "SaveInputsAndRecording"
 RecFileName$(lineNo%) = GetRecFileName$(Mailbox$)
 OleLog = "Saving Inputs to " & RecFileName$(lineNo%) & "."
 res% = WritePrivateProfileString("General", "Input", "0:" & Inp$(lineNo%, 0) & ",1:" & Inp$
(lineNo%, 1) & ",2:" & Inp$(lineNo%, 2) & ",3:" & Inp$(lineNo%, 3) & ",4:" & Inp$(lineNo%, 4) &
",5:" & Inp$(lineNo%, 5) & ",6:" & Inp$(lineNo%, 6) & ",7:" & Inp$(lineNo%, 7) & ",8:" & Inp$
(lineNo%, 8) & ",9:" & Inp$(lineNo%, 9), Left(RecFileName$(lineNo%), Len(RecFileName$(lineNo%))
- 3) & "inf")
 If isfile(App.Path & "\sysvox\temp" & lineNo% & ".raw") Then
  Call CvtWaveFile(App.Path & "\sysvox\temp" & lineNo% & ".raw", Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "wav")
 End If
 If isfile(RecFileName$(lineNo%)) Then
  On Error Resume Next
  Kill RecFileName$(lineNo%)
 End If
```

```
    res% = GetPrivateProfileString("General", "State", "", retstr$, Len(retstr$) - 1, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 12) & "status")
    mState$ = Str(Val(Left(retstr$, res%)) + 1)
    If mState$ > "30000" Then mState$ = "0"
    res% = WritePrivateProfileString("General", "State", mState$, Left(RecFileName$(lineNo%), Len
(RecFileName$(lineNo%)) - 12) & "status")
    RecFileName$(lineNo%) = ""
    Call vcfContinue("0")

  Case "SetInputAs"
    OleLog = "setting Input" & Mid(ss$, InStr(ss$, ",") - 1, 1) & " as " & Right(ss$, Len(ss$) - InStr(ss$,
",",))
    Inp$(lineNo%, Val(Mid(ss$, InStr(ss$, ",") - 1, 1))) = Right(ss$, Len(ss$) - InStr(ss$, ","))
    Call vcfContinue("0")

  Case "AddRecord"
    dsi% = Val(Mid(ss$, InStr(ss$, ",") - 1, 1))
    s$ = Right(ss$, Len(ss$) - InStr(ss$, ","))
    ds(dsi%).AddNew
    While s$ <> ""
     If InStr(s$, ",") <> 0 Then
      ss$ = Left(s$, InStr(s$, ",") - 1)
      s$ = Right(s$, Len(s$) - InStr(s$, ","))
     Else
      ss$ = s$:  s$ = ""
     End If

     f$ = Right(ss$, Len(ss$) - InStr(ss$, ":"))

     If UCase(Left(ss$, 5)) = "INPUT" Then
      fvalue$ = Inp$(lineNo%, Val(Mid(ss$, InStr(ss$, ":") - 1, 1)))
     Else
      fvalue$ = Left(ss$, InStr(ss$, ":") - 1)
```

```
  End If
  If ds(dsi%).Fields(f$).Type = 10 Then
    ds(dsi%)(f$) = fvalue$
  ElseIf ds(dsi%).Fields(f$).Type >= 3 And ds(dsi%).Fields(f$).Type <= 5 Then
    ds(dsi%)(f$) = Val(fvalue$)
  End If
 Wend
 ds(dsi%).Update

 OleLog = "adding record for AddRecord script function"
 Call vcfContinue("0")

Case "PlayLongFieldAsCurrency"

     'PlayLongFieldAsCurrency(Data0,CID,CallerID$,Credit)

 dsi% = Val(Mid(ss$, InStr(ss$, ",") - 1, 1))
 s$ = Right(ss$, Len(ss$) - InStr(ss$, ","))

 ss$ = Left(s$, InStr(s$, ",") - 1)
 s$ = Right(s$, Len(s$) - InStr(s$, ","))
 FindField$ = ss$

 ss$ = Left(s$, InStr(s$, ",") - 1)
 s$ = Right(s$, Len(s$) - InStr(s$, ","))
 If Left(ss$, 5) = "Input" Then
  If InStr(ss$, "*") <> 0 Then
    FindVariable$ = Left(Inp$(lineNo%, Val(Mid(ss$, 6, 1))), Val(Mid(ss$, 8, 1)))
  Else
    FindVariable$ = Inp$(lineNo%, Right(ss$, 1))
  End If
 ElseIf Left(ss$, 8) = "CallerID" Then
  FindVariable$ = OleData(4)   'CallerID variable"
```

```
Else
  FindVariable$ = ss$
End If

ss$ = s$

ds(dsi%).FindFirst FindField$ & " = '" & FindVariableS & "'"
If Not ds(dsi%).NoMatch Then
  OleLog = "playing currency of " & ds(dsi%)(ssS) & " cents..."
  Call ssfPlayCurrency("Dummy", ds(dsi%)(ss$) \ 100, ds(dsi%)(ss$) Mod 100)
  If State$(lineNo%) = "Dummy" Then
    State$(lineNo%) = "0"
  Else
    counter%(lineNo%) = counter%(lineNo%) - 1
  End If
 Else
  OleLog = "record not found"
  Call vcfContinue("0")
 End If

Case "PlayLongFieldAsNumber"

  dsi% = Val(Mid(ss$, InStr(ss$, ",") - 1, 1))
  s$ = Right(ss$, Len(ss$) - InStr(ss$, ","))

  ss$ = Left(s$, InStr(s$, ",") - 1)
  s$ = Right(s$, Len(s$) - InStr(s$, ","))
  FindField$ = ss$

  ss$ = Left(s$, InStr(s$, ",") - 1)
  s$ = Right(s$, Len(s$) - InStr(s$, ","))
  If Left(ss$, 5) = "Input" Then
    If InStr(ss$, "*") <> 0 Then
```

```
    FindVariable$ = Left(Inp$(lineNo%, Val(Mid(ss$, 6, 1))), Val(Mid(ss$, 8, 1)))
   Else
    FindVariable$ = Inp$(lineNo%, Right(ss$, 1))
   End If
 ElseIf Left(ss$, 8) = "CallerID" Then
   FindVariable$ = OleData(4)   'CallerID variable"
 Else
   FindVariable$ = ss$
 End If

 ss$ = s$

 ds(dsi%).FindFirst FindField$ & " = '" & FindVariable$ & "'"
 If Not ds(dsi%).NoMatch Then
   OleLog = "playing number " & ds(dsi%)(ss$) & "..."
   Call ssfPlayNumber("Dummy", ds(dsi%)(ss$))
   If State$(lineNo%) = "Dummy" Then
     State$(lineNo%) = "0"
   Else
     counter%(lineNo%) = counter%(lineNo%) - 1
   End If
 Else
   OleLog = "record not found"
   Call vcfContinue("0")
 End If

Case "PlayInputAsString"

 FindVariable$ = Inp$(lineNo%, Right(ss$, 1))
 OleLog = "playing input string " & FindVariable$ & "..."
 Call ssfPlayString("Dummy", FindVariable$)
 If State$(lineNo%) = "Dummy" Then
   State$(lineNo%) = "0"
```

```
    Else
      counter%(lineNo%) = counter%(lineNo%) - 1
    End If

  Case "PlayDate"
    FindVariable$ = Inp$(lineNo%, Right(ss$, 1))

    If FindVariable$ = "1" Then
      FindVariable$ = Format(Now, "mmdd")
    ElseIf FindVariable$ = "2" Then
      FindVariable$ = Format(DateAdd("d", 1, Now), "mmdd")
    ElseIf FindVariable$ = "3" Then
      FindVariable$ = Format(DateAdd("d", 2, Now), "mmdd")
    End If

    OleLog = "playing date on input string '" & FindVariable$ & "'..."

    Call ssfPlayDate("Dummy", FindVariable$)

    If State$(lineNo%) = "Dummy" Then
      State$(lineNo%) = "0"
    Else
      counter%(lineNo%) = counter%(lineNo%) - 1
    End If

  Case "PlayTime"
    FindVariable$ = Inp$(lineNo%, Val(Mid(ss$, InStr(ss$, ",") - 1, 1)))
    If Inp$(lineNo%, Right(ss$, 1)) = "1" Then
      If Left(FindVariable$, 2) = "12" Then FindVariable$ = "00" & Right(FindVariable$, 2)
    Else
      If Left(FindVariable$, 2) <> "12" Then FindVariable$ = Val(Left(FindVariable$, 2)) + 12 & Right
(FindVariable$, 2)
    End If
```

```
OleLog = "playing time on input string '" & FindVariable$ & "'..."

    Call ssfPlayTime("Dummy", FindVariable$)

    If State$(lineNo%) = "Dummy" Then
     State$(lineNo%) = "0"
    Else
     counter%(lineNo%) = counter%(lineNo%) - 1
    End If

#If ext Then
'###########################################################################

   Case "SchedAlarm"

    Dim sDate$, sTime$, sAMPM$
    Dim dDate As Date

    Call ParseString(ss$, p())
    sDate$ = ""
    If Left(p(1).Parastr, 5) = "Input" Then sDate$ = Inp$(lineNo%, Right(p(1).Parastr, 1))

    sTime$ = ""
    If Left(p(2).Parastr, 5) = "Input" Then sTime$ = Inp$(lineNo%, Right(p(2).Parastr, 1))

    sAMPM$ = ""
    If Left(p(3).Parastr, 5) = "Input" Then sAMPM$ = Inp$(lineNo%, Right(p(3).Parastr, 1))

    If sDate$ = "1" Then
     sDate$ = Format(Now, "mmdd")
    ElseIf sDate$ = "2" Then
     sDate$ = Format(DateAdd("d", 1, Now), "mmdd")
    ElseIf sDate$ = "3" Then
```

```
    sDate$ = Format(DateAdd("d", 2, Now), "mmdd")
   End If

   If sAMPM$ = 2 Then
    dDate = CDate(Left(sDate$, 2) & "/" & Right(sDate$, 2) & " " & Left(sTime$, 2) & ":" & Right
(sTime$, 2) & " PM")
   Else
    dDate = CDate(Left(sDate$, 2) & "/" & Right(sDate$, 2) & " " & Left(sTime$, 2) & ":" & Right
(sTime$, 2) & " AM")
   End If

   If dDate < Now Then dDate = DateAdd("yyyy", 1, dDate)

   If DateDiff("d", Now, dDate) <= 100 Then
    Schedule.AddNew
    Schedule("TimeScheduled") = dDate
    Schedule("TimeInserted") = Now
    Schedule("Status") = "Pending"   'status$
    Schedule("Action") = "GotoBox"
    Schedule("PhoneNumber") = Inp$(lineNo%, Right(p(4).Parastr, 1))
    Schedule("GotoMailbox") = p(5).Parastr
    Schedule("Attempts") = Val(p(6).Parastr)
    Schedule("Name") = ""
    Schedule("OwnerMailbox") = Mailbox$
    Schedule("AttemptsMade") = 0
    Schedule.Update
   End If

   OleLog = "alarm call at " & dDate & " to " & Inp$(lineNo%, Right(p(4).Parastr, 1)) & " scheduled"
   OleCallLog = "Alarm call at " & dDate & " to " & Inp$(lineNo%, Right(p(4).Parastr, 1)) & "
scheduled."
   Call vcfContinue("0")
```

```
#End If
'EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE


   Case Else
    OleLog = "commented or invalid script function " & func$
    Call vcfContinue("0")

   End Select

   counter%(lineNo%) = counter%(lineNo%) + 1

  Else

   OleLog = "no more script, "
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If
  End If

 End Select

End Sub

Private Sub fDoTransfer()

 Dim lineNo%, Mailbox$, digits$, nextstate$, callstate%, CallerID$
```

```
Dim dialstring$, res%, retstr As String * 255
Dim Profile$, i%, s$
Dim mState$
Dim TransferHour As Boolean, dow$

Static tmpCallState%(1 To 74)

lineNo% = OleData(0)
Mailbox$ = OleData(1)
digits$ = OleData(2)
callstate% = OleData(3)
CallerID$ = OleData(4)
Profile$ = AppDir & "mboxes\" & Mailbox$ & "\transfer.ini"

If State$(lineNo%) = "" Then
  State$(lineNo%) = "GetAllowTransfer"
  tmpCallState%(lineNo%) = 0
  On Error Resume Next
  If Not isfile(Profile$) Then FileCopy App.Path & "\transfer.ini", Profile$
End If

Select Case State$(lineNo%)

'********************** Get Allow Transfer
Case "GetAllowTransfer"
  If PreviousFunction$(lineNo%) = "DIDMenu" Then
    OleLog = "Previous function was DIDMenu so do not do the transfer"
    Call vcfContinue("GetPlayPersonalGreeting")
  ElseIf PreviousFunction$(lineNo%) = "DIDMenu2" Then
    If DIDinfo$(lineNo%) = "Busy" Then
      tmpCallState%(lineNo%) = 7
      OleLog = "Previous function was DIDMenu2 with busy, so do not do the transfer"
    Else
```

```
      tmpCallState%(lineNo%) = 8
      OleLog = "Previous function was DIDMenu2 with no answer, so do not do the transfer"
    End If
    Call vcfContinue("PlayNoTransfer")
  Else
    res% = GetPrivateProfileString("Transfer", "AllowTransfer", "Yes", retstr$, Len(retstr$) - 1, Profile$)
    If UCase(Left(retstr$, res%)) = "YES" Then

      TransferHour = False
      res% = GetPrivateProfileString("General", "Holidays", "", retstr$, Len(retstr$) - 1, AppDir &
"vmail.ini")
      If InStr(Left(retstr$, res%), Format(Now, "mm/dd")) Then
        OleLog = "Holiday hours,"
        res% = GetPrivateProfileString("Transfer", "HolidayTransfer", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
        If UCase(Left(retstr$, res%)) = "YES" Then TransferHour = True
      Else
        dow$ = Format(Now, "dddd")
        res% = GetPrivateProfileString("General", dow$ & "Hours", "Off", retstr$, Len(retstr$) - 1, AppDir
& "vmail.ini")
        s$ = Left(retstr$, res%)
        If s$ = "Off" Then
          OleLog = "Closed hours,"
          res% = GetPrivateProfileString("Transfer", "ClosedTransfer", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
          If UCase(Left(retstr$, res%)) = "YES" Then TransferHour = True
        ElseIf Len(s$) > 12 Then
          If Time < CDate(Mid(s$, 1, 5)) Or Time > CDate(Mid(s$, 19, 5)) Then
            OleLog = "Closed hours,"
            res% = GetPrivateProfileString("Transfer", "ClosedTransfer", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
            If UCase(Left(retstr$, res%)) = "YES" Then TransferHour = True
          ElseIf Time > CDate(Mid(s$, 7, 5)) And Time < CDate(Mid(s$, 13, 5)) Then
```

```
        OleLog = "Lunch hours,"
        res% = GetPrivateProfileString("Transfer", "LunchTransfer", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
        If UCase(Left(retstr$, res%)) = "YES" Then TransferHour = True
       Else
        OleLog = "Open hours,"
        res% = GetPrivateProfileString("Transfer", "OpenTransfer", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
        If UCase(I eft(retstr$, res%)) = "YES" Then TransferHour = True
       End If
      Else
       If Time < CDate(Mid(s$, 1, 5)) Or Time > CDate(Mid(s$, 7, 5)) Then
        OleLog = "Closed hours,"
        res% = GetPrivateProfileString("Transfer", "ClosedTransfer", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
        If UCase(Left(retstr$, res%)) = "YES" Then TransferHour = True
       Else
        OleLog = "Open hours,"
        res% = GetPrivateProfileString("Transfer", "OpenTransfer", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
        If UCase(Left(retstr$, res%)) = "YES" Then TransferHour = True
       End If
      End If
     End If

     If TransferHour = True Then
      OleLog = OleLog & " (Hour)Transfer set to Yes"
      Call vcfContinue("AnnounceYourName")
     Else
      OleLog = OleLog & " (Hour)Transfer set to No"
      Call vcfContinue("GetPlayPersonalGreeting")
     End If
    Else
```

```
      OleLog = "AllowTransfer set to No"
      Call vcfContinue("GetPlayPersonalGreeting")
     End If
    End If
'********************** Get Announce Your Name
  Case "AnnounceYourName"
    res% = GetPrivateProfileString("Transfer", "PlayTransferringTo", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
    If UCase(Left(retstr$, res%)) = "YES" Then
     If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.raw") Then
      For i = 1 To VoiceLines%
       If FilePlaying$(i) = AppDir & "mboxes\" & Mailbox$ & "\prompt.wav" Or FilePlaying$(i) =
AppDir & "mboxes\" & Mailbox$ & "\prompt.raw" Then Exit For
      Next i
      If i > VoiceLines% Then
       Call CvtWaveFile(AppDir & "mboxes\" & Mailbox$ & "\prompt.raw", AppDir & "mboxes\" &
Mailbox$ & "\prompt.wav")
       OleLog = "converting prompt and "
      End If
     End If
     If isfile(AppDir & "mboxes\" & Mailbox$ & "\prompt.wav") Then
      OleLog = OleLog & "playing 'transferring to...' file"
      Call vcfPlayFile("AnnounceYourName2", "System", "xferto.wav")
     Else
      Call Jump("GetScreenCaller")
     End If
    Else
     Call Jump("GetScreenCaller")
    End If

  Case "AnnounceYourName2"
    OleLog = "playing prompt greeting"
    Call vcfPlayFile("GetScreenCaller", "Current", "prompt.wav")
```

```
'********************** Get Screen Caller
 Case "GetScreenCaller"
  res% = GetPrivateProfileString("Transfer", "ScreenCaller", "No", retstr$, Len(retstr$) - 1, Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then
   OleLog = "playing 'please record your name...' file"
   Call vcfPlayFile("RecordNameForScreenCaller", "System", "recname.wav")
  Else
   Call Jump("PlayPromptToAnnounceTransfer")
  End If

 Case "RecordNameForScreenCaller"
  OleLog = "recording caller name"
  Call vcfRecord("PlayPromptToAnnounceTransfer", "", App.Path & "\temp" & lineNo% & ".raw", 2)

'********************** Announce Transfer
 Case "PlayPromptToAnnounceTransfer"
  res% = GetPrivateProfileString("Transfer", "PlayPleaseHold", "Yes", retstr$, Len(retstr$) - 1, Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then
   OleLog = "playing 'please hold...' file"
   Call vcfPlayFile("GetDtmfForTransferBypass", "System", "plshold.wav")
  Else
   Call Jump("GetDtmfForTransferBypass")
  End If

'********************** Transfer Bypass
 Case "GetDtmfForTransferBypass"
  res% = GetPrivateProfileString("Transfer", "AllowTransferBypass", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then
   OleLog = "getting DTMF input for transfer bypass"
   Call vcfGetDigits("CheckDtmfForTransferBypass", 1, 1)
  Else
   Call Jump("GetFormQueue")
```

```
  End If

 Case "CheckDtmfForTransferBypass"
  If digits$ = "" Or InStr(digits$, "F") Then Call Jump("GetFormQueue") Else Call Jump
("GetPlayPersonalGreeting")

'********************** Get Form Queue
 Case "GetFormQueue"
  res% = GetPrivateProfileString("Transfer", "FormQueue", "No", retstr$, Len(retstr$) - 1, Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then Call Jump("AddToQueue") Else Call Jump
("StartTransfer")

 Case "AddToQueue"
  Queue(lineNo%).Mailbox = Mailbox$
  Queue(lineNo%).TimeAdded = Now
  counter%(lineNo%) = 0
  For i = 1 To VoiceLines%
   If Queue(i).Mailbox = Mailbox$ And i <> lineNo% Then
    If Queue(i).TimeAdded < Queue(lineNo%).TimeAdded Then
     counter%(lineNo%) = counter%(lineNo%) + 1
    Else
     Queue(lineNo%).TimeAdded = DateAdd("s", 1, Queue(lineNo%).TimeAdded)
    End If
   End If
  Next i
  If counter%(lineNo%) > 0 Then Call Jump("GetPlayQueuePosition") Else Call Jump("StartTransfer")

'---------------------------- Get Play Queue Position
 Case "GetPlayQueuePosition"
  res% = GetPrivateProfileString("Transfer", "PlayQueuePosition", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then Call Jump("PlayQueuePosition1") Else Call Jump
("GetOnWaitingInQueue")
```

```
Case "PlayQueuePosition1"
  OleLog = "playing 'there are...' file"
  Call vcfPlayFile("PlayQueuePosition2", "System", "ThereAre.wav")

 Case "PlayQueuePosition2"
  OleLog = "playing number of callers ahead in queue"
  Call ssfPlayNumber("PlayQueuePosition3", CLng(counter%(lineNo%)))

 Case "PlayQueuePosition3"
  OleLog = "playing 'callers ahead in queue...' file"
  Call vcfPlayFile("GetOnWaitingInQueue", "System", "Ahead.wav")

'---------------------------- Get On Waiting In Queue
 Case "GetOnWaitingInQueue"
  res% = GetPrivateProfileString("Transfer", "OnWaitingInQueue", "GoToLeaveMessage", retstr$, Len
(retstr$) - 1, Profile$)
  If UCase(Left(retstr$, res%)) = "1TOLEAVEMESSAGE" Then
   Call Jump("Play1ToLeaveMessage")
  ElseIf UCase(Left(retstr$, res%)) = "1TOHOLD" Then
   Call Jump("Play1ToHold")
  ElseIf UCase(Left(retstr$, res%)) = "HOLD" Then
   Call Jump("PlayHoldInQueue")
  Else     'If UCase(Left(retstr$, res%)) = "GOTOLEAVEMESSAGE" Then
   Call Jump("GetPlayPersonalGreeting")
  End If

'---------------------------- 1 to Quit Queue and Leave a Message
 Case "Play1ToLeaveMessage"
  OleLog = "playing 'press 1 to leave message...' file"
  Call vcfPlayFile("GetDtmf1ToLeaveMessage", "System", "1tolvmsg.wav")

 Case "GetDtmf1ToLeaveMessage"
  OleLog = "getting DTMF input for option to leave message"
```

```
  Call vcfGetDigits("CheckDtmf1ToLeaveMessage", 1, 6)

 Case "CheckDtmf1ToLeaveMessage"
  If digits$ = "1" Then Call Jump("GetPlayPersonalGreeting") Else Call Jump("PlayHoldInQueue")

'---------------------------- 1 to Hold in Queue
 Case "Play1ToHold"
  OleLog = "playing 'press 1 to hold...' file"
  Call vcfPlayFile("GetDtmf1ToHold", "System", "1tohold.wav")

 Case "GetDtmf1ToHold"
  OleLog = "getting DTMF input for option to hold"
  Call vcfGetDigits("CheckDtmf1ToHold", 1, 3)

 Case "CheckDtmf1ToHold"
  If digits$ = "1" Then Call Jump("PlayHoldInQueue") Else Call Jump("GetPlayPersonalGreeting")

'---------------------------- Holding in Queue
 Case "PlayHoldInQueue"
  OleLog = "playing 'please hold...' file"
  Call vcfPlayFile("HookFlashToHoldInQueue", "System", "plshold.wav")

 Case "HookFlashToHoldInQueue"
  res% = GetPrivateProfileInt("Transfer", "QueueHoldTime", 30, Profile$)
  lineTime(lineNo%) = DateAdd("s", res%, Now)
  OleLog = "dialing hook flash to hold in queue"
  Call vcfDial("Hold5SecInQueue", "&")

 Case "Hold5SecInQueue"
  OleLog = "delay 5 seconds in queue"
  Call vcfDelay("CheckOnQueuePosition", 5)

 Case "CheckOnQueuePosition"
```

```
  counter%(lineNo%) = 0    'counter is to count the number of callers ahead in queue
  For i = 1 To VoiceLines%
   If Queue(i).Mailbox = Mailbox$ And i <> lineNo% Then
    If Queue(i).TimeAdded < Queue(lineNo%).TimeAdded Then counter%(lineNo%) = counter%
(lineNo%) + 1
   End If
  Next i
  If counter%(lineNo%) = 0 Then
   OleLog = "dialing hook flash to finish holding in queue"
   Call vcfDial("StartTransfer", "&,,")
  Else
   If Now < lineTime(lineNo%) Then
    Call Jump("Hold5SecInQueue")
   Else
    OleLog = "dialing hook flash to return from holding in queue"
    Call vcfDial("GetPlayQueuePosition", "&,,")
   End If
  End If

'********************** Transfer to Extension
 Case "StartTransfer"
  counter%(lineNo%) = 0    'counter is to create the extension name for GetPrivateProfileString
  Call Jump("TransferToExt")

 Case "TransferToExt"
  counter%(lineNo%) = counter%(lineNo%) + 1

  res% = GetPrivateProfileString("Transfer", "TransferMethod", "PBX-Centrex", retstr$, Len(retstr$) - 1,
Profile$)
  If Left(retstr$, res%) = "NoPBX" Then
   If counter%(lineNo%) > 1 Then
    Call Jump("PlayNoTransfer")
   Else
```

```
      OleCallLog = "(" & Mailbox$ & ") Transferring "
      OleLog = "transferring without dialing extension..."     'used by status log
      counter2%(lineNo%) = 0      'counter2 is for RingsToWait
      Call vcfContinue("PlayPressKey")
     End If
    Else
     res% = GetPrivateProfileString("Transfer", "Extension" & counter%(lineNo%), "", retstr$, Len
(retstr$) - 1, Profile$)
     If Left(retstr$, res%) = "" Then
      Call Jump("PlayNoTransfer")
     Else
      dialstring = Left(retstr$, res%)
      res% = GetPrivateProfileString("General", "InitiateTransferSeq", "&,", retstr$, Len(retstr$) - 1,
App.Path & "\konnects.ini")
      dialstring = Left(retstr$, res%) & dialstring
      OleCallLog = "(" & Mailbox$ & ") Transfer to ext. '" & dialstring & "' "
    '  dialstring = "&," & dialstring

      res% = GetPrivateProfileString("Transfer", "CallProgressMonitor", "No", retstr$, Len(retstr$) - 1,
Profile$)
      If UCase(Left(retstr$, res%)) = "YES" Then
       OleLog = "CPM dialing extension " & dialstring & "..."     'used by status log
       res% = GetPrivateProfileString("Transfer", "RingsToWait", "4", retstr$, Len(retstr$) - 1, Profile$)
       Call vcfCallOut("GetCPMResult", dialstring, Val(Left(retstr$, res%)))
      Else
       OleLog = "dialing extension " & dialstring & "..."     'used by status log
       counter2%(lineNo%) = 0      'counter2 is for RingsToWait
       Call vcfDial("PlayPressKey", dialstring)
      End If

     End If
    End If
```

```
  Case "GetCPMResult"
    tmpCallState%(lineNo%) = callstate%
    If callstate% = 7 Or callstate% = 8 Or callstate% = 9 Then
      Select Case callstate%
      Case 7
        OleLog = "line busy, "
      Case 8
        OleLog = "no answer, "
      Case 9
        OleLog = "no dialtone or no ring/busy tone, "
      End Select
      OleCallLog = "failed. "
      res% = GetPrivateProfileString("General", "ReturnTransferSeq", "&,", retstr$, Len(retstr$) - 1,
App.Path & "\konnects.ini")
      dialstring = Left(retstr$, res%)
      OleLog = OleLog & "getting call back with " & dialstring
'     Call vcfDial("TransferToExt", "&,")
      Call vcfDial("TransferToExt", dialstring)
    Else
      res% = GetPrivateProfileString("General", "AnnounceCPMTransfer", "No", retstr$, Len(retstr$) - 1,
App.Path & "\konnects.ini")
      If UCase(Left(retstr$, res%)) = "YES" Then
        OleLog = "transfer answered, playing 'transferring call' file"
        Call vcfPlayFile("CPMConnected", "System", "xfercall.wav")
      Else
        OleLog = "transfer answered, hang up."
        OleCallLog = "completed. "
        Call vcfHangup
      End If
    End If

  Case "CPMConnected"
    OleLog = "transfer acknowleged, hang up."
```

```
  OleCallLog = "completed. "
  Call vcfHangup

 Case "PlayPressKey"
  res% = GetPrivateProfileString("Transfer", "BlindTransfer", "No", retstr$, Len(retstr$) - 1, Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then
   OleLog = "hangup for blind transfer"
   Call vcfHangup
  Else
   res% = GetPrivateProfileString("Transfer", "TransferMethod", "PBX-Centrex", retstr$, Len(retstr$) -
1, Profile$)
   If Left(retstr$, res%) = "NoPBX" Then
    OleLog = "playing on hold without PBX file nopbxhld.wav"
    Call vcfPlayFile("GetAcceptCallDTMF", "System", "nopbxhld.wav")
   Else
    OleLog = "playing prompt greeting"
    Call vcfPlayFile("PlayPressKey2", "Current", "prompt.wav")
   End If
  End If

 Case "PlayPressKey2"
  OleLog = "playing 'press a key to receive call ...' file"
  Call vcfPlayFile("PlayCallerName", "System", "prsstar.wav")

 Case "PlayCallerName"
  res% = GetPrivateProfileString("Transfer", "ScreenCaller", "No", retstr$, Len(retstr$) - 1, Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then
   OleLog = "playing caller name ..."
   Call vcfPlayFile("GetAcceptCallDTMF", "", App.Path & "\temp" & lineNo% & ".raw")
  Else
   Call Jump("GetAcceptCallDTMF")
  End If
```

```
 Case "GetAcceptCallDTMF"
  OleLog = "waiting for * digit..."
  Call vcfGetDigits("CheckAcceptCallDTMF", 1, 1)

 Case "CheckAcceptCallDTMF"
  If digits$ = "" Then
   counter2%(lineNo%) = counter2%(lineNo%) + 1
   res% = GetPrivateProfileInt("Transfer", "RingsToWait", 4, Profile$)
   If counter2%(lineNo%) < res% Then
    Call Jump("PlayPressKey")
   Else
    OleCallLog = "failed. "
    res% = GetPrivateProfileString("Transfer", "TransferMethod", "PBX-Centrex", retstr$, Len(retstr$) -
1, Profile$)
    If Left(retstr$, res%) = "NoPBX" Then
     OleLog = "no DTMF response on transfer..."
     Call vcfContinue("TransferToExt")
    Else
     res% = GetPrivateProfileString("General", "ReturnTransferSeq", "&,", retstr$, Len(retstr$) - 1,
App.Path & "\konnects.ini")
     dialstring = Left(retstr$, res%)
     OleLog = "no DTMF response on transfer, getting call back with " & dialstring
     Call vcfDial("TransferToExt", dialstring)
'    Call vcfDial("TransferToExt", "&,")
    End If
   End If
  Else
   OleCallLog = "completed. "
   res% = GetPrivateProfileString("Transfer", "TransferMethod", "PBX-Centrex", retstr$, Len(retstr$) -
1, Profile$)
   If Left(retstr$, res%) = "NoPBX" Then
    OleLog = "digit received, play nopbxend.wav file"
    Call vcfPlayFile("NoPBXEnd", "System", "nopbxend.wav")
```

```
    Else
      OleLog = "digit received, hangup"
      Call vcfHangup
    End If
  End If

 Case "NoPBXEnd"
  OleLog = "nopbxend.wav played, hangup"
  Call vcfHangup


'--------------------------- No Transfer Options
 Case "PlayNoTransfer"
  If tmpCallState%(lineNo%) = 7 Then
    OleLog = "playing 'busy...' file"
    Call vcfPlayFile("GetOnNoTransfer", "System", "busy.wav")
  Else
    OleLog = "playing 'no answer...' file"
    Call vcfPlayFile("GetOnNoTransfer", "System", "noanswer.wav")
  End If

 Case "GetOnNoTransfer"
  If tmpCallState%(lineNo%) = 7 Then
    res% = GetPrivateProfileString("Transfer", "OnBusy", "", retstr$, Len(retstr$) - 1, Profile$)
  Else
    res% = GetPrivateProfileString("Transfer", "OnNoAnswer", "", retstr$, Len(retstr$) - 1, Profile$)
  End If
  If Left(retstr$, res%) = "" Then res% = GetPrivateProfileString("Transfer", "OnNoTransfer",
"GoToLeaveMessage", retstr$, Len(retstr$) - 1, Profile$)

  If UCase(Left(retstr$, res%)) = "1TOHOLD" Then
    Call Jump("Play1ToHoldT")
  ElseIf UCase(Left(retstr$, res%)) = "1TOLEAVEMESSAGE" Then
```

```
    Call Jump("Play1ToLeaveMessageT")
  ElseIf UCase(Left(retstr$, res%)) = "HOLD" Then
    Call Jump("StartTransfer")
  Else ' If UCase(Left(retstr$, res%)) = "GOTOLEAVEMESSAGE" Then
    If tmpCallState%(lineNo%) = 7 Then
      OleLog = "busy,"
      Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Busy'"
    Else
      OleLog = "no answer,"
      Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'NoAnswer'"
    End If

    If Not Tonetables.NoMatch Then
      Mailbox$ = Tonetables("Assignment")
      OleLog = OleLog & " going to " & Mailbox$
      Call vcfGoto(Mailbox$)
    Else
      Call Jump("GetPlayBusyNoAnswerGreeting")
    End If
  End If

'---------------------------- 1 to Quit Transfer and Leave Message
  Case "Play1ToLeaveMessageT"
    OleLog = "playing 'press 1 to leave message...' file"
    Call vcfPlayFile("GetDtmf1ToLeaveMessageT", "System", "1tolvmsg.wav")

  Case "GetDtmf1ToLeaveMessageT"
    OleLog = "getting DTMF input for option to leave message"
    Call vcfGetDigits("CheckDtmf1ToLeaveMessageT", 1, 6)

  Case "CheckDtmf1ToLeaveMessageT"
    If digits$ = "1" Then
      Call Jump("GetPlayBusyNoAnswerGreeting")
```

```
  Else
   OleLog = "1 not received, playing 'please hold...' file"
   Call vcfPlayFile("StartTransfer", "System", "plshold.wav")
  End If

'--------------------------- 1 to Hold in Transfer
 Case "Play1ToHoldT"
  OleLog = "playing 'press 1 to hold...' file"
  Call vcfPlayFile("GetDtmf1ToHoldT", "System", "1tohold.wav")

 Case "GetDtmf1ToHoldT"
  OleLog = "getting DTMF input for option to hold"
  Call vcfGetDigits("CheckDtmf1ToHoldT", 1, 3)

 Case "CheckDtmf1ToHoldT"
  If digits$ = "1" Then
   OleLog = "1 received, playing 'please hold...' file"
   Call vcfPlayFile("StartTransfer", "System", "plshold.wav")
  Else
   Call Jump("GetPlayBusyNoAnswerGreeting")
  End If

'********************** Busy/No-Answer Greeting
 Case "GetPlayBusyNoAnswerGreeting"
  Queue(lineNo%).Mailbox = ""

  If tmpCallState%(lineNo%) = 7 Then
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\Busy.wav") Then
    OleLog = OleLog & "playing busy personal greeting"
    Call vcfPlayFile("GetAllowBusyGotoTones", "Current", "busy.wav")
   Else
    Call Jump("GetPlayPersonalGreeting")
   End If
```

```
  Else
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\NoAnswer.wav") Then
    OleLog = OleLog & "playing no answer personal greeting"
    Call vcfPlayFile("GetAllowNoAnswerGotoTones", "Current", "noanswer.wav")
   Else
    Call Jump("GetPlayPersonalGreeting")
   End If
  End If

 Case "GetAllowBusyGotoTones"
  res% = GetPrivateProfileString("Message", "AllowBusyGotoTones", "No", retstr$, Len(retstr$) - 1,
Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then
   OleLog = "getting DTMF input for busy goto"
   Call vcfGetDigits("CheckDtmfBusyGotoTones", 1, 3)
  Else
   Call Jump("GetPlayPersonalGreeting")
  End If

 Case "CheckDtmfBusyGotoTones"
  res% = GetPrivateProfileString("Message", "BusyGotoTone" & digits$, "", retstr$, Len(retstr$) - 1,
Profile$)
  If Left(retstr$, res%) = "" Then
   If digits$ = "" Then
    Call Jump("GetPlayPersonalGreeting")
   Else
    Call vcfPlayFile("GetPlayBusyNoAnswerGreeting", "System", "invalid.wav")
   End If
  Else

   If UCase(Left(retstr$, res%)) = "M" Then
    Call Jump("GetPlayPersonalGreeting")
   Else
```

```
    Mailboxes.FindFirst "MailboxNumber='" & Left(retstr$, res%) & "'"
    If Not Mailboxes.NoMatch Then
     OleLog = "Busy goto tone " & digits$ & " entered, going to " & Left(retstr$, res%)
     Call vcfGoto(Left(retstr$, res%))
    Else
     Call vcfPlayFile("GetPlayBusyNoAnswerGreeting", "System", "invalid.wav")
    End If
   End If
  End If

 Case "GetAllowNoAnswerGotoTones"
  res% = GetPrivateProfileString("Message", "AllowNoAnswerGotoTones", "No", retstr$, Len(retstr$) -
1, Profile$)
  If UCase(Left(retstr$, res%)) = "YES" Then
   OleLog = "getting DTMF input for NoAnswer goto"
   Call vcfGetDigits("CheckDtmfNoAnswerGotoTones", 1, 3)
  Else
   Call Jump("GetPlayPersonalGreeting")
  End If

 Case "CheckDtmfNoAnswerGotoTones"
  res% = GetPrivateProfileString("Message", "NoAnswerGotoTone" & digits$, "", retstr$, Len(retstr$) -
1, Profile$)
  If Left(retstr$, res%) = "" Then
   If digits$ = "" Then
    Call Jump("GetPlayPersonalGreeting")
   Else
    Call vcfPlayFile("GetPlayBusyNoAnswerGreeting", "System", "invalid.wav")
   End If
  Else

   If UCase(Left(retstr$, res%)) = "M" Then
    Call Jump("GetPlayPersonalGreeting")
```

```
    Else
     Mailboxes.FindFirst "MailboxNumber='" & Left(retstr$, res%) & "'"
     If Not Mailboxes.NoMatch Then
      OleLog = "NoAnswer goto tone " & digits$ & " entered, going to " & Left(retstr$, res%)
      Call vcfGoto(Left(retstr$, res%))
     Else
      Call vcfPlayFile("GetPlayBusyNoAnswerGreeting", "System", "invalid.wav")
     End If
    End If
   End If


'********************** Personal Greeting
  Case "GetPlayPersonalGreeting"
'   Queue(lineNo%).Mailbox = ""
   res% = GetPrivateProfileString("Message", "PlayPersonalGreeting", "Yes", retstr$, Len(retstr$) - 1,
Profile$)
   If UCase(Left(retstr$, res%)) = "YES" Then Call Jump("PlayPersonalGreeting") Else Call Jump
("GetTakePhoneNumber")


  Case "PlayPersonalGreeting"
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\personal.raw") Then
    For i = 1 To VoiceLines%
     If FilePlaying$(i) = AppDir & "mboxes\" & Mailbox$ & "\personal.wav" Or FilePlaying$(i) =
AppDir & "mboxes\" & Mailbox$ & "\personal.raw" Then Exit For
    Next i
    If i > VoiceLines% Then
     Call CvtWaveFile(AppDir & "mboxes\" & Mailbox$ & "\personal.raw", AppDir & "mboxes\" &
Mailbox$ & "\personal.wav")
     OleLog = "converting and "
    End If
   End If
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\personal.wav") Then
```

```
    OleLog = OleLog & "playing personal greeting"
    Call vcfPlayFile("GetAllowGotoTones", "Current", "personal.wav")
   Else
    Call Jump("GetAllowGotoTones")
   End If

'---------------------------- Goto Tones
  Case "GetAllowGotoTones"
   res% = GetPrivateProfileString("Message", "AllowGotoTones", "No", retstr$, Len(retstr$) - 1, Profile$)
   If UCase(Left(retstr$, res%)) = "YES" Then
    OleLog = "getting DTMF input for goto"
    Call vcfGetDigits("CheckDtmfGotoTones", 1, 3)
   Else
    Call Jump("GetTakePhoneNumber")
   End If

  Case "CheckDtmfGotoTones"
   res% = GetPrivateProfileString("Message", "GotoTone" & digits$, "", retstr$, Len(retstr$) - 1, Profile$)

   If Left(retstr$, res%) = "" Then
    If digits$ = "" Then
     Call Jump("GetTakePhoneNumber")
    Else
     Call vcfPlayFile("PlayPersonalGreeting", "System", "invalid.wav")
    End If
   Else
    If UCase(Left(retstr$, res%)) = "M" Then
     Call Jump("GetTakePhoneNumber")
    Else
     Mailboxes.FindFirst "MailboxNumber='" & Left(retstr$, res%) & "'"
     If Not Mailboxes.NoMatch Then
      OleLog = "Goto tone " & digits$ & " entered, going to " & Left(retstr$, res%)
      Call vcfGoto(Left(retstr$, res%))
```

```
      Else
       Call vcfPlayFile("PlayPersonalGreeting", "System", "invalid.wav")
'      Call Jump("GetTakePhoneNumber")
      End If
     End If
    End If

'********************** Take Phone Number
  Case "GetTakePhoneNumber"
    RecFileName$(lineNo%) = GetRecFileName$(Mailbox$)
    res% = GetPrivateProfileString("Message", "TakePhoneNumber", "No", retstr$, Len(retstr$) - 1,
Profile$)
    If UCase(Left(retstr$, res%)) = "YES" Then
     OleLog = "flushing digits..."
     Call vcfFlushDigits("PlayEnterPhone")
    Else
     Call Jump("GetTakeMessage")
    End If

  Case "PlayEnterPhone"
    OleLog = "playing 'please enter phone number...' file"
    Call vcfPlayFile("GetDtmfPhoneNumber", "System", "enterphn.wav")

  Case "GetDtmfPhoneNumber"
    OleLog = "getting phone number DTMF"
    Call vcfGetDigits("WritePhoneNumber", 15, 3)

  Case "WritePhoneNumber"
'   OleLog = "digits " & digits$ & " received, "
    If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$, - 1)
    If digits$ <> "" Then
     res% = WritePrivateProfileString("General", "PhoneNumber", digits$, Left(RecFileName$(lineNo%),
Len(RecFileName$(lineNo%)) - 3) & "inf")
```

```
    res% = WritePrivateProfileString("General", "CallerID", CallerID$, Left(RecFileName$(lineNo%),
Len(RecFileName$(lineNo%)) - 3) & "inf")
    res% = GetPrivateProfileString("General", "State", "", retstr$, Len(retstr$) - 1, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 12) & "status")
    mState$ = Str(Val(Left(retstr$, res%)) + 1)
    If mState$ > "30000" Then mState$ = "0"
    res% = WritePrivateProfileString("General", "State", mState$, Left(RecFileName$(lineNo%), Len
(RecFileName$(lineNo%)) - 12) & "status")

   End If
   Call Jump("GetTakeMessage")

'********************** Recording Message
  Case "GetTakeMessage"
   res% = GetPrivateProfileString("Message", "TakeMessage", "Yes", retstr$, Len(retstr$) - 1, Profile$)
   If UCase(Left(retstr$, res%)) = "YES" Then
    OleLog = "play 'please record...' file"
    Call vcfPlayFile("FlushBeforeRecording", "System", "recgrt.wav")
   Else
    Call Jump("End")
   End If

  Case "FlushBeforeRecording"
   OleLog = "flushing digits..."
   Call vcfFlushDigits("RecordMessage")

  Case "RecordMessage"
   OleLog = "recording message"   'used by status log
   OleCallLog = "(" & Mailbox$ & ") Message recorded. "
   res% = GetPrivateProfileInt("Message", "MaxMessageLength", 60, Profile$)
   Call vcfRecord("FlushAfterRecording", "", RecFileName$(lineNo%), CLng(res%))

  Case "FlushAfterRecording"
```

```
  OleLog = "flushing digits..."
  Call vcfFlushDigits("PlayRecordOptions")

'---------------------------- Record Options
 Case "PlayRecordOptions"
  OleLog = "1 to play, 2 record again, 3 to delete..."
  Call vcfPlayFile("GetDTMFRecordOptions", "System", "recmenu.wav")

 Case "GetDTMFRecordOptions"
  OleLog = "getting digits..."
  Call vcfGetDigits("CheckDTMFRecordOptions", 1, 3)

 Case "CheckDTMFRecordOptions"
  OleLog = "digits '" & digits$ & "' received..."
  Select Case digits$
  Case "1"
   OleLog = OleLog & "play recorded message"
   Call vcfPlayFile("PlayRecordOptions", "", RecFileName(lineNo%))
  Case "2"
   On Error Resume Next
   If isfile(RecFileName(lineNo%)) Then Kill RecFileName(lineNo%)
   Call Jump("PlayPleaseRecordAgain")
  Case "3"
   OleLog = OleLog & "play 'message deleted...' file"
   On Error Resume Next
   If isfile(RecFileName(lineNo%)) Then Kill RecFileName(lineNo%)
   Call vcfPlayFile("End", "System", "recdel.wav")
  Case "4"
   OleLog = OleLog & "play 'message saved' file"
   Call vcfPlayFile("End", "System", "recsave.wav")
  Case Else
   OleLog = OleLog & "play 'message saved' file"
   Call vcfPlayFile("End", "System", "recsave.wav")
```

```
  End Select

 Case "PlayPleaseRecordAgain"
  OleLog = "play 'please record...' file"
  Call vcfPlayFile("RecordMessage", "System", "recgrt.wav")

'********************** End
 Case "End"
  If RecFileName$(lineNo%) <> "" Then

   If isfile(RecFileName(lineNo%)) Then
    OleCallLog = "Length " & FileLen(RecFileName$(lineNo%)) \ 8000 & " sec."
    If FileLen(RecFileName$(lineNo%)) < GetPrivateProfileInt("General", "MinMessageLength", 4,
App.Path & "\konnects.ini") * 8000& Then
     On Error Resume Next
     Kill RecFileName(lineNo%)
    Else
     Call CvtWaveFile(RecFileName$(lineNo%), Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "wav")
     If Not isfile(Left(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf") Then
      res% = WritePrivateProfileString("General", "PhoneNumber", "", Left(RecFileName$(lineNo%),
Len(RecFileName$(lineNo%)) - 3) & "inf")
      res% = WritePrivateProfileString("General", "CallerID", CallerID$, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf")
      res% = GetPrivateProfileString("General", "State", "", retstr$, Len(retstr$) - 1, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 12) & "status")
      mState$ = Str(Val(Left(retstr$, res%)) + 1)
      If mState$ > "30000" Then mState$ = "0"
      res% = WritePrivateProfileString("General", "State", mState$, Left(RecFileName$(lineNo%), Len
(RecFileName$(lineNo%)) - 12) & "status")
     End If

     res% = GetPrivateProfileString("Paging", "EnablePaging", "No", retstr$, Len(retstr$) - 1, Profile$)
```

```
If UCase(Left(retstr$, res%)) = "YES" Then
 If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then
  Schedule.AddNew
  Schedule("TimeScheduled") = DateAdd("s", 10, Now)
  Schedule("TimeInserted") = Now
  Schedule("Status") = "Pending"   'status$
  Schedule("Action") = "Page"
  res% = GetPrivateProfileString("Paging", "PhoneNumber", "", retstr$, Len(retstr$) - 1, Profile$)
  Schedule("PhoneNumber") = Left(retstr$, res%)
  res% = GetPrivateProfileString("Paging", "PagerString", "", retstr$, Len(retstr$) - 1, Profile$)
  s$ = Left(retstr$, res%)
  res% = GetPrivateProfileString("General", "PhoneNumber", "", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf")
  Schedule("PagerString") = s$ & "," & Left(retstr$, res%) & "#"
  Schedule("Attempts") = 3
  Schedule("Name") = ""
  Schedule("OwnerMailbox") = Mailbox$
  Schedule("AttemptsMade") = 0
  Schedule.Update
 End If
End If

res% = GetPrivateProfileString("Notification", "EnableNotification", "No", retstr$, Len(retstr$) - 1,
Profile$)
If UCase(Left(retstr$, res%)) = "YES" Then
 If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then
  Schedule.AddNew
  Schedule("TimeScheduled") = DateAdd("n", 5, Now)
  Schedule("TimeInserted") = Now
  Schedule("Status") = "Pending"   'status$
  Schedule("Action") = "Notify"
```

```
        res% = GetPrivateProfileString("Notification", "PhoneNumber", "", retstr$, Len(retstr$) - 1,
Profile$)
        Schedule("PhoneNumber") = Left(retstr$, res%)
        Schedule("Attempts") = 2
        Schedule("Name") = ""
        Schedule("OwnerMailbox") = Mailbox$
        Schedule("AttemptsMade") = 0
        Schedule.Update
       End If
      End If

      res% = GetPrivateProfileString("Message Lamp", "EnableMessageLamp", "No", retstr$, Len
(retstr$) - 1, Profile$)
      If UCase(Left(retstr$, res%)) = "YES" Then
       If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then

        Schedule.FindFirst "OwnerMailbox = '" & Mailbox$ & "'  and Action = 'Lamp' and Status =
'Pending'"
        If Schedule.NoMatch Then Schedule.AddNew Else Schedule.Edit
        Schedule("TimeScheduled") = DateAdd("s", 10, Now)
        Schedule("TimeInserted") = Now
        Schedule("Status") = "Pending"   'status$
        Schedule("Action") = "Lamp"
        res% = GetPrivateProfileString("Message Lamp", "DialString", "", retstr$, Len(retstr$) - 1,
Profile$)
        Schedule("PhoneNumber") = Left(retstr$, res%)
        Schedule("Attempts") = GetPrivateProfileInt("Message Lamp", "Attempts", 1, Profile$)
        Schedule("Name") = ""
        Schedule("OwnerMailbox") = Mailbox$
        Schedule("AttemptsMade") = 0
        Schedule.Update
       End If
```

```
        End If


       End If
      End If


      RecFileName$(lineNo%) = ""
     End If

     OleLog = "finished transfer function,"
     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
     If Not Tonetables.NoMatch Then
      Mailbox$ = Tonetables("Assignment")
      OleLog = OleLog & " going to " & Mailbox$
      Call vcfGoto(Mailbox$)
     Else
      OleLog = OleLog & " going to Hangup"
      Call vcfGoto("Hangup")
     End If

    Case Else
     OleLog = "invalid state, going to Hangup"
     Call vcfGoto("Hangup")

    End Select

End Sub

Private Sub fDoRemote()

    Dim lineNo%, Mailbox$, digits$, nextstate$, i%, recording%
    Dim s$, current$, ltemp&
```

```
Dim res%, retstr As String * 255
Dim ATime As Date, Filename$, Profile$
Dim MsgPath$
Dim mState$

Static tmpInput$(1 To 74)

lineNo% = OleData(0)
Mailbox$ = OleData(1)
digits$ = OleData(2)
Profile$ = AppDir & "mboxes\" & temp$(lineNo%) & "\transfer.ini"

If State$(lineNo%) = "" Then
  State$(lineNo%) = "MB0"
End If

Select Case State$(lineNo%)

Case "MB0"
  counter%(lineNo%) = 0
  Call Jump("MB1")
Case "MB1"
  OleLog = "playing 'please enter mailbox #...' file"
  Call vcfPlayFile("MB2", "System", "entermb.wav")
Case "MB2"
  OleLog = "getting DTMF for mailbox #"
  Call vcfGetDigits("MB3", 9, 3)
Case "MB3"
  If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
  Mailboxes.FindFirst "MailBoxNumber='" & digits$ & "'"
  If Not Mailboxes.NoMatch And (Mailboxes("Function") = "DoTransfer" Or Mailboxes("Function") =
"PlayClassifiedAd") Then
    temp$(lineNo%) = digits$
```

```
    On Error Resume Next
    If Not isfile(AppDir & "mboxes\" & temp$(lineNo%) & "\transfer.ini") Then FileCopy App.Path &
"\transfer.ini", AppDir & "mboxes\" & temp$(lineNo%) & "\transfer.ini"
    Call Jump("PW1")
   Else
    OleLog = "invalid digits '" & digits$ & "' received, playing 'invalid input...' file"
    counter(lineNo%) = counter(lineNo%) + 1
    If counter(lineNo%) = 3 Then nextstate$ = "EXIT" Else nextstate$ = "MB1"
    Call vcfPlayFile(nextstate$, "System", "invalid.wav")
   End If

  Case "PW1"
   OleLog = "playing 'please enter password...' file"
   Call vcfPlayFile("PW2", "System", "enterpw.wav")
  Case "PW2"
   OleLog = "getting DTMF for password"
   Call vcfGetDigits("PW3", 15, 3)
  Case "PW3"
   If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
'  Mailboxes.FindFirst "MailBoxNumber='" & input1$(lineNo%) & "'"
   res% = GetPrivateProfileString("General", "Password", "", retstr$, Len(retstr$) - 1, Profile$)
   If Left(retstr$, res%) = digits$ Then
    Call Jump("PLAY_#OF_NEW_MSG1")
   Else
    OleLog = "invalid password received, playing 'invalid input...' file"
    counter(lineNo%) = counter(lineNo%) + 1
    If counter(lineNo%) = 3 Then nextstate$ = "EXIT" Else nextstate$ = "MB1"
    Call vcfPlayFile(nextstate$, "System", "invalid.wav")
   End If

  Case "PLAY_#OF_NEW_MSG1"
   On Error Resume Next
   MkDir AppDir & "mboxes\" & temp$(lineNo%) & "\msg"
```

```
On Error GoTo 0

MsgPath$ = AppDir & "mboxes\" & temp$(lineNo%) & "\msg\"
counter%(lineNo%) = 0
s$ = Dir(MsgPath$ & "*.i*")
While s$ <> ""
  If Right(s$, 1) <> "_" Then
    counter%(lineNo%) = counter%(lineNo%) + 1
  ElseIf Right(s$, 2) <> "__" Then
    On Error Resume Next
    Name MsgPath$ & s$ As MsgPath$ & Left(s$, 9) & "i__"
    On Error GoTo 0
  End If
  s$ = Dir
Wend
If counter%(lineNo%) = 0 Then
  OleLog = "playing 'no new messages...' file"
  Call vcfPlayFile("MENU0", "System", "nonewmsg.wav")
Else
  OleLog = "playing 'you have...' file"
  Call vcfPlayFile("PLAY_#OF_NEW_MSG2", "System"  "youhave.wav")
End If
Case "PLAY_#OF_NEW_MSG2"
  OleLog = "playing number of messages (" & counter(lineNo%) & ")"
  Call ssfPlayNumber("PLAY_#OF_NEW_MSG3", CLng(counter(lineNo%)))
Case "PLAY_#OF_NEW_MSG3"
  If counter(lineNo%) = 1 Then
    OleLog = "playing 'new message...' file"
    Call vcfPlayFile("MENU0", "System", "newmsg.wav")
  Else
    OleLog = "playing 'new messages...' file"
    Call vcfPlayFile("MENU0", "System", "newmsgs.wav")
  End If
```

```
Case "MENU0"
   counter%(lineNo%) = 0
   OleLog = "flushing digits"
   Call vcfFlushDigits("MENU1")
  Case "MENU1"
   OleLog = "playing remote menu file"
   Call vcfPlayFile("MENU2", "System", "remmenu.wav")
  Case "MENU2"
   OleLog = "getting DTMF for remote menu"
   Call vcfGetDigits("MENU3", 1, 3)
  Case "MENU3"
   Select Case digits$
   Case "1"
    temp1$(lineNo%) = "New"
    Call Jump("PLAY_MESSAGE_INSTRUCTIONS")
   Case "2"
    temp1$(lineNo%) = "Old"
    Call Jump("PLAY_MESSAGE_INSTRUCTIONS")

   Case "3"
    If isfile(AppDir & "mboxes\" & temp$(lineNo%) & "\personal.raw") Then
     For i = 1 To VoiceLines%
      If FilePlaying$(i) = AppDir & "mboxes\" & temp$(lineNo%) & "\personal.wav" Or FilePlaying$(i)
= AppDir & "mboxes\" & temp$(lineNo%) & "\personal.raw" Then Exit For
     Next i
     If i > VoiceLines% Then
      Call CvtWaveFile(AppDir & "mboxes\" & temp$(lineNo%) & "\personal.raw", AppDir &
"mboxes\" & temp$(lineNo%) & "\personal.wav")
      OleLog = "converting and "
     End If
    End If
    OleLog = OleLog & "playing personal greeting"
    Call vcfPlayFile("MENU0", "", AppDir & "mboxes\" & temp$(lineNo%) & "\personal.wav")
```

```
  Case "4"
'   OleLog = "recording personal greeting"
'   Call vcfRecord("MENU0", "", AppDir & "mboxes\" & temp$(lineNo%) & "\personal.raw", 90)
    OleLog = "play 'record personal greeting' file"
    Call vcfPlayFile("RecordPersonal", "System", "recpersn.wav")

  Case "5"
    Call Jump("SUBMENU0")
  Case "#"
    Call Jump("EXIT")
  Case "*"
    Call Jump("MB0")
  Case Else
    OleLog = "digits " & digits$ & " received, playing 'invalid input...' file"
    counter(lineNo%) = counter(lineNo%) + 1
    If counter(lineNo%) = 3 Then nextstate$ = "EXIT" Else nextstate$ = "MENU1"
    Call vcfPlayFile(nextstate$, "System", "invalid.wav")
  End Select

 Case "RecordPersonal"
  res% = GetPrivateProfileInt("Message", "MaxPersonalLength", 120, AppDir & "mboxes\" & temp$
(lineNo%) & "\transfer.ini")
  OleLog = "recording personal greeting"
  Call vcfRecord("MENU0", "", AppDir & "mboxes\" & temp$(lineNo%) & "\personal.raw", CLng
(res%))

 Case "SUBMENU0"
  counter%(lineNo%) = 0
  OleLog = "flushing digits"
  Call vcfFlushDigits("SUBMENU1")
 Case "SUBMENU1"
  OleLog = "playing remote submenu file"
  Call vcfPlayFile("SUBMENU2", "System", "remsubmn.wav")
```

```
  Case "SUBMENU2"
    OleLog = "getting DTMF for remote menu"
    Call vcfGetDigits("SUBMENU3", 1, 3)
  Case "SUBMENU3"
    Select Case digits$

    Case "1"
     If isfile(AppDir & "mboxes\" & temp$(lineNo%) & "\prompt.raw") Then
      For i = 1 To VoiceLines%
       If FilePlaying$(i) = AppDir & "mboxes\" & temp$(lineNo%) & "\prompt.wav" Or FilePlaying$(i)
= AppDir & "mboxes\" & temp$(lineNo%) & "\prompt.raw" Then Exit For
      Next i
      If i > VoiceLines% Then
       Call CvtWaveFile(AppDir & "mboxes\" & temp$(lineNo%) & "\prompt.raw", AppDir & "mboxes\"
& temp$(lineNo%) & "\prompt.wav")
       OleLog = "converting and "
      End If
     End If
     OleLog = OleLog & "playing prompt greeting"
     Call vcfPlayFile("MENU0", "", AppDir & "mboxes\" & temp$(lineNo%) & "\prompt.wav")
    Case "2"
'    OleLog = "recording prompt greeting"
'    Call vcfRecord("MENU0", "", AppDir & "mboxes\" & temp$(lineNo%) & "\prompt.raw", 90)
     OleLog = "play 'record name' file"
     Call vcfPlayFile("RecordName", "System", "recprmt.wav")

    Case "3"
     res% = WritePrivateProfileString("Transfer", "AllowTransfer", "Yes", Profile$)
     OleLog = "playing 'transfer enabled' file"
     Call vcfPlayFile("MENU0", "System", "enablet.wav")
    Case "4"
     res% = WritePrivateProfileString("Transfer", "AllowTransfer", "No", Profile$)
     OleLog = "playing 'transfer disabled' file"
```

```
    Call vcfPlayFile("MENU0", "System", "disablet.wav")

  Case "5"
    OleLog = "playing 'enter new password' file"
    Call vcfPlayFile("NEWPW0", "System", "enternpw.wav")

  Case "6"
    counter(lineNo%) = 0
    Call Jump("NEWPGR0")

  Case "7"
    counter(lineNo%) = 0
    Call Jump("NEWMNN0")

  Case "#"
    Call Jump("MENU0")

  Case Else
    OleLog = "digits " & digits$ & " received, playing 'invalid input...' file"
    counter(lineNo%) = counter(lineNo%) + 1
    If counter(lineNo%) = 2 Then nextstate$ = "MENU0" Else nextstate$ = "SUBMENU1"
    Call vcfPlayFile(nextstate$, "System", "invalid.wav")
  End Select

Case "RecordName"
  OleLog = "recording prompt greeting"
  Call vcfRecord("MENU0", "", AppDir & "mboxes\" & temp$(lineNo%) & "\prompt.raw", 30)



'*************************************************************************
' CASE "5" ---- NEW PASSWORD
'*************************************************************************
```

```
Case "NEWPW0"
  OleLog = "getting DTMF for new password"
  Call vcfGetDigits("NEWPW1", 15, 3)

 Case "NEWPW1"
  If digits$ = "" Then
   OleLog = "nothing entered, playing 'invalid' file"
   Call vcfPlayFile("MENU0", "System", "invalid.wav")
  Else
   If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
   temp1$(lineNo%) = digits$
   OleLog = "playing 'enter new password again' file"
   Call vcfPlayFile("NEWPW2", "System", "entrnpwa.wav")
  End If

 Case "NEWPW2"
  OleLog = "getting DTMF for new password verify"
  Call vcfGetDigits("NEWPW3", 15, 3)

 Case "NEWPW3"
  If digits$ = "" Then
   OleLog = "nothing entered for verification, playing 'invalid' file"
   Call vcfPlayFile("MENU0", "System", "invalid.wav")
  Else
   If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
   If temp1S(lineNo%) = digits$ Then
    res% = WritePrivateProfileString("General", "Password", digits$, Profile$)
    OleLog = "new password saved, playing 'password saved' file"
    Call vcfPlayFile("MENU0", "System", "pwsaved.wav")
   Else
    OleLog = "can not verify new pasword, playing 'invalid' file"
    Call vcfPlayFile("MENU0", "System", "invalid.wav")
   End If
```

```
  End If

'*************************************************************************
' CASE "6" ---- NEW PAGER NUMBER
'*************************************************************************

 Case "NEWPGR0"
  OleLog = "playing 'enter new pager number' file"
  Call vcfPlayFile("NEWPGR1", "System", "enterpgr.wav")

 Case "NEWPGR1"
  OleLog = "getting pager number DTMF"
  Call vcfGetDigits("NEWPGR2", 15, 3)

 Case "NEWPGR2"
  OleLog = "digits " & digits$ & " received, "
  If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
  If digits$ <> "" Then
   OleLog = OleLog & "playing file 'you have entered...'"
   tmpInput$(lineNo%) = digits$
   Call vcfPlayFile("NEWPGR3", "System", "youenter.wav")
  Else
   OleLog = OleLog & "playing file 'invalid input...'"
   counter(lineNo%) = counter(lineNo%) + 1
   If counter(lineNo%) = 2 Then
    Call vcfPlayFile("MENU0", "System", "invalid.wav")
   Else
    Call vcfPlayFile("NEWPGR0", "System", "invalid.wav")
   End If
  End If

 Case "NEWPGR3"
  OleLog = "playing pager number " & tmpInput$(lineNo%) & " for verification"
```

```
  Call ssfPlayString("NEWPGR4", tmpInput$(lineNo%))

 Case "NEWPGR4"
  OleLog = "play file '1 if correct, 2 to enter again...'"
  Call vcfPlayFile("NEWPGR5", "System", "correct.wav")

 Case "NEWPGR5"
  OleLog = "getting response..."
  Call vcfGetDigits("NEWPGR6", 1, 3)

 Case "NEWPGR6"
  If digits$ = "1" Then
   res% = WritePrivateProfileString("Paging", "PhoneNumber", tmpInput$(lineNo%), Profile$)
   OleLog = "pager number correct"
   Call vcfContinue("MENU0")
  Else
   OleLog = "get pager number again..."
   Call vcfContinue("NEWPGR0")
  End If

'*************************************************************************
' CASE "7" ---- NEW MESSAGE NOTIFICATION NUMBER
'*************************************************************************

 Case "NEWMNN0"
  OleLog = "playing 'enter new message notification number' file"
  Call vcfPlayFile("NEWMNN1", "System", "entermnn.wav")

 Case "NEWMNN1"
  OleLog = "getting message notification number DTMF"
  Call vcfGetDigits("NEWMNN2", 15, 3)

 Case "NEWMNN2"
```

```
 OleLog = "digits " & digits$ & " received, "
 If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
 If digits$ <> "" Then
  OleLog = OleLog & "playing file 'you have entered...'"
  tmpInput$(lineNo%) = digits$
  Call vcfPlayFile("NEWMNN3", "System", "youenter.wav")
 Else
  OleLog = OleLog & "playing file 'invalid input...'"
  counter(lineNo%) = counter(lineNo%) + 1
  If counter(lineNo%) = 2 Then
   Call vcfPlayFile("MENU0", "System", "invalid.wav")
  Else
   Call vcfPlayFile("NEWMNN0", "System", "invalid.wav")
  End If
 End If

Case "NEWMNN3"
 OleLog = "playing message notification number " & tmpInput$(lineNo%) & " for verification"
 Call ssfPlayString("NEWMNN4", tmpInput$(lineNo%))

Case "NEWMNN4"
 OleLog = "play file '1 if correct, 2 to enter again...'"
 Call vcfPlayFile("NEWMNN5", "System", "correct.wav")

Case "NEWMNN5"
 OleLog = "getting response..."
 Call vcfGetDigits("NEWMNN6", 1, 3)

Case "NEWMNN6"
 If digits$ = "1" Then
  res% = WritePrivateProfileString("Notification", "PhoneNumber", tmpInput$(lineNo%), Profile$)
  OleLog = "message notification number correct"
  Call vcfContinue("MENU0")
```

```
  Else
    OleLog = "get pager number again..."
    Call vcfContinue("NEWMNN0")
  End If



'*****************************************************************************
' CASE "1" ---- PLAY MESSAGES
'*****************************************************************************

 Case "PLAY_MESSAGE_INSTRUCTIONS"
  OleLog = "playing 'press * at any time...' file"
  If temp1$(lineNo%) = "Old" Then
    Call vcfPlayFile("PLAY_#OF_OLD_MSG1", "System", "pbinstru.wav")
  Else
    Call vcfPlayFile("GET_INSTRUCT_DIGITS", "System", "pbinstru.wav")
  End If

 Case "PLAY_#OF_OLD_MSG1"
  counter%(lineNo%) = 0
  s$ = Dir(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\*.i__")
  While s$ <> ""
    counter%(lineNo%) = counter%(lineNo%) + 1
    s$ = Dir
  Wend
  If counter%(lineNo%) = 0 Then
    OleLog = "playing 'no messages...' file"
    Call vcfPlayFile("MENU0", "System", "nomsg.wav")
  Else
    OleLog = "playing 'you have...' file"
    Call vcfPlayFile("PLAY_#OF_OLD_MSG2", "System", "youhave.wav")
  End If
```

```
Case "PLAY_#OF_OLD_MSG2"
 OleLog = "playing number of messages (" & counter(lineNo%) & ")"
 Call ssfPlayNumber("PLAY_#OF_OLD_MSG3", CLng(counter(lineNo%)))
Case "PLAY_#OF_OLD_MSG3"
 If counter(lineNo%) = 1 Then
  OleLog = "playing 'message...' file"
  Call vcfPlayFile("GET_INSTRUCT_DIGITS", "System", "message.wav")
 Else
  OleLog = "playing 'messages...' file"
  Call vcfPlayFile("GET_INSTRUCT_DIGITS", "System", "messages.wav")
 End If

Case "GET_INSTRUCT_DIGITS"
 OleLog = "get DTMF for  instructions"
 Call vcfGetDigits("PROCESS_INSTRUCT_DIGITS", 1, 1)
Case "PROCESS_INSTRUCT_DIGITS"
 If digits$ = "*" Then
  OleLog = "playing message playback menu file"
  If temp1$(lineNo%) = "Old" Then
   Call vcfPlayFile("PLAY_#OF_OLD_MSG1", "System", "pbmenu.wav")
  Else
   Call vcfPlayFile("FIND_FIRST_MSG", "System", "pbmenu.wav")
  End If
 ElseIf digits$ = "" Then
  Call Jump("FIND_FIRST_MSG")
 Else
  If temp1$(lineNo%) = "Old" Then
   Call Jump("PLAY_#OF_OLD_MSG1")
  Else
   Call Jump("FIND_FIRST_MSG")
  End If
 End If
```

```
Case "FIND_FIRST_MSG"
 temp2$(lineNo%) = ""
 If temp1$(lineNo%) = "Old" Then
  s$ = Dir(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\*.i__")
 Else
  s$ = Dir(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\*.in*")
 End If
 While s$ <> ""
  If s$ > temp2$(lineNo%) Then temp2$(lineNo%) = s$
  s$ = Dir
 Wend
 If temp2$(lineNo%) <> "" Then
  counter(lineNo%) = 1
  Call Jump("PLAY_MSG_COUNT1")
 Else
  OleLog = "playing 'no messages...' file"
  Call vcfPlayFile("MENU0", "System", "nomsg.wav")
 End If

Case "PLAY_MSG_COUNT1"
 OleLog = "playing 'message...' file"
 Call vcfPlayFile("PLAY_MSG_COUNT2", "System", "message.wav")
Case "PLAY_MSG_COUNT2"
 OleLog = "playing message number (" & counter%(lineNo%) & ")"
 Call ssfPlayNumber("PLAY_DATE1", CLng(counter%(lineNo%)))

Case "PLAY_DATE1"
 i% = DateDiff("d", Left(temp2$(lineNo%), 2) & "/" & Mid(temp2$(lineNo%), 3, 2), Now)
 If i% < 0 Then
  i% = DateDiff("d", DateAdd("yyyy", -1, Left(temp2$(lineNo%), 2) & "/" & Mid(temp2$(lineNo%), 3,
2)), Now)
 End If
 If i% = 0 Then
```

```
    OleLog = "playing 'today...' file"
    Call vcfPlayFile("PLAY_MSG", "System", "today.wav")
   ElseIf i% = 1 Then
    OleLog = "playing 'yesterday...' file"
    Call vcfPlayFile("PLAY_MSG", "System", "yestrday.wav")
   Else
    OleLog = "playing 'received...' file"
    Call vcfPlayFile("PLAY_DATE2", "System", "received.wav")
   End If
  Case "PLAY_DATE2"
   i% = DateDiff("d", Left(temp2$(lineNo%), 2) & "/" & Mid(temp2$(lineNo%), 3, 2), Now)
   If i% < 0 Then
    i% = DateDiff("d", DateAdd("yyyy", -1, Left(temp2$(lineNo%), 2) & "/" & Mid(temp2$(lineNo%), 3,
2)), Now)
   End If
   OleLog = "playing number of days old (" & i% & ")"
   Call ssfPlayNumber("PLAY_DATE3", CLng(i%))
  Case "PLAY_DATE3"
   OleLog = "playing 'days ago...' file"
   Call vcfPlayFile("PLAY_MSG", "System", "daysago.wav")

  Case "PLAY_MSG"
   MsgPath$ = AppDir & "mboxes\" & temp$(lineNo%) & "\msg\"
   OleLog = "playing message " & MsgPath$ & Left(temp2$(lineNo%), 9) & "wav"
   If Right(temp2$(lineNo%), 1) <> "_" Then
    On Error Resume Next
    Name MsgPath$ & temp2$(lineNo%) As MsgPath$ & Left(temp2$(lineNo%), 9) & "in_"
    If Err.Number = 0 Then
     temp2$(lineNo%) = Left(temp2$(lineNo%), 9) & "in_"
     res% = GetPrivateProfileString("Genera.", "State", "", retstr$, Len(retstr$) - 1, MsgPath$ & "status")
     mState$ = Str(Val(Left(retstr$, res%)) + 1)
     If mState$ > "30000" Then mState$ = "0"
     res% = WritePrivateProfileString("General", "State", mState$, MsgPath$ & "status")
```

```
    End If
    On Error GoTo 0
  End If
  If isfile(MsgPath$ & Left(temp2$(lineNo%), 9) & "wav") Then
    Call vcfPlayFile("GET_MSG_DIGITS", "", MsgPath$ & Left(temp2$(lineNo%), 9) & "wav")
  Else
    Call Jump("PLAY_TIME0")
  End If

 Case "GET_MSG_DIGITS"
  OleLog = "get DTMF for  playback options"
  Call vcfGetDigits("PROCESS_MSG_DIGITS", 1, 1)

 Case "PROCESS_MSG_DIGITS"
  OleLog = "playback command digit " & digits$ & " received, "
  Select Case digits$
  Case "1"   'rewind
    OleLog = OleLog & "rewinding message"
    Call vcfPlayForward("GET_MSG_DIGITS", "", AppDir & "mboxes\" & temp$(lineNo%) & "\msg\"
& Left(temp2$(lineNo%), 9) & "wav", -65536)
  Case "3"   'forward
    OleLog = OleLog & "fast forwarding message"
    Call vcfPlayForward("GET_MSG_DIGITS", "", AppDir & "mboxes\" & temp$(lineNo%) & "\msg\"
& Left(temp2$(lineNo%), 9) & "wav", 65536)
  Case "4"   'previous
    Call Jump("FIND_PREV_MSG")
  Case "6"   'next
    Call Jump("FIND_NEXT_MSG")
  Case "7"   'delete
    MsgPath$ = AppDir & "mboxes\" & temp$(lineNo%) & "\msg\"
    On Error Resume Next
    If isfile(MsgPath$ & Left(temp2$(lineNo%), 9) & "wav") Then Kill MsgPath$ & Left(temp2$
(lineNo%), 9) & "wav"
```

```
  If Err.Number <> 0 Then
    OleLog = OleLog & "playing 'file in use...' file"
    Call vcfPlayFile("FIND_NEXT_MSG", "System", "inuse.wav")
  Else
    Kill MsgPath$ & Left(temp2$(lineNo%), 9) & "i*"
    OleLog = OleLog & "playing 'message deleted...' file"
    Call vcfPlayFile("FIND_NEXT_MSG", "System", "msgdel.wav")
    counter%(lineNo%) = counter%(lineNo%) - 1

    res% = GetPrivateProfileString("General", "State", "", retstr$, Len(retstr$) - 1, MsgPath$ & "status")
    mState$ = Str(Val(Left(retstr$, res%)) + 1)
    If mState$ > "30000" Then mState$ = "0"
    res% = WritePrivateProfileString("General", "State", mState$, MsgPath$ & "status")
  End If
 Case "8"   'play message details
  Call Jump("PLAY_TIME0")
 Case "9"   'forward message
  Call Jump("SEND_MSG0")
 Case "*"   'instructions
  OleLog = OleLog & "playing message playback menu file"
  Call vcfPlayFile("GET_MSG_DIGITS", "System", "pbmenu.wav")
 Case "#"
  Call Jump("MENU0")
 Case Else  'all other message digits
  Call Jump("FIND_NEXT_MSG")
 End Select

Case "SEND_MSG0"
 OleLog = "playing 'please enter mailbox #...' file"
 Call vcfPlayFile("SEND_MSG1", "System", "entrsnmb.wav")
Case "SEND_MSG1"
 OleLog = "getting DTMF for mailbox number"
 Call vcfGetDigits("SEND_MSG2", 9, 3)
```

```
 Case "SEND_MSG2"
  If Right(digits$, 1) = "#" Then digits$ = Left(digits$, Len(digits$) - 1)
  Mailboxes.FindFirst "MailBoxNumber='" & digits$ & "'"
  If Not Mailboxes.NoMatch And digits$ <> temp$(lineNo%) Then
   On Error Resume Next
   MkDir AppDir & "mboxes\" & digits$ & "\msg"
   ATime = Left(temp2$(lineNo%), 2) & "/" & Mid(temp2$(lineNo%), 3, 2) & " " & Mid(temp2$
(lineNo%), 5, 2) & ":" & Mid(temp2$(lineNo%), 7, 2)
   Do
    Filename$ = AppDir & "mboxes\" & digits$ & "\msg\" & Format$(ATime, "mmddhhnn") & ".wav"
    ATime = DateAdd("n", 1, ATime)
   Loop While isfile%(Filename$) Or isfile%(Left(Filename$, Len(Filename$) - 3) & "raw") Or isfile%
(Left(Filename$, Len(Filename$) - 3) & "inf") Or isfile%(Left(Filename$, Len(Filename$) - 3) & "in_")
Or isfile%(Left(Filename$, Len(Filename$) - 3) & "i__")  'Loop until a new file is found

   If isfile(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\" & temp2$(lineNo%)) Then
    FileCopy AppDir & "mboxes\" & temp$(lineNo%) & "\msg\" & temp2$(lineNo%), AppDir &
"mboxes\" & digits$ & "\msg\" & Left(Right(Filename$, 12), 9) & "inf"
    If isfile(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\" & Left(temp2$(lineNo%), 9) & "wav")
Then
     FileCopy AppDir & "mboxes\" & temp$(lineNo%) & "\msg\" & Left(temp2$(lineNo%), 9) &
"wav", Filename$
    End If
    res% = WritePrivateProfileString("General", "ForwardedFrom", temp$(lineNo%), AppDir &
"mboxes\" & digits$ & "\msg\" & Left(Right(Filename$, 12), 9) & "inf")
   End If
   OleLog = "digits " & digits$ & " received, playing 'message sent...' file"
   Call vcfPlayFile("SEND_MSG3", "System", "msgsent.wav")
  Else
   OleLog = "digits " & digits$ & " received, playing 'invalid input...' file"
   Call vcfPlayFile("SEND_MSG3", "System", "invalid.wav")
  End If
 Case "SEND_MSG3"
```

```
    OleLog = "continuing message"
'   Call vcfPlayFile("GET_MSG_DIGITS", "", AppDir & "mboxes\" & temp$(lineNo%) & "\msg\" &
temp2$(lineNo%))
    Call vcfPlayFile("GET_MSG_DIGITS", "", AppDir & "mboxes\" & temp$(lineNo%) & "\msg\" &
Left(temp2$(lineNo%), 9) & "wav")

  Case "PLAY_TIME0"
    OleLog = "playing 'time of message...' file"
    Call vcfPlayFile("PLAY_TIME1", "System", "timemsg.wav")
  Case "PLAY_TIME1"
    OleLog = "playing time of message(" & Mid(temp2$(lineNo%), 5, 2) & ":" & Mid(temp2$(lineNo%),
7, 2) & ")"
    Call ssfPlayTime("PLAY_PHONE_NUMBER0", Mid(temp2$(lineNo%), 5, 4))
  Case "PLAY_PHONE_NUMBER0"
    res% = GetPrivateProfileString("General", "PhoneNumber", "", retstr$, Len(retstr$) - 1, AppDir &
"mboxes\" & temp$(lineNo%) & "\msg\" & temp2$(lineNo%))
    If Left(retstr$, res%) = "" Then
      res% = GetPrivateProfileString("General", "CallerID", "", retstr$, Len(retstr$) - 1, AppDir &
"mboxes\" & temp$(lineNo%) & "\msg\" & temp2$(lineNo%))
    End If
    If Left(retstr$, res%) <> "" And Left(retstr$, 1) <> "N" Then
      OleLog = "playing 'phone number...' file"
      Call vcfPlayFile("PLAY_PHONE_NUMBER1", "System", "phonenum.wav")
    Else
      Call Jump("PLAY_PHONE_NUMBER2")
    End If
  Case "PLAY_PHONE_NUMBER1"
    res% = GetPrivateProfileString("General", "PhoneNumber", "", retstr$, Len(retstr$) - 1, AppDir &
"mboxes\" & temp$(lineNo%) & "\msg\" & temp2$(lineNo%))
    If Left(retstr$, res%) = "" Then
      res% = GetPrivateProfileString("General", "CallerID", "", retstr$, Len(retstr$) - 1, AppDir &
"mboxes\" & temp$(lineNo%) & "\msg\" & temp2$(lineNo%))
    End If
```

```
    OleLog = "playing phone number/CallerID (" & Left(retstr$, res%) & ")"
    Call ssfPlayString("PLAY_PHONE_NUMBER2", Left(retstr$, res%))
  Case "PLAY_PHONE_NUMBER2"
    OleLog = "continuing message"
    Call vcfPlayFile("GET_MSG_DIGITS", "", AppDir & "mboxes\" & temp$(lineNo%) & "\msg\" &
Left(temp2$(lineNo%), 9) & "wav")

  Case "FIND_NEXT_MSG"
    current$ = temp2$(lineNo%)
    temp2$(lineNo%) = ""
    If temp1$(lineNo%) = "Old" Then
      s$ = Dir(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\*.i__")
    Else
      s$ = Dir(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\*.in*")
    End If
    While s$ <> ""
      If s$ > temp2$(lineNo%) And Left(s$, 8) < Left(current$, 8) Then
'       Debug.Print s$, temp2$(lineNo%), current$
        temp2$(lineNo%) = s$
      End If
      s$ = Dir
    Wend
    If temp2$(lineNo%) <> "" Then
      counter%(lineNo%) = counter%(lineNo%) + 1
      Call Jump("PLAY_MSG_COUNT1")
    Else

      If temp1$(lineNo%) <> "Old" Then
        res% = GetPrivateProfileString("Message Lamp", "DisableMessageLamp", "No", retstr$, Len
(retstr$) - 1, AppDir & "mboxes\" & temp(lineNo%) & "\transfer.ini")
        If UCase(Left(retstr$, res%)) = "YES" Then
          Schedule.FindFirst "OwnerMailbox = '" & temp(lineNo%) & "' and Action = 'Lamp' and Status =
'Pending'"
```

```
        If Schedule.NoMatch Then Schedule.AddNew Else Schedule.Edit
        Schedule("TimeScheduled") = DateAdd("s", 10, Now)
        Schedule("TimeInserted") = Now
        Schedule("Status") = "Pending"   'status$
        Schedule("Action") = "Lamp"
        res% = GetPrivateProfileString("Message Lamp", "OffDialString", "", retstr$, Len(retstr$) - 1,
AppDir & "mboxes\" & temp(lineNo%) & "\transfer.ini")
        Schedule("PhoneNumber") = Left(retstr$, res%)
        Schedule("Attempts") = GetPrivateProfileInt("Message Lamp", "Attempts", 1, AppDir &
"mboxes\" & temp(lineNo%) & "\transfer.ini")
        Schedule("Name") = ""
        Schedule("OwnerMailbox") = temp(lineNo%)
        Schedule("AttemptsMade") = 0
        Schedule.Update
       End If
      End If

      OleLog = "playing 'no messages...' file"
      Call vcfPlayFile("GET_NO_MORE_MSG_DIGITS", "System", "nomsg.vav")
      temp2$(lineNo%) = current$
     End If

   Case "GET_NO_MORE_MSG_DIGITS"
     OleLog = "get DTMF for  no more msg"
     Call vcfGetDigits("PROCESS_NO_MORE_MSG_DIGITS", 1, 2)
   Case "PROCESS_NO_MORE_MSG_DIGITS"
     If digits$ = "4" Then
      Call Jump("PLAY_MSG_COUNT1")
     Else
'     Call Jump("PLAY_MESSAGE_INSTRUCTIONS")
      Jump ("MENU0")
     End If
```

```
Case "FIND_PREV_MSG"
 current$ = temp2$(lineNo%)
 temp2$(lineNo%) = "99999999.INF"
 If temp1$(lineNo%) = "Old" Then
  s$ = Dir(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\*.i__")
 Else
  s$ = Dir(AppDir & "mboxes\" & temp$(lineNo%) & "\msg\*.in*")
 End If
 While s$ <> ""
  If s$ < temp2$(lineNo%) And Left(s$, 8) > Left(current$, 8) Then temp2$(lineNo%) = s$
  s$ = Dir
 Wend
 If temp2$(lineNo%) <> "99999999.INF" Then
  counter%(lineNo%) = counter%(lineNo%) - 1
  Call Jump("PLAY_MSG_COUNT1")
 Else
  OleLog = "playing 'no messages...' file"
  Call vcfPlayFile("MENU0", "System", "nomsg.wav")
 End If

Case "EXIT"
 OleLog = "finished remote function,"
 Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
 If Not Tonetables.NoMatch Then
  Mailbox$ = Tonetables("Assignment")
  OleLog = OleLog & " going to " & Mailbox$
  Call vcfGoto(Mailbox$)
 Else
  OleLog = OleLog & " going to Hangup"
  Call vcfGoto("Hangup")
 End If

 Case Else
```

```
    OleLog = "invalid state, going to Hangup"
    Call vcfGoto("Hangup")

  End Select

End Sub

Private Sub CopySchedule()

  SchedLog("ScheduleID") = Schedule("ScheduleID")
  SchedLog("OwnerMailbox") = Schedule("OwnerMailbox")
  SchedLog("TimeInserted") = Schedule("TimeInserted")
  SchedLog("TimeScheduled") = Schedule("TimeScheduled")
  SchedLog("Action") = Schedule("Action")
  SchedLog("PhoneNumber") = Schedule("PhoneNumber")
  SchedLog("GotoMailbox") = Schedule("GotoMailbox")
  SchedLog("Attempts") = Schedule("Attempts")
  SchedLog("PagerString") = Schedule("PagerString")
  SchedLog("Prompt") = Schedule("Prompt")
  SchedLog("GetConfirmation") = Schedule("GetConfirmation")
  SchedLog("GetReply") = Schedule("GetReply")
  SchedLog("Name") = Schedule("Name")
  SchedLog("Extension") = Schedule("Extension")
  SchedLog("FaxFrom") = Schedule("FaxFrom")
  SchedLog("Status") = Schedule("Status")
  SchedLog("FaxFile") = Schedule("FaxFile")
  SchedLog("AttemptsMade") = Schedule("AttemptsMade")

End Sub

#If ext Then
'##########################################################################
```

```
Private Sub extScheduleAlarm()

 Dim FirstBox$, LastBox$, Count&
 Dim lineNo&, Mailbox$, digits$
 Dim iday&, ihour&

 lineNo& = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)

 Static sDate$(1 To Maxline)
 Static counter&(1 To Maxline)

' If TERMDEV& <> 0 Then
'   OleLog$ = "terminating condition, hanging up"
'   Call vcfHangup
'   Exit Sub
' End If

  If State$(lineNo&) = "" Then
    State$(lineNo&) = "Start"
  End If

  Select Case State$(lineNo&)

  Case "Start"
    counter&(lineNo&) = 0
    OleLog$ = "setting counter to 0"
    Call Jump("RequestDateOption")

'------DateOption
  Case "RequestDateOption"
    OleLog$ = "playing 'press 1 to enter date, or press 2 for today...' file"
```

```
  Call vcfPlayFile("GetDateOption", "Current", "prompt1.wav")

Case "GetDateOption"
  OleLog$ = "getting DTMF input"
  Call vcfGetDigits("ProcessDateOption", 1, 3)

Case "ProcessDateOption"
  OleLog$ = "digit '" & digits$ & "' received, "

  Select Case digits$
  Case "1"
    OleLog$ = OleLog & "get date"
    Call vcfContinue("ResetMonthCounter")
  Case "2"
    sDate$(lineNo&) = Format(Now, "mmdd")
    OleLog$ = OleLog$ & "set for today"
    Call vcfContinue("ResetHourCounter")
  Case "3"
    sDate$(lineNo&) = Format(DateAdd(Now, 1, "d"), "mmdd")
    OleLog$ = OleLog$ & "set for tomorrow"
    Call vcfContinue("ResetHourCounter")
  Case "4"
    sDate$(lineNo&) = Format(DateAdd(Now, 2, "d"), "mmdd")
    OleLog$ = OleLog$ & "set for day after tomorrow"
    Call vcfContinue("ResetHourCounter")
  Case Else
    counter&(lineNo&) = counter&(lineNo&) + 1
    If counter&(lineNo&) >= 2 Then
      OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, going to End"
      Call vcfContinue("End")
    Else
      OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, playing invalid prompt"
      Call vcfPlayFile("Start", "RequestDateOption", "invalid.wav")
```

```
    End If
   End Select

'------Month
  Case "ResetMonthCounter"
   OleLog$ = "setting counter to 0"
   counter&(lineNo&) = 0
   Call Jump("RequestMonth")

  Case "RequestMonth"
   OleLog$ = "playing 'enter month...' file"
   Call vcfPlayFile("GetMonth", "Current", "prompt2.wav")

  Case "GetMonth"
   OleLog$ = "getting DTMF input"
   Call vcfGetDigits("ProcessMonth", 2, 3)

  Case "ProcessMonth"
   OleLog$ = "digits '" & digits$ & "' received, "
   If digits$ = "" Or Val(digits$) = 0 And Val(digits$) > 12 Then
    counter&(lineNo&) = counter&(lineNo&) + 1
    If counter&(lineNo&) >= 2 Then
     OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, going to End"
     Call vcfContinue("End")
    Else
     OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, playing invalid prompt"
     Call vcfPlayFile("Start", "RequestMonth", "invalid.wav")
    End If
   Else
    sDate$(lineNo&) = Format(Val(digits$), "00")
    OleLog$ = OleLog$ & "setting month to " & sDate$(lineNo&)
    Call vcfContinue("ResetDayCounter")
   End If
```

```
'------Day
 Case "ResetDayCounter"
  OleLog$ = "setting counter to 0"
  counter&(lineNo&) = 0
  Call Jump("RequestDay")

 Case "RequestDay"
  OleLog$ = "playing 'enter day...' file"
  Call vcfPlayFile("GetDay", "Current", "prompt3.wav")

 Case "GetDay"
  OleLog$ = "getting DTMF input"
  Call vcfGetDigits("ProcessDay", 2, 3)

 Case "ProcessDay"
  OleLog$ = "digits '" & digits$ & "' received, "

  If digits$ <> "" Then
   iday& = Val(digits$)
   If sDate$(lineNo&) = "02" Then
    If Val(Format(Now, "yyyy")) Mod 4 = 0 Then
     If iday& < 1 Or iday& > 29 Then iday& = 0
    Else
     If iday& < 1 Or iday& > 28 Then iday& = 0
    End If
   ElseIf sDate$(lineNo&) = "04" Or sDate$(lineNo&) = "06" Or sDate$(lineNo&) = "09" Or sDate$
(lineNo&) = "11" Then
    If iday& < 1 Or iday& > 30 Then iday& = 0
   Else
    If iday& < 1 Or iday& > 31 Then iday& = 0
   End If
  End If
```

```
  If digits$ = "" Or iday& = 0 Then
    counter&(lineNo&) = counter&(lineNo&) + 1
    If counter&(lineNo&) >= 2 Then
      OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, going to End"
      Call vcfContinue("End")
    Else
      OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, playing invalid prompt"
      Call vcfPlayFile("Start", "RequestDay", "invalid.wav")
    End If
  Else
    sDate$(lineNo&) = sDate$(lineNo&) & Format(iday&, "00")
    OleLog$ = OleLog$ & "setting month/day to " & sDate$(lineNo&)
    Call vcfContinue("ResetHourCounter")
  End If

'------Hour
  Case "ResetHourCounter"
    OleLog$ = "setting counter to 0"
    counter&(lineNo&) = 0
    Call Jump("RequestHour")

  Case "RequestHour"
    OleLog$ = "playing 'enter hour...' file"
    Call vcfPlayFile("GetHour", "Current", "prompt4.wav")

  Case "GetHour"
    OleLog$ = "getting DTMF input"
    Call vcfGetDigits("ProcessHour", 2, 3)

  Case "ProcessHour"
    OleLog$ = "digits '" & digits$ & "' received, "

    If digits$ <> "" Then
```

```
    ihour& = Val(digits$)
    If ihour& < 0 Or ihour& > 23 Then ihour& = -1
  End If

  If digits$ = "" Or ihour& = -1 Then
    counter&(lineNo&) = counter&(lineNo&) + 1
    If counter&(lineNo&) >= 2 Then
      OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, going to End"
      Call vcfContinue("End")
    Else
      OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, playing invalid prompt"
      Call vcfPlayFile("Start", "RequestHour", "invalid.wav")
    End If
  Else
    sDate$(lineNo&) = sDate$(lineNo&) & Format(ihour&, "00")
    OleLog$ = OleLog$ & "setting month/day/hour to " & sDate$(lineNo&)
    If ihour& = 0 Or ihour& > 12 Then
      Call vcfContinue("ResetMinuteCounter")
    Else
      Call vcfContinue("ResetAMPMCounter")
    End If
  End If

'------AMPM
  Case "ResetAMPMCounter"
    OleLog$ = "setting counter to 0"
    counter&(lineNo&) = 0
    Call Jump("RequestAMPM")

  Case "RequestAMPM"
    OleLog$ = "playing 'enter am or pm...' file"
    Call vcfPlayFile("GetAMPM", "Current", "prompt5.wav")
```

```
Case "GetAMPM"
 OleLog$ = "getting DTMF input"
 Call vcfGetDigits("ProcessAMPM", 1, 3)

Case "ProcessAMPM"
 OleLog$ = "digits '" & digits$ & "' received, "

 Select Case digits$
 Case "1"
  If Right(sDate$(lineNo&), 2) = "12" Then sDate$(lineNo&) = Left(sDate$(lineNo&), 4) & "00"
  OleLog$ = OleLog$ & "setting military month/day/hour to " & sDate$(lineNo&)
  Call vcfContinue("ResetMinuteCounter")
 Case "2"
  If Right(sDate$(lineNo&), 2) <> "12" Then sDate$(lineNo&) = Left(sDate$(lineNo&), 4) & Format
(Val(Right(sDate$(lineNo&), 2)) + 12, "00")
  OleLog$ = OleLog$ & "setting military month/day/hour to " & sDate$(lineNo&)
  Call vcfContinue("ResetMinuteCounter")
 Case Else
  counter&(lineNo&) = counter&(lineNo&) + 1
  If counter&(lineNo&) >= 2 Then
   OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, going to End"
   Call vcfContinue("End")
  Else
   OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, playing invalid prompt"
   Call vcfPlayFile("Start", "RequestAMPM", "invalid.wav")
  End If
 End Select

'------Minute
 Case "ResetMinuteCounter"
  OleLog$ = "setting counter to 0"
  counter&(lineNo&) = 0
  Call Jump("RequestMinute")
```

```
Case "RequestMinute"
  OleLog$ = "playing 'enter minute...' file"
  Call vcfPlayFile("GetMinute", "Current", "prompt6.wav")

 Case "GetMinute"
  OleLog$ = "getting DTMF input"
  Call vcfGetDigits("ProcessMinute", 2, 3)

 Case "ProcessMinute"
  OleLog$ = "digits '" & digits$ & "' received, "
  If digits$ = "" Or Val(digits$) < 0 And Val(digits$) > 59 Then
   counter&(lineNo&) = counter&(lineNo&) + 1
   If counter&(lineNo&) >= 2 Then
    OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, going to End"
    Call vcfContinue("End")
   Else
    OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, playing invalid prompt"
    Call vcfPlayFile("Start", "RequestMinute", "invalid.wav")
   End If
  Else
   sDate$(lineNo&) = sDate$(lineNo&) & Format(Val(digits$), "00")
   OleLog$ = OleLog$ & "setting military month/day/hour/minute to " & sDate$(lineNo&)
   Call vcfContinue("PrePlayDate")
  End If

 Case "PrePlayDate"
 Case "PlayDate"
 Case "PlayTime"

'------TimeCorrectOption
 Case "ResetTimeCorrectOptionCounter"
  OleLog$ = "setting counter to 0"
  counter&(lineNo&) = 0
```

```
  Call Jump("RequestTimeCorrectOption")

 Case "RequestTimeCorrectOption"
  OleLog$ = "playing 'enter am or pm...' file"
  Call vcfPlayFile("GetAMPM", "Current", "prompt5.wav")

 Case "GetTimeCorrectOption"
  OleLog$ = "getting DTMF input"
  Call vcfGetDigits("ProcessTimeCorrectOption", 1, 3)

 Case "ProcessTimeCorrectOption"
  OleLog$ = "digits '" & digits$ & "' received, "

  Select Case digits$
  Case "1"
   If Right(sDate$(lineNo&), 2) = "12" Then sDate$(lineNo&) = Left(sDate$(lineNo&), 4) & "00"
   OleLog$ = OleLog$ & "setting military month/day/hour to " & sDate$(lineNo&)
   Call vcfContinue("ResetNumberCounter")
  Case "2"
   If Right(sDate$(lineNo&), 2) <> "12" Then sDate$(lineNo&) = Left(sDate$(lineNo&), 4) & Format
(Val(Right(sDate$(lineNo&), 2)) + 12, "00")
   OleLog$ = OleLog$ & "setting military month/day/hour to " & sDate$(lineNo&)
   Call vcfContinue("ResetNumberCounter")
  Case Else
   counter&(lineNo&) = counter&(lineNo&) - 1
   If counter&(lineNo&) >= 2 Then
    OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, going to End"
    Call vcfContinue("End")
   Else
    OleLog$ = OleLog$ & "invalid attempt #" & counter&(lineNo&) & " of 2, playing invalid prompt"
    Call vcfPlayFile("Start", "RequestTimeCorrectOption", "invalid.wav")
   End If
  End Select
```

```
Case "End"
   OleLog = "finished function, "
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & "going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & "going to Hangup"
    Call vcfGoto("Hangup")
   End If

 Case Else
   OleLog$ = "invalid state, going to Hangup"
   Call vcfGoto("Hangup")

 End Select

' Exit Sub

'ErrorHandler:
' OleLog$ = "Error #" & Str(Err.Number) & " was generated by " & Err.Source & ":" & Err.Description
& " Hanging up."
' Call vcfHangup

End Sub

Private Sub extGoBySequence()

  Dim FirstBox$, LastBox$, Count&
  Dim lineNo%, Mailbox$

  lineNo% = OleData(0)
```

```
Mailbox$ = OleData(1)

Static CurrentBox$

' If TERMDEV& <> 0 Then
'   OleLog$ = "terminating condition, hanging up"
'   Call vcfHangup
'   Exit Sub
' End If

If State$(lineNo%) = "" Then
  State$(lineNo%) = "Start"
End If

Select Case State$(lineNo%)

Case "Start"

  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"

  FirstBox$ = Mailboxes("FirstBox")
  LastBox$ = Mailboxes("LastBox")

  If CurrentBox$ < FirstBox$ Then CurrentBox$ = Mailboxes("StartBox")

  If CurrentBox$ > LastBox$ Then CurrentBox$ = FirstBox$

  Mailboxes.FindFirst "MailBoxNumber='" & CurrentBox$ & "'"
  Count& = 0
  While Mailboxes.NoMatch And Count& < 2
    CurrentBox$ = Format(Val(CurrentBox$) + 1, "")
    If CurrentBox$ > LastBox$ Then
      CurrentBox$ = FirstBox$
```

```
    Count& = Count& + 1
   End If
   Mailboxes.FindFirst "MailBoxNumber='" & CurrentBox$ & "'"
  Wend

  If Count < 2 Then
   OleLog$ = "next mailbox in sequence found, going to " & CurrentBox$
   Call vcfGoto(CurrentBox$)
   CurrentBox$ = Format(Val(CurrentBox$) + 1, "")
  Else
   OleLog$ = "no mailbox found in sequence, going to Hangup"
   Call vcfGoto("Hangup")
  End If

 Case Else
  OleLog$ = "invalid state, going to Hangup"
  Call vcfGoto("Hangup")

 End Select

' Exit Sub

'ErrorHandler:
' OleLog$ = "Error #" & Str(Err.Number) & " was generated by " & Err.Source & ":" & Err.Description
& " Hanging up."
' Call vcfHangup

End Sub

Private Sub extPlayRanPrompt()

  Dim lineNo%, Mailbox$, digits$, nextstate$
  Dim dialstring$, MaxDigits%
```

```
Dim MaxPrompts%

lineNo% = OleData(0)
Mailbox$ = OleData(1)
digits$ = OleData(2)

If State$(lineNo%) = "" Then
  counter%(lineNo%) = 0
  State$(lineNo) = "0"
End If

Select Case State$(lineNo%)

Case "0"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"   'I forgot this for Guideposts
  MaxPrompts% = 3
  If Not IsNull(Mailboxes("MaxCount")) Then MaxPrompts% = Mailboxes("MaxCount")

  OleLog = "playing random prompt file"
  Call vcfPlayFile("10", "Current", Int((MaxPrompts% * Rnd(-1 * Val(Format(Now, "mmddyy")))) + 1)
& ".wav")

Case "10"
  OleLog = "finished prompt,"
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
  Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
  End If
```

```
End Select

End Sub

Private Sub extPlayClassifiedAd()

 Dim lineNo%, Mailbox$, digits$, nextstate$, callstate%, CallerID$
 Dim dialstring$, MaxDigits%
 Dim res%, Profile$, retstr As String * 255, s$
 Dim mState$, i%



 lineNo% = OleData(0)
 Mailbox$ = OleData(1)
 digits$ = OleData(2)
 callstate% = OleData(3)
 CallerID$ = OleData(4)

 Profile$ = AppDir & "mboxes\" & Mailbox$ & "\transfer.ini"

 If State$(lineNo%) = "" Then
  State$(lineNo) = "Start"
  On Error Resume Next
  If Not isfile(Profile$) Then FileCopy App.Path & "\transfer.ini", Profile$
  On Error GoTo 0
 End If

 Select Case State$(lineNo%)

 Case "Start"
  counter%(lineNo%) = 0

  If isfile(AppDir & "mboxes\" & Mailbox$ & "\personal.raw") Then
```

```
    For i% = 1 To VoiceLines%
     If FilePlaying$(i) = AppDir & "mboxes\" & Mailbox$ & "\personal.wav" Or FilePlaying$(i) =
AppDir & "mboxes\" & Mailbox$ & "\personal.raw" Then Exit For
    Next i
    If i > VoiceLines% Then
     Call CvtWaveFile(AppDir & "mboxes\" & Mailbox$ & "\personal.raw", AppDir & "mboxes\" &
Mailbox$ & "\personal.wav")
      OleLog = "converting and "
    End If
   End If
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\personal.wav") Then
    OleLog = OleLog & "playing personal greeting"
    Call vcfPlayFile("PlayOptions", "Current", "personal.wav")
   Else
    Call Jump("PlayOptions")
   End If



  Case "PlayOptions"
   OleLog = "playing options file"
   Call vcfPlayFile("GetDTMF", "", AppDir & "mboxes\99\prompt1.wav")

  Case "GetDTMF"
   OleLog = "getting DTMF input"
   Call vcfGetDigits("ProcessInput", 1, 3)

  Case "ProcessInput"

   OleLog = "digit " & digits$ & " received, "

   Select Case digits
   Case "1"
```

```
    OleLog = OleLog & "replay prompt"
    vcfContinue ("Start")
  Case "2"
    OleLog = OleLog & "record message"
    vcfContinue ("GetTakeMessage")
  Case "3"
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='Next'"
    If Not Tonetables.NoMatch Then
      Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
      OleLog = OleLog & "going to next mailbox " & Tonetables("Assignment") & " - " & Mailboxes
("Name")
      Call vcfGoto(Tonetables("Assignment"))
    Else
      OleLog = OleLog & "next mailbox does not exist, playing no more ads"
      Call vcfPlayFile("PlayOptions", "", AppDir & "mboxes\99\prompt2.wav")
    End If
  Case "4"
    OleLog = OleLog & "going to previous mailbox"
    Call vcfGoto("Previous")
  Case "0"
    OleLog = OleLog & "going to exit to Mailbox 100"
    Call vcfGoto("100")
  Case Else
    counter(lineNo%) = counter(lineNo%) + 1
    If counter(lineNo%) >= 2 Then
      OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of 2  going to exit to Mailbox 100"
      Call vcfGoto("100")
    Else
      OleLog = OleLog & "invalid attempt #" & counter(lineNo%) & " of 2, playing invalid prompt"
      Call vcfPlayFile("PlayOptions", "System", "invalid.wav")
    End If
  End Select
```

```
'********************** Recording Message
 Case "GetTakeMessage"
  OleLog = "play 'please record...' file"
  Call vcfPlayFile("FlushBeforeRecording", "System", "recgrt.wav")

 Case "FlushBeforeRecording"
  OleLog = "flushing digits..."
  Call vcfFlushDigits("RecordMessage")

 Case "RecordMessage"
  OleLog = "recording message"   'used by status log
  OleCallLog = "(" & Mailbox$ & ") Message recorded. "
  res% = GetPrivateProfileInt("Message", "MaxMessageLength", 90, Profile$)
  RecFileName$(lineNo%) = GetRecFileName$(Mailbox$)
  Call vcfRecord("FlushAfterRecording", "", RecFileName$(lineNo%), CLng(res%))

 Case "FlushAfterRecording"
  OleLog = "flushing digits..."
  Call vcfFlushDigits("PlayRecordOptions")

'---------------------------- Record Options
 Case "PlayRecordOptions"
  OleLog = "1 to play, 2 record again, 3 to delete..."
  Call vcfPlayFile("GetDTMFRecordOptions", "System", "recmenu.wav")

 Case "GetDTMFRecordOptions"
  OleLog = "getting digits..."
  Call vcfGetDigits("CheckDTMFRecordOptions", 1, 3)

 Case "CheckDTMFRecordOptions"
  OleLog = "digits '" & digits$ & "' received..."
  Select Case digits$
  Case "1"
```

```
    OleLog = OleLog & "play recorded message"
    Call vcfPlayFile("PlayRecordOptions", "", RecFileName(lineNo%))
   Case "2"
    On Error Resume Next
    If isfile(RecFileName(lineNo%)) Then Kill RecFileName(lineNo%)
    Call Jump("PlayPleaseRecordAgain")
   Case "3"
    OleLog = OleLog & "play 'message deleted...' file"
    On Error Resume Next
    If isfile(RecFileName(lineNo%)) Then Kill RecFileName(lineNo%)
    Call vcfPlayFile("End", "System", "recdel.wav")
   Case "4"
    OleLog = OleLog & "play 'message saved' file"
    Call vcfPlayFile("End", "System", "recsave.wav")
   Case Else
    OleLog = OleLog & "play 'message saved' file"
    Call vcfPlayFile("End", "System", "recsave.wav")
   End Select

  Case "PlayPleaseRecordAgain"
   OleLog = "play 'please record...' file"
   Call vcfPlayFile("RecordMessage", "System", "recgrt.wav")

'********************** End
  Case "End"
   If RecFileName$(lineNo%) <> "" Then

    If isfile(RecFileName(lineNo%)) Then
     OleCallLog = "Length " & FileLen(RecFileName$(lineNo%)) \ 8000 & " sec."
     If FileLen(RecFileName$(lineNo%)) < GetPrivateProfileInt("General", "MinMessageLength", 4,
App.Path & "\konnects.ini") * 8000& Then
      On Error Resume Next
      Kill RecFileName(lineNo%)
```

```
    Else
    Call CvtWaveFile(RecFileName$(lineNo%), Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "wav")
    If Not isfile(Left(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf") Then
     res% = WritePrivateProfileString("General", "PhoneNumber", "", Left(RecFileName$(lineNo%),
Len(RecFileName$(lineNo%)) - 3) & "inf")
     res% = WritePrivateProfileString("General", "CallerID", CallerID$, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf")
     res% = GetPrivateProfileString("General", "State", "", retstr$, Len(retstr$) - 1, Left(RecFileName$
(lineNo%), Len(RecFileName$(lineNo%)) - 12) & "status")
     mState$ = Str(Val(Left(retstr$, res%)) + 1)
     If mState$ > "30000" Then mState$ = "0"
     res% = WritePrivateProfileString("General", "State", mState$, Left(RecFileName$(lineNo%), Len
(RecFileName$(lineNo%)) - 12) & "status")
    End If

    res% = GetPrivateProfileString("Paging", "EnablePaging", "No", retstr$, Len(retstr$) - 1, Profile$)
    If UCase(Left(retstr$, res%)) = "YES" Then
     If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then
      Schedule.AddNew
      Schedule("TimeScheduled") = DateAdd("s", 10, Now)
      Schedule("TimeInserted") = Now
      Schedule("Status") = "Pending"  'status$
      Schedule("Action") = "Page"
      res% = GetPrivateProfileString("Paging", "PhoneNumber", "", retstr$, Len(retstr$) - 1, Profile$)
      Schedule("PhoneNumber") = Left(retstr$, res%)
      res% = GetPrivateProfileString("Paging", "PagerString", "", retstr$, Len(retstr$) - 1, Profile$)
      s$ = Left(retstr$, res%)
      res% = GetPrivateProfileString("General", "PhoneNumber", "", retstr$, Len(retstr$) - 1, Left
(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf")
      Schedule("PagerString") = s$ & "," & Left(retstr$, res%) & "#"
      Schedule("Attempts") = 3
```

```
            Schedule("Name") = ""
            Schedule("OwnerMailbox") = Mailbox$
            Schedule("AttemptsMade") = 0
            Schedule.Update
          End If
        End If

        res% = GetPrivateProfileString("Notification", "EnableNotification", "No", retstr$, Len(retstr$) - 1,
Profile$)
        If UCase(Left(retstr$, res%)) = "YES" Then
          If isfile(RecFileName(lineNo%)) Or isfile(Left(RecFileName$(lineNo%), Len(RecFileName$
(lineNo%)) - 3) & "inf") Then
            Schedule.AddNew
            Schedule("TimeScheduled") = DateAdd("n", 5, Now)
            Schedule("TimeInserted") = Now
            Schedule("Status") = "Pending"  'status$
            Schedule("Action") = "Notify"
            res% = GetPrivateProfileString("Notification", "PhoneNumber", "", retstr$, Len(retstr$) - 1,
Profile$)
            Schedule("PhoneNumber") = Left(retstr$, res%)
            Schedule("Attempts") = 2
            Schedule("Name") = ""
            Schedule("OwnerMailbox") = Mailbox$
            Schedule("AttemptsMade") = 0
            Schedule.Update
          End If
        End If
      End If
    End If


    RecFileName$(lineNo%) = ""
  End If
```

```
OleLog = "finished PlayClassfiy function,"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone='Next'"
   If Not Tonetables.NoMatch Then
     Mailboxes.FindFirst "MailBoxNumber='" & Tonetables("Assignment") & "'"
     OleLog = OleLog & "going to next mailbox " & Tonetables("Assignment") & " - " & Mailboxes
("Name")
     Call vcfGoto(Tonetables("Assignment"))
   Else
     OleLog = OleLog & "next mailbox does not exist, playing no more ads"
     Call vcfPlayFile("PlayOptions", "", AppDir & "mboxes\99\prompt2.wav")
   End If

  End Select

End Sub

Private Sub extCreateClassifiedAd()

  Dim lineNo%, Mailbox$, digits$, nextstate$
  Dim dialstring$, MaxDigits%
  Dim res%, retstr As String * 255
  Dim PWValue$
  Dim mState$, i%

  Static MBValue$(1 To 74)

  lineNo% = OleData(0)
  Mailbox$ = OleData(1)
  digits$ = OleData(2)

  If State$(lineNo%) = "" Then
    State$(lineNo) = "Start"
  End If
```

```
Select Case State$(lineNo%)

 Case "Start"

  Randomize ' Initialize random-number generator.
  ' Generate random value between 1 and 6.  Int((upperbound - lowerbound + 1) * Rnd + lowerbound)
  MBValue$(lineNo%) = Right((Str(Int(((99999 - 10001 + 1) * Rnd) + 10001))), 5)
  Mailboxes.FindFirst "MailBoxNumber=" & MBValue$(lineNo%) & ""
  While Not Mailboxes.NoMatch
   MBValue$(lineNo%) = Right((Str(Int(((99999 - 10001 + 1) * Rnd) + 10001))), 5)
   Mailboxes.FindFirst "MailBoxNumber=" & MBValue$(lineNo%) & ""
  Wend

  PWValue$ = Right((Str(Int(((99999 - 10001 + 1) * Rnd) + 10001))), 5)

  Mailboxes.AddNew
  Mailboxes("MailboxNumber") = MBValue$(lineNo%)
  Mailboxes("Name") = "New PlayClassifiedAd " & MBValue$(lineNo%)
  Mailboxes("Function") = "PlayClassifiedAd"
  Mailboxes("ExpandTones") = True
  Mailboxes.Update

  On Error Resume Next
  MkDir AppDir & "MBOXES\" & MBValue$(lineNo%)
  If Not isfile(AppDir & "mboxes\" & MBValue$(lineNo%) & "\transfer.ini") Then FileCopy App.Path
& "\transfer.ini", AppDir & "mboxes\" & MBValue$(lineNo%) & "\transfer.ini"
  On Error GoTo 0

  res% = WritePrivateProfileString("General", "Password", PWValue$, AppDir & "mboxes\" &
MBValue$(lineNo%) & "\transfer.ini")

  RecFileName$(lineNo%) = GetRecFileName$(Mailbox$)
  OleLog = "Saving Inputs to " & RecFileName$(lineNo%) & "."
```

```
    res% = WritePrivateProfileString("General", "Input", "Mailbox:" & MBValue$(lineNo%) & ",
Password:" & PWValue$ & ", 0:" & Inp$(lineNo%, 0) & ", 1:" & Inp$(lineNo%, 1) & ", 2:" & Inp$
(lineNo%, 2) & ", 3:" & Inp$(lineNo%, 3) & ", 4:" & Inp$(lineNo%, 4) & " ,5:" & Inp$(lineNo%, 5) & "
,6:" & Inp$(lineNo%, 6) & " ,7:" & Inp$(lineNo%, 7) & " ,8:" & Inp$(lineNo%, 8) & " ,9:" & Inp$
(lineNo%, 9), Left(RecFileName$(lineNo%), Len(RecFileName$(lineNo%)) - 3) & "inf")
    If isfile(App.Path & "\sysvox\temp" & lineNo% & ".raw") Then
      Call CvtWaveFile(App.Path & "\sysvox\temp" & lineNo% & ".raw", Left(RecFileName$(lineNo%),
Len(RecFileName$(lineNo%)) - 3) & "wav")
    End If
    If isfile(RecFileName$(lineNo%)) Then
      On Error Resume Next
      Kill RecFileName$(lineNo%)
    End If
    res% = GetPrivateProfileString("General", "State", "", retstr$, Len(retstr$) - 1, Left(RecFileNameS
(lineNo%), Len(RecFileName$(lineNo%)) - 12) & "status")
    mState$ = Str(Val(Left(retstrS, res%)) + 1)
    If mState$ > "30000" Then mState$ = "0"
    res% = WritePrivateProfileString("General", "State", mState$, Left(RecFileName$(lineNo%), Len
(RecFileName$(lineNo%)) - 12) & "status")
    RecFileName$(lineNo%) = ""

    vcfContinue ("PlayCustomerPrompt")

  Case "PlayCustomerPrompt"
    OleLog = "playing 'Your customer # is...'"
    Call vcfPlayFile("PlayCustomerNumber", "Current", "prompt1.wav")

  Case "PlayCustomerNumber"
    OleLog = "playing customer #"
    Call ssfPlayString("PlayPWPrompt", MBValue$(lineNo%))

  Case "PlayPWPrompt"
    OleLog = "playing 'Your password # is...'"
```

```
  Call vcfPlayFile("PlayPWNumber", "Current", "prompt2.wav")

 Case "PlayPWNumber"
  OleLog = "playing password #"
  res% = GetPrivateProfileString("General", "Password", "", retstr$, Len(retstr$) - 1, AppDir &
"mboxes\" & MBValue$(lineNo%) & "\transfer.ini")
  PWValue$ = Left(retstr$, res%)
  Call ssfPlayString("PlayRepeatOption", PWValue$)
'  Call ssfPlayString("PlayRepeatOption", MBValue$(lineNo%))

 Case "PlayRepeatOption"
  OleLog = "playing repeat option"
  Call vcfPlayFile("GetRepeatOption", "Current", "prompt3.wav")

 Case "GetRepeatOption"
  OleLog = "getting DTMF for repeat option"
  Call vcfGetDigits("ProcessRepeatOption", 1, 3)

 Case "ProcessRepeatOption"
  If digits$ = "1" Then
   Call Jump("PlayCustomerPrompt")
  Else
   Call Jump("PlayRecordPrompt")
  End If

 Case "PlayRecordPrompt"
  OleLog = "playing record personal prompt"
  Call vcfPlayFile("RecordPersonal", "Current", "prompt4.wav")

 Case "RecordPersonal"
  res% = GetPrivateProfileInt("Message", "MaxPersonalLength", 120, AppDir & "mboxes\" &
MBValue$(lineNo%) & "\transfer.ini")
  OleLog = "recording personal greeting"
```

```
  Call vcfRecord("MENU0", "", AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.raw", CLng
(res%))

 Case "MENU0"
  counter%(lineNo%) = 0
  OleLog = "flushing digits"
  Call vcfFlushDigits("MENU1")

 Case "MENU1"
  OleLog = "playing personal greeting options"
  Call vcfPlayFile("MENU2", "Current", "prompt5.wav")

 Case "MENU2"
  OleLog = "getting DTMF for personal greeting options"
  Call vcfGetDigits("MENU3", 1, 3)

 Case "MENU3"
  Select Case digits$
  Case "1"
   If isfile(AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.raw") Then
    For i% = 1 To VoiceLines%
     If FilePlaying$(i) = AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.wav" Or
FilePlaying$(i) = AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.raw" Then Exit For
    Next i
    If i > VoiceLines% Then
     Call CvtWaveFile(AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.raw", AppDir &
"mboxes\" & MBValue$(lineNo%) & "\personal.wav")
     OleLog = "converting and "
    End If
   End If
   OleLog = OleLog & "playing personal greeting"
   Call vcfPlayFile("MENU0", "", AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.wav")
  Case "2"
```

```
    Call Jump("PlayRecordPrompt")
   Case "3"
    Call Jump("Exit")
   Case Else
    OleLog = "digits " & digits$ & " received, playing 'invalid input...' file"
    counter(lineNo%) = counter(lineNo%) + 1
    If counter(lineNo%) = 3 Then nextstate$ = "EXIT" Else nextstate$ = "MENU1"
    Call vcfPlayFile(nextstate$, "System", "invalid.wav")
   End Select

  Case "Exit"
   If isfile(AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.raw") Then
    For i% = 1 To VoiceLines%
     If FilePlaying$(i) = AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.wav" Or
FilePlaying$(i) = AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.raw" Then Exit For
    Next i
    If i > VoiceLines% Then
     Call CvtWaveFile(AppDir & "mboxes\" & MBValue$(lineNo%) & "\personal.raw", AppDir &
"mboxes\" & MBValue$(lineNo%) & "\personal.wav")
     OleLog = "converting and "
    End If
   End If

   OleLog = "finished CreateClassify,"
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = OleLog & " going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
   End If
```

```
End Select

End Sub

Private Sub extDoDateVerify()

 Dim lineNo%, Mailbox$, sDate$, sTime$
 Dim sMonth$, sDay$, sHour$, sMinute$

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)

 If State$(lineNo%) = "" Then
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"

  sDate$ = ""
  If Left(Mailboxes("DateVariable"), 5) = "Input" Then
   sDate$ = Inp$(lineNo%, Right(Mailboxes("DateVariable"), 1))
   If sDate$ <> "1" And sDate$ <> "2" And sDate$ <> "3" Then
    sMonth$ = Left(sDate$, 2)
    sDay$ = Right(sDate$, 2)
    If sMonth$ < "01" Or sMonth$ > "12" Then
     sDate$ = "i"
    ElseIf sMonth$ = "02" Then
     If Val(Format(Now, "yyyy")) Mod 4 = 0 Then
      If sDay$ < "01" Or sDay$ > "29" Then sDate$ = "i"
     Else
```

```
      If sDay$ < "01" Or sDay$ > "28" Then sDate$ = "i"
     End If
    ElseIf sMonth$ = "04" Or sMonth$ = "06" Or sMonth$ = "09" Or sMonth$ = "11" Then
     If sDay$ < "01" Or sDay$ > "30" Then sDate$ = "i"
    Else
     If sDay$ < "01" Or sDay$ > "30" Then sDate$ = "i"
    End If
  End If
End If

sTime$ = ""
If Left(Mailboxes("TimeVariable"), 5) = "Input" Then
  sTime$ = Inp$(lineNo%, Right(Mailboxes("TimeVariable"), 1))
  sHour$ = Left(sTime$, 2)
  sMinute$ = Right(sTime$, 2)
  If sHour$ < "01" Or sHour$ > "12" Then
    sTime$ = "i"
  ElseIf sMinute$ < "00" Or sMinute$ > "59" Then
    sTime$ = "i"
  End If
End If

If sDate$ = "i" Or sTime$ = "i" Or (sDate$ = "" And sTime$ = "") Then
  If isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
    OleLog = "date/time invalid, playing invalid file"
    Call vcfPlayFile("10", "Current", "invalid.wav")
  Else
    OleLog = "date/time invalid"
    Call vcfContinue("10")
  End If
Else
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  If Not Tonetables.NoMatch Then
```

```
    Mailbox$ = Tonetables("Assignment")
    OleLog = "date/time valid, going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = "date/time valid, going to Hangup"
    Call vcfGoto("Hangup")
   End If
  End If

 Case "10"
  OleLog = "date/time invalid,"
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Invalid'"
  If Tonetables.NoMatch Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  End If

  If Not Tonetables.NoMatch Then
   Mailbox$ = Tonetables("Assignment")
   OleLog = OleLog & " going to " & Mailbox$
   Call vcfGoto(Mailbox$)
  Else
   OleLog = OleLog & " going to Hangup"
   Call vcfGoto("Hangup")
  End If

 End Select

End Sub

Private Sub extEvalDateTime()

 Dim lineNo%, Mailbox$, sDate$, sTime$, sAMPM$
 Dim sMonth$, sDay$, sHour$, sMinute$
```

```
Dim dDate As Date


lineNo% = OleData(0)
Mailbox$ = OleData(1)

If State$(lineNo%) = "" Then
  State$(lineNo) = "0"
End If

Select Case State$(lineNo%)

Case "0"
  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"

  sDate$ = ""
  If Left(Mailboxes("DateVariable"), 5) = "Input" Then
    sDate$ = Inp$(lineNo%, Right(Mailboxes("DateVariable"), 1))
    If sDate$ <> "1" And sDate$ <> "2" And sDate$ <> "3" Then
      sMonth$ = Left(sDate$, 2)
      sDay$ = Right(sDate$, 2)
      If sMonth$ < "01" Or sMonth$ > "12" Then
        sDate$ = "i"
      ElseIf sMonth$ = "02" Then
        If Val(Format(Now, "yyyy")) Mod 4 = 0 Then
          If sDay$ < "01" Or sDayS > "29" Then sDate$ = "i"
        Else
          If sDay$ < "01" Or sDay$ > "28" Then sDate$ = "i"
        End If
      ElseIf sMonth$ = "04" Or sMonth$ = "06" Or sMonth$ = "09" Or sMonth$ = "11" Then
        If sDay$ < "01" Or sDay$ > "30" Then sDate$ = "i"
      Else
        If sDay$ < "01" Or sDay$ > "30" Then sDate$ = "i"
```

```
    End If
   End If
  End If

  sTime$ = ""
  If Left(Mailboxes("TimeVariable"), 5) = "Input" Then
   sTime$ = Inp$(lineNo%, Right(Mailboxes("TimeVariable"), 1))
   sHour$ = Left(sTime$, 2)
   sMinute$ = Right(sTime$, 2)
   If sHour$ < "01" Or sHour$ > "12" Then
    sTime$ = "i"
   ElseIf sMinute$ < "00" Or sMinute$ > "59" Then
    sTime$ = "i"
   End If
  End If

  If sDate$ = "i" Or sTime$ = "i" Or (sDate$ = "" And sTime$ = "") Then
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
    OleLog = "date/time invalid, playing invalid file"
    Call vcfPlayFile("10", "Current", "invalid.wav")
   Else
    OleLog = "date/time invalid"
    Call vcfContinue("10")
   End If
  Else
   sAMPM$ = ""
   If Left(Mailboxes("AMPMVariable"), 5) = "Input" Then sAMPM$ = Inp$(lineNo%, Right(Mailboxes
("AMPMVariable"), 1))

   If sAMPM$ = "" Or sDate$ = "" Or sTime$ = "" Then
    Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
    If Not Tonetables.NoMatch Then
     Mailbox$ = Tonetables("Assignment")
```

```
        OleLog = "date/time valid, going to " & Mailbox$
        Call vcfGoto(Mailbox$)
      Else
        OleLog = "date/time valid, going to Hangup"
        Call vcfGoto("Hangup")
      End If
    Else
      If sDate$ = "1" Then
        sDate$ = Format(Now, "mmdd")
      ElseIf sDate$ = "2" Then
        sDate$ = Format(DateAdd("d", 1, Now), "mmdd")
      ElseIf sDate$ = "3" Then
        sDate$ = Format(DateAdd("d", 2, Now), "mmdd")
      End If

      If sAMPM$ = 2 Then
        dDate = CDate(Left(sDate$, 2) & "/" & Right(sDate$, 2) & " " & Left(sTime$, 2) & ":" & Right
(sTime$, 2) & " PM")
      Else
        dDate = CDate(Left(sDate$, 2) & "/" & Right(sDate$, 2) & " " & Left(sTime$, 2) & ":" & Right
(sTime$, 2) & " AM")
      End If

      If dDate < Now Then dDate = DateAdd("yyyy", 1, dDate)

      If DateDiff("d", Now, dDate) > 100 Then
        If isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
          OleLog = "date/time invalid, playing invalid file"
          Call vcfPlayFile("10", "Current", "invalid.wav")
        Else
          OleLog = "date/time invalid"
          Call vcfContinue("10")
        End If
```

```
      Else
       Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '>'"
       If Tonetables.NoMatch Then Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and
Tone = '~'"
       If Not Tonetables.NoMatch Then
        Mailbox$ = Tonetables("Assignment")
        OleLog = "date/time valid, going to " & Mailbox$
        Call vcfGoto(Mailbox$)
       Else
        OleLog = "date/time valid, going to Hangup"
        Call vcfGoto("Hangup")
       End If
      End If
     End If
    End If

   Case "10"
    OleLog = "date/time invalid,"
    Tonetables FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Invalid'"
    If Tonetables.NoMatch Then
     Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
    End If

    If Not Tonetables.NoMatch Then
     Mailbox$ = Tonetables("Assignment")
     OleLog = OleLog & " going to " & Mailbox$
     Call vcfGoto(Mailbox$)
    Else
     OleLog = OleLog & " going to Hangup"
     Call vcfGoto("Hangup")
    End If

   End Select
```

```
End Sub

Private Sub extDoCallNumVerify()

 Dim lineNo%, Mailbox$
 Dim CallNum$, Forbidden As Boolean
 Dim res%, retstr As String * 255
 Dim s$

 lineNo% = OleData(0)
 Mailbox$ = OleData(1)

 If State$(lineNo%) = "" Then
  State$(lineNo) = "0"
 End If

 Select Case State$(lineNo%)

 Case "0"
  Forbidden = False

  Mailboxes.FindFirst "MailBoxNumber='" & Mailbox$ & "'"
  If Left(Mailboxes("CallNumVariable"), 5) = "Input" Then
   CallNum$ = Inp$(lineNo%, Right(Mailboxes("CallNumVariable"), 1))

   res% = GetPrivateProfileString("General", "ForbiddenPrefixes", "", retstr$, Len(retstr$) - 1, App.Path
& "\konnects.ini")
   s$ = Left(retstr$, res%)
   While InStr(s$, ",") <> 0
    If InStr(CallNum$, Left(s$, InStr(s$, ",") - 1)) = 1 Then Forbidden = True
    s$ = Right(s$, Len(s$) - InStr(s$, ","))
   Wend
   If s$ <> "" Then If InStr(CallNum$, s$) = 1 Then Forbidden = True
```

```
  End If

  If Forbidden Then
   If isfile(AppDir & "mboxes\" & Mailbox$ & "\invalid.wav") Then
    OleLog = "date/time invalid, playing invalid file"
    Call vcfPlayFile("10", "Current", "invalid.wav")
   Else
    OleLog = "date/time invalid"
    Call vcfContinue("10")
   End If
  Else
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
   If Not Tonetables.NoMatch Then
    Mailbox$ = Tonetables("Assignment")
    OleLog = "date/time valid, going to " & Mailbox$
    Call vcfGoto(Mailbox$)
   Else
    OleLog = "date/time valid, going to Hangup"
    Call vcfGoto("Hangup")
   End If
  End If

Case "10"
  OleLog = "date/time invalid,"
  Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = 'Forbidden'"
  If Tonetables NoMatch Then
   Tonetables.FindFirst "MailBoxNumber='" & Mailbox$ & "'" & "and Tone = '~'"
  End If
    If Not Tonetables.NoMatch Then
   Mailbox$ = Tonetables("Assignment")
   OleLog = OleLog & " going to " & Mailbox$
   Call vcfGoto(Mailbox$)
  Else
```

```
    OleLog = OleLog & " going to Hangup"
    Call vcfGoto("Hangup")
  End If

 End Select

End Sub

#End If
```

ประวัติผู้วิจัย

1. **ชื่อ (ภาษาไทย)** : ดร.ชนวัฒน์ ศรีสอ้าน

   ภาษาอังกฤษ : Chonawat Srisa-An, Ph.D.

2. **ประวัติการศึกษา**

ระดับปริญญา

ปีที่จบการศึกษา	(ตรี โท เอก และประกาศนียบัตร) ปริญญา	อักษรย่อ สาขาวิชาเอก และชื่อเต็ม	ชื่อสถาบันการศึกษา	ประเทศ
1989	ตรี	วศ.บ. ไฟฟ้า	ม.เชียงใหม่	ไทย
1991	โท	MSCE Computer Science	IIT	USA
1995	โท	MBA Finance	Loyola University	USA
1995	เอก	Ph.D. Computer Science	IIT	USA

3. **Award:**

ทุนศึกษาต่อต่างประเทศ โท-เอก ด้าน Computer Communication Network ในโครงการพัฒนาบุคคลกรที่ขาดแคลน ตามความต้องการของเทคโนโลยีพระจอมเกล้าลาดกระบัง

4. ประวัติการทำงาน

1996-present	อาจารย์ประจำ สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีสุรนารี
1995-1996	Technical Development Manager, บริษัทเอเชียอินโฟเน็ต (บริษัทในเครือของเทเลคอมเอเชีย)
1989-1990	อาจารย์ประจำ คณะวิศวกรรมคอมพิวเตอร์ ของ เทคโนโลยีพระจอมเกล้าลาดกระบัง