

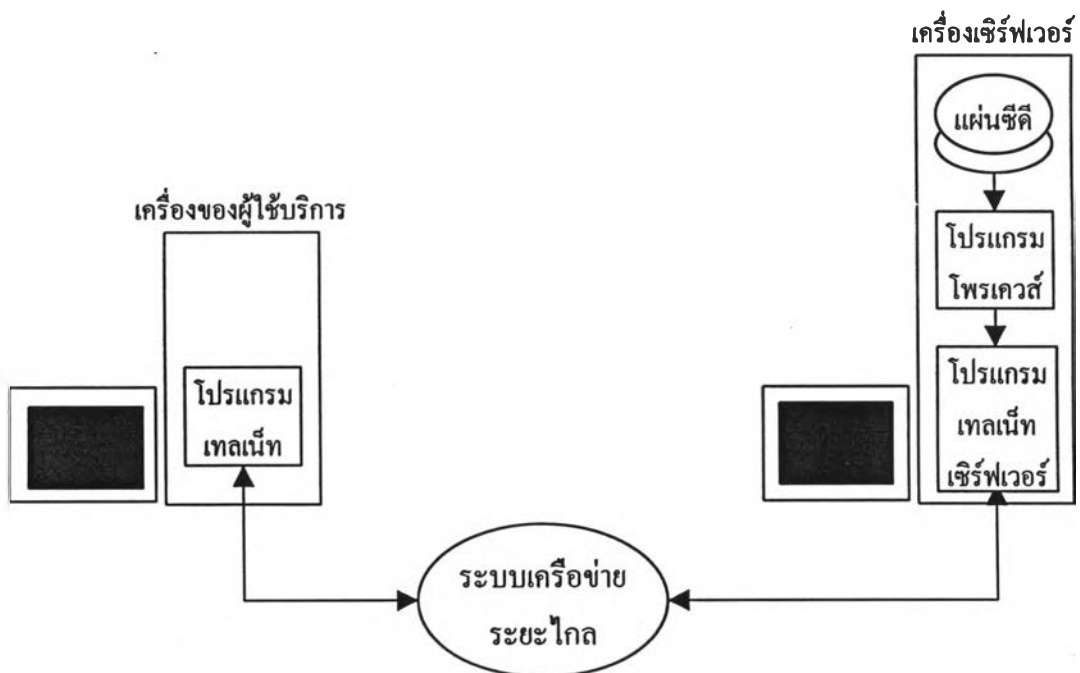
บทที่ 4

การเข้าถึงโปรแกรมสืบค้นที่เครื่องให้บริการแบบรีโมทแอกเซส

การเข้าถึงโปรแกรมสืบค้นโดยวิธีรีโมทแอกเซสนี้ เป็นวิธีการหนึ่งที่พัฒนาขึ้นเพื่อสามารถให้บริการฐานข้อมูลผ่านระบบเครือข่ายระยะไกล โดยการใช้โพรโทคอลเทลเน็ตเพื่อให้สามารถเข้าใช้งานฐานข้อมูลได้โดยสะดวก

4.1 แนวคิดของการเข้าถึงโปรแกรมสืบค้นที่เครื่องให้บริการแบบรีโมทแอกเซส

การเข้าถึงโปรแกรมสืบค้นโดยวิธีรีโมทแอกเซสนี้ จะมีการส่งข้อมูลผ่านระบบเครือข่ายเพียงแต่ ข้อมูลสำหรับการควบคุมเป็นพิมพ์ และการแสดงผลลัพท์ คาดการณ์ว่าวิธีนี้จะสามารถลดปริมาณข้อมูลที่มีการรับส่งผ่านทางเครือข่ายให้น้อยลง ทำให้เหมาะสมกับการใช้งานผ่านระบบเครือข่ายระยะไกลยิ่งขึ้น



รูปที่ 4.1 การเชื่อมต่อผ่านระบบเครือข่ายของการเข้าถึงฐานข้อมูลแบบรีโมทแอกเซส

เนื่องจากข้อมูลจากฐานข้อมูลมุลเอบีไออินฟอรม์ และฐานข้อมูลคิเอโอ ของบริษัทยูเอ็มไอ นั้นเป็นข้อมูลประเภทข้อความเพียงอย่างเดียว และ โปรแกรมโพรเซสซึ่งเป็น โปรแกรมสำหรับค้นหาข้อมูลจากฐานข้อมูลทั้งสองนั้น ทำงานในโหมดข้อความภายใต้ระบบปฏิบัติการไมโครซอฟต์

คอส ดังนั้น โพรโตคอลเทลเน็ตจึงมีความสามารถเพียงพอที่จะสนับสนุนการทำงานในการค้นหาข้อมูล และการแสดงผลจากโปรแกรมโพรวอร์

4.2 คุณสมบัติของโปรแกรมเทลเน็ต

ผู้พัฒนาได้เลือกใช้โปรแกรมเทลเน็ตเซิร์ฟเวอร์จากบริษัทออตต้าแมน(Ataman) สำหรับระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์เอ็นที ซึ่งสามารถเลียนแบบการทำงานของระบบปฏิบัติการ ไมโครซอฟต์คอส ได้ค่อนข้างสมบูรณ์ผ่านโปรโตคอลเทลเน็ต โดยใช้รหัสควบคุมการแสดงผลแบบแอนซี และการรับข้อมูลเป็นพิมพ์ต้องอาศัยรหัสพิเศษในการควบคุม ดังนั้นโปรแกรมเทลเน็ตที่จะสามารถใช้งานได้จะต้องมีการดัดแปลงหรือสร้างขึ้นใหม่ นอกจากนี้เพื่อลดขั้นตอนในการติดตั้งโปรแกรมเทลเน็ตให้กับผู้ใช้งาน จึงได้เลือกที่จะพัฒนาโปรแกรมเทลเน็ตบนภาษาจาวา ข้อดีของการพัฒนาโปรแกรมเทลเน็ตบนภาษาจาวาคือ ผู้ใช้สามารถที่จะใช้โปรแกรมเวปเบราว์เซอร์แบบใดก็ได้ที่มีความสามารถในการทำงานกับภาษาจาวา ก็จะได้รับโปรแกรมเทลเน็ตนี้โดยอัตโนมัติเมื่อเปิดหน้าโฮมเพจของฐานข้อมูลนี้ ดังนั้นการส่งโปรแกรมที่ได้ทำการแก้ไขปรับปรุงภายหลังก็สามารถทำได้โดยสะดวก

การพัฒนาโปรแกรมเทลเน็ตบนภาษาจาวาเพื่อที่สามารถทำงานได้กับโปรแกรมเทลเน็ตเซิร์ฟเวอร์จากบริษัทออตต้าแมน โดยแสดงผลได้ถูกต้องสวยงามต้องมีคุณสมบัติดังนี้

- 1 ต้องสามารถส่งรหัสพิเศษในการควบคุมเป็นพิมพ์ตามความต้องการของโปรแกรมเทลเน็ตเซิร์ฟเวอร์ได้ ตัวอย่างเช่นปุ่ม F1 จะต้องส่งรหัส Control-A และ 1 (ดูรายละเอียดของรหัสควบคุมพิเศษได้ในภาคผนวก ข.)
- 2 สามารถแสดงผลตามรหัสควบคุมแบบแอนซีได้โดยเฉพาะรหัสการแสดงสี เพื่อให้ใช้งานง่าย และใกล้เคียงกับการใช้แบบผู้ใช้คนเดียว
- 3 การแสดงผลของสัญลักษณ์บางตัวอาจจะต้องสร้างขึ้นใหม่ เนื่องจากข้อจำกัดของภาษาจาวาที่ไม่สามารถเรียกใช้ฟอนต์อื่น ได้นอกจากฟอนต์ของระบบ ดังนั้นการแสดงผลของสัญลักษณ์บางตัวเช่นสัญลักษณ์ที่ใช้ในการตีกรอบจะต้องถูกสร้างขึ้นใหม่
- 4 สิ่งที่ต้องเพิ่มเติมอีกประการหนึ่งคือเพิ่มความสามารถในการรับส่งเพิ่มข้อมูล เนื่องจากโปรแกรมเทลเน็ตเซิร์ฟเวอร์ไม่มีความสามารถในการรับส่งเพิ่มข้อมูลที่ได้จากการการ

บันทึกผลลัพธ์ลงบนฮาร์ดดิสก์ ดังนั้นจึงต้องใช้โปรโตคอลอื่นช่วยเช่น โปรโตคอลเอฟทีพี หรือ โปรโตคอลเซชทีทีพี

4.3 ขั้นตอนการพัฒนาโปรแกรมเทอร์มินัลบนภาษาจาวา

เพื่อให้สามารถใช้โปรแกรมเทอร์มินัลบนภาษาจาวานี้สามารถใช้งานได้จากโปรแกรมเว็บเบราว์เซอร์แบบใดก็ได้ที่มีความสามารถในการทำงานกับภาษาจาวา จึงจำเป็นที่จะต้องพัฒนาโปรแกรมนี้กับจาวาคอมไพเลอร์ เวอร์ชัน 1.0 ซึ่งในที่นี้จะเลือกใช้จาวาคอมไพเลอร์ เวอร์ชัน 1.0.2

การพัฒนาโปรแกรมเทอร์มินัลบนภาษาจาวาจะแยกการทำงานส่วนต่างๆ ออกเป็นคลาส โดยแต่ละคลาส มีหน้าที่การทำงานที่แยกจากกันตามฟังก์ชันการใช้งานดังต่อไปนี้

ส่วนโปรแกรมหลัก

- `Telnet.class` ทำหน้าที่สร้าง `TelnetIO.class` อ็อบเจกต์เพื่อใช้ในการติดต่อรับส่งข้อมูลจากโปรแกรมเทอร์มินัลเซิร์ฟเวอร์ `Ansi.class` อ็อบเจกต์เพื่อทำหน้าที่ติดต่อกับผู้ใช้ และ `Buttonbar.class` เพื่อสร้างและรับข้อมูลจากแถบเมนูและเป็นศูนย์กลางรับส่งคำสั่งจากคลาสต่างๆ
- `AppWrapper.class` ทำหน้าที่สร้างเนื้อหาในการแสดงผลและรับค่าต่างๆ จากภาษาเซชทีเอ็มแอล (HTML : HyperText Markup Language)

ส่วนติดต่อกับผู้ใช้

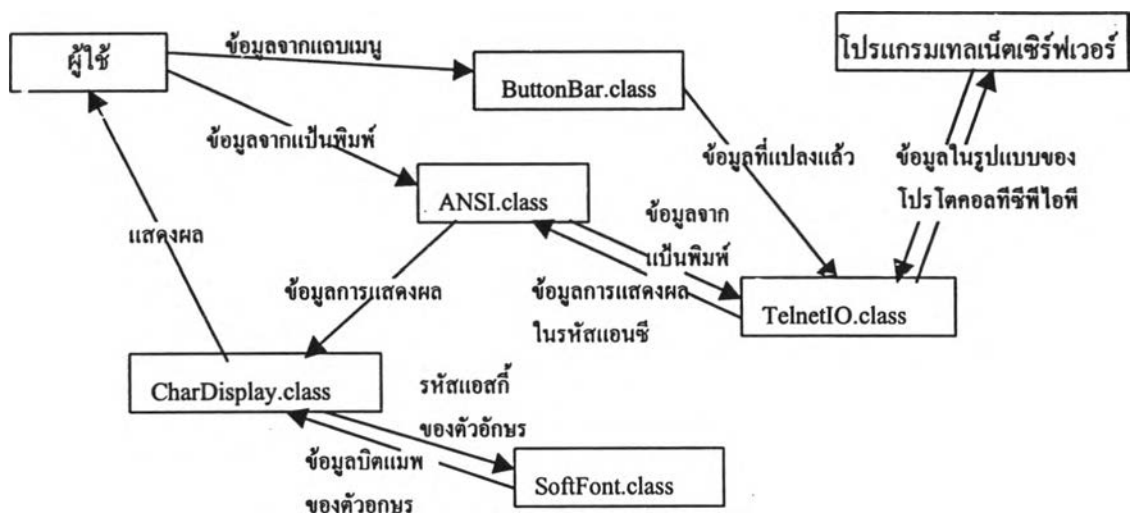
- `Ansi.class` ทำหน้าที่แปลงชุดคำสั่งควบคุมแบบแอนซีเพื่อส่งให้ `CharDisplay.class` แสดงผล เช่น สี ตัวชี้ตำแหน่งเป็นต้น และทำหน้าที่รับข้อมูลจากแป้นพิมพ์ แล้วแปลงรหัสของปุ่มควบคุมพิเศษได้แก่ F1..F12 ,Del ,Page Up ,Page Down ,Home ,End เป็นต้น เพื่อเป็นรหัสที่โปรแกรมเทอร์มินัลเซิร์ฟเวอร์ต้องการ
- `CharDisplay.Class` ทำหน้าที่สร้างหน้าต่างสำหรับแสดงผลและจัดการแสดงผลตามคำสั่งที่ส่งมาจาก `Ansi.class` เช่นการเพิ่มและลดตัวอักษร การสร้างเส้นตีกรอบ การเปลี่ยนแปลงขนาดตัวอักษร และการคัดลอกหน้าจอ เป็นต้น

- SoftFont.class ทำหน้าที่เก็บตัวอักษรพิเศษเพื่อใช้ในการแสดงผล เนื่องจากภาษาจาวาสามารถเรียกใช้สัญลักษณ์จากฟอนต์ของระบบเช่นฟอนต์คูเรีย (Courier) ซึ่งไม่มีตัวอักษรที่เหมาะสมสำหรับการสร้างตาราง
- ButtonBar.class ทำหน้าที่สร้างแถบเมนูซึ่งประกอบด้วยปุ่มกด และช่องข้อความ

ส่วนติดต่อกับ โปรแกรมเทลเน็ตเซิร์ฟเวอร์

- TelnetIO.class ซึ่งทำหน้าที่ ติดต่อกับ ส่วนของ โปรโตคอลเทลเน็ต

ความสัมพันธ์ระหว่างคลาสต่างๆ แสดงดังรูปต่อไปนี้



รูปที่ 4.2 แสดงการเชื่อมต่อของคลาสต่างๆ ใน โปรแกรมเทลเน็ต

เนื่องจากภาษาจาวาเป็นภาษาที่พัฒนาขึ้นมาเพื่อใช้กับระบบ Network ทำให้การเชื่อมต่อของ TCP/IP ในลักษณะของ Stream หรือ TCP ทำได้ง่ายด้วยคำสั่งเพียงคำสั่งเดียวดังตัวอย่างต่อไปนี้

```

socket = new Socket(address, port);
is = new BufferedInputStream(socket.getInputStream());
os = new BufferedOutputStream(socket.getOutputStream());
  
```

ตัวแปร `is` สร้างขึ้นเพื่อใช้รับข้อมูล และตัวแปร `os` สร้างขึ้นเพื่อใช้ส่งข้อมูลผ่านซ็อกเก็ตที่ได้สร้างไว้

และการยกเลิกการติดต่อสามารถทำได้โดยคำสั่งเดียวเช่นกันดังนี้

```
socket.close();
```

การรับข้อมูลจากซ็อกเก็ตโปรแกรมจะสร้างเธรด (thread) ขึ้นมาเพื่อคอยตรวจสอบ (Polling) ว่ามีข้อมูลเข้ามาหรือไม่โดยอาศัยคำสั่ง `read` มีลักษณะดังต่อไปนี้

```
connect statements...
```

```
...
```

```
t = new Thread(this);
```

```
t.setPriority(Thread.MIN_PRIORITY);
```

```
t.start();
```

```
...
```

```
While(t!=null)
```

```
Try {
```

```
    int count = is.available();
```

```
    byte buf[] = new byte[count];
```

```
    count = is.read(buf);
```

```
    if(count < 0) throw new IOException("Connection  
closed.");
```

```
}
```

เช่นเดียวกับการส่งข้อมูลผ่านซ็อกเก็ตลักษณะของการส่งข้อมูลจะเป็นดังนี้

```
os.write(buf);
```

```
os.flush();
```

โดยจะต้องมีคำสั่ง `flush` เพื่อลบข้อมูลที่ค้างอยู่หลังการส่ง

การรับข้อมูลจากผู้ใช้ในภาษาจาวาจะมีการดักจับเหตุการณ์ที่เกิดขึ้น คล้ายกับการเขียนโปรแกรมในระบบปฏิบัติการไมโครซอฟต์วินโดว โดยมีฟังก์ชันต่อไปนี้ทำหน้าที่ดักจับเหตุการณ์อยู่ตลอด

```
public boolean handleEvent(Event evt)
```

เมื่อฟังก์ชันนี้ทำงานตัวแปร `evt` ซึ่งเป็นตัวแปรแบบอ็อบเจกต์ที่มีตัวแปรย่อยทั้งแบบที่เป็นฟังก์ชันและตัวแปรย่อยแบบธรรมดาเช่นตัวแปรย่อย `id` ซึ่งจะเก็บค่าของเหตุการณ์ที่เกิดขึ้นตัวอย่างเช่น

```
if(evt.id == Event.MOUSE_ENTER)
    { display.requestFocus(); return true; }
```

และยังมีตัวแปรย่อย `key` ซึ่งเก็บค่าของแป้นพิมพ์ และตัวแปรย่อย `modifiers` ซึ่งเก็บค่าของปุ่มพิเศษบนแป้นพิมพ์ ได้แก่ Shift (`evt.modifiers & 1 == 1`) , Control (`evt.modifiers & 2 == 2`) และ Alternate (`evt.modifiers & 8 == 8`)

สำหรับรหัสของปุ่มพิเศษอื่นๆ บนแป้นพิมพ์เช่น F1..F12 Page Up Page Down เป็นต้น จะมีค่า `evt.key > 1000` ตัวอย่างเช่น F1..F12 จะมีค่า `evt.key` เท่ากับ 1001 ถึง 1012⁶

การส่งข้อมูลของปุ่มพิเศษเหล่านี้ออกไปเป็นค่าที่โปรแกรมเทลเน็ตเซิร์ฟเวอร์ต้องการ จะต้องส่งข้อมูลที่มีค่าเท่ากับ 1 (Control-A) ออกไป 1 ไบต์ และตามด้วยค่าต่างๆ (ดูรายละเอียดของรหัสควบคุมพิเศษได้ในภาคผนวก ข.)

สำหรับค่าคงที่ต่างๆ ที่บอกสถานะของ `evt.id` , `evt.key` และ `evt.modifier` สามารถดูได้จากเอกสารของ `java.awt.Event.class` ซึ่งสามารถหาได้จากเวปเพจ ของภาษาจาวาเอง (java.sun.com)

⁶James Gosling and Frank Yellin, "Class Event," in Java API Documentation Version 1.0.2 (1996), p. 1.

การแสดงผลของโปรแกรมภาษาจาวานั้นจะทำการแสดงผลผ่านทางฟังก์ชันที่มีชื่อว่า paint ซึ่งมีการประกาศใช้ดังนี้

```
public synchronized void paint(Graphics g)
```

จะได้ตัวแปร g ที่เป็นตัวแปรแบบอ็อบเจกต์สามารถใส่ฟังก์ชันใน อ็อบเจกต์ g ได้เช่น

- g.setColor(color); ใช้สำหรับตั้งค่าของสีของตัวอักษรที่จะเขียนลงไปบนหน้าจอ
- g.fillRect(x,y, charWidth, charHeight); สำหรับวาดรูปสี่เหลี่ยมลงบนหน้าจอ โดยมีตัวแปร charWidth และ charHeight สำหรับกำหนดขนาดความกว้างและความยาวของรูปสี่เหลี่ยม
- g.drawChars(charArray[0],offset,nChar,x,y); สำหรับวาดตัวอักษรลงบนหน้าจอ โดยมีตัวแปร charArray เป็นตัวแปรที่เก็บค่าของตัวอักษรที่จะแสดงบนหน้าจอ offset เป็นตัวแปรสำหรับบอกตำแหน่งของตัวอักษร nChar สำหรับบอกจำนวนตัวอักษรที่วาดลงบนหน้าจอ และตัวแปร x และ y ใช้สำหรับบอกตำแหน่งเริ่มต้น

รายละเอียดของอ็อบเจกต์ Graphic สามารถดูได้จากเอกสารของ java.awt.Graphics

การสร้างปุ่มสำหรับให้เมาส์คลิกและกล่องข้อความทำได้โดยคำสั่งดังนี้

```
Button b = new Button("Label");
TextField t = new TextField("Label", input.length,
size);
```

หลังจากนั้นให้ใช้คำสั่ง add เช่น add(b); add(t); เพื่อวางปุ่มหรือกล่องข้อความลงไปบนหน้าจอ ซึ่งการจะให้ปุ่มนั้นอยู่ในบริเวณที่กำหนดให้ ต้องใช้คำสั่ง Resize แต่ในการพัฒนานี้ จะใช้วิธีกำหนดคลาสที่สร้าง ปุ่มนี้เป็นชนิดพานเนลดังนี้ public class ButtonBar extends Panel เพื่อที่จะได้วางตำแหน่งการแสดงผลของคลาสนี้ไว้ด้านบนของหน้าจอแสดงผลแล้วเพิ่มปุ่มกล่องข้อความในเนื้อที่ที่ได้กำหนดไว้ แล้วปล่อยให้กลไกของภาษา Java กำหนดตำแหน่งให้โดยอัตโนมัติ

เนื่องจากการเข้าถึงโปรแกรมสืบทอดที่เครื่องให้บริการแบบรีโมทแอกเซส การเก็บข้อมูลที่
ได้จากการสืบทอดจะถูกเก็บลงบนเครื่องเซิร์ฟเวอร์ การดึงข้อมูลที่เก็บไว้สามารถทำได้โดยการ
กำหนดสารบบ สำหรับการเก็บข้อมูลให้สามารถเข้าถึงได้โดยผ่านทางโปรแกรมเอพีทีพี หรือให้วิธี
สร้างปุ่มสำหรับดึงข้อมูลผ่านโปรโตคอลเฮชทีทีพี โดยการใช้คำสั่ง `showDocument` รูปแบบ
การสั่งใช้งานเป็นดังนี้

```
getAppletContext().showDocument( new URL ("http://" +  
address) );
```

เมื่อโปรแกรมภาษาจาวาได้รับคำสั่งนี้จะทำการสร้างหน้าต่างของเว็บเบราว์เซอร์ขึ้นใหม่และทำการ
ดึงเพิ่มข้อมูลนั้น ซึ่งเพิ่มข้อมูลนี้ เป็นเพิ่มข้อมูลแบบตัวอักษรดังนั้นเพื่อให้การแสดงผลถูกต้องจะ
ต้องเพิ่มชนิดของข้อมูลไว้ที่คอนคั้นของ เพิ่มข้อมูล เช่น

```
Content-Type: text/plain; charset=us-ascii
```

ซึ่งสามารถที่จะใช้โปรแกรมภาษาสคริปช่วยได้ โดยในที่นี้จะใช้เพิ่มข้อมูลแบบแบบเป็นตัวจัดการ

4.4 สรุป

ข้อดีของการเข้าถึง โปรแกรมสืบทอดที่เครื่องให้บริการแบบรีโมทแอกเซสคือการรับส่งข้อมูล
ในระหว่างการสืบทอดจะมีน้อยกว่าวิธีการเข้าถึงฐานข้อมูลแบบเน็ตเวิร์คไครฟ์ เพราะข้อมูลการ
แสดงผลในลักษณะของโหมคข้อความมีน้อยทำให้สามารถใช้กับเครือข่ายที่มีความเร็วต่ำได้ และ
การใช้ภาษาจาวาในการพัฒนาช่วยให้การส่ง โปรแกรมที่มีการปรับปรุงให้แก่ผู้ใช้ทำได้ง่ายขึ้น แต่
ยังมีข้อเสียคือใช้งานได้กับฐานข้อมูลที่มีโปรแกรมสำหรับการสืบทอดทำงานใน โหมคตัวอักษรเท่านั้น
นั้น และการรับส่งเพิ่มข้อมูลที่บันทึกผลการสืบทอดยังมีขั้นตอนที่ยุ่งยาก

ในกรณีที่ฐานข้อมูลนั้นมีข้อมูลที่เป็นรูปภาพ โปรแกรมสืบทอดก็จะทำงานในโหมคกราฟฟิก
ทำให้ต้องเปลี่ยนไปใช้วิธีอื่นเช่นการใช้ระบบปฏิบัติการวินเฟรม(winframe) หรือระบบปฏิบัติการ
ไมโครซอฟต์วินโดวส์เอ็นทีเทอร์มินอลเซิร์ฟเวอร์เอดิชั่น ซึ่งมีความสามารถสูงกว่าเช่นมีความ
สามารถในการแสดงผลข้อมูลชนิดกราฟฟิก การส่งสัญญาณควบคุมเป็นพิมพ์และเมาส์ และการรับ
ส่งเพิ่มข้อมูล เนื่องจากระบบปฏิบัติการวินเฟรมจำทำการเชื่อมต่อไครฟ์ของเครื่องผู้ใช้โดยผ่าน
ระบบเน็ตเวิร์คไครฟ์ให้โดยอัตโนมัติ รวมทั้งทรัพยากรอื่นๆ เช่น เครื่องพิมพ์ เป็นต้น

เนื่องจากความสามารถที่เพิ่มขึ้นในระบบปฏิบัติการวินเฟรมทำให้จำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงเพื่อทำหน้าที่เป็นเซิร์ฟเวอร์ และระบบเครือข่ายจะต้องมีความเร็วพอสมควรเช่น ถ้าต้องการ การแสดงผลที่ความละเอียด 640 x 480 จุด 256 สี ระบบเครือข่ายจะต้องมีความเร็วไม่ต่ำกว่า 14400 บิตต่อวินาทีเป็นต้น