



รายการอ้างอิง

ภาษาไทย

- กลางเดือน โพนนา. ระบบสนับสนุนการตัดสินใจสำหรับการวางแผนการผลิตในโรงงานแปรรูปเนื้อไก่. วิทยานิพนธ์ปริญญามหาบัณฑิต บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2533.
- จิรภัทร ราศรี. การวางแผนการผลิตและการจัดการพัสดุคงคลังของโรงงานอุตสาหกรรมผลิตท่อโพลีเอทิลีน. วิทยานิพนธ์ปริญญามหาบัณฑิต บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2539.
- เจริญ สุนทราวาณิชย์. การวางแผนการผลิตและพัสดุคงคลังสำหรับโรงงานกระดาษเหนียว. วิทยานิพนธ์ปริญญามหาบัณฑิต บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2529.
- ชุมพล ศฤงคารศิริ. การวางแผนและควบคุมการผลิต. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2535.
- दनัย จินดารัตน์. ระบบผู้เชี่ยวชาญสำหรับการวางแผนการผลิตในโรงงานแผ่นวงจรพิมพ์. วิทยานิพนธ์ปริญญามหาบัณฑิต บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2533.
- บรรเลง ศรีนิล. เทคโนโลยีพลาสติก. พิมพ์ครั้งที่ 13. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2542
- พิชิต สุขเจริญพงษ์. การจัดการวิศวกรรมการผลิต. บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2540
- วสันต์ ฐิติภูมิเดชา. การจัดลำดับการผลิตสำหรับการผลิตพีวีซีคอมพาวด์เกรดสายเคเบิล. วิทยานิพนธ์ปริญญามหาบัณฑิต บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2538.
- วิโรจน์ เตชะวิญญธรรม. งานฉีดพลาสติก. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2540.

ภาษาอังกฤษ

- Baker, K.R. Introduction to sequencing and scheduling. New York: John Wiler & Sons, 1974.
- Rubin, Irvin I. Injection molding theory & practice. New York: Wiley, 1972.

ภาคผนวก

ภาคผนวก ก
เวลาเปลี่ยนแม่พิมพ์

ตารางที่ ก-1 เวลาเปลี่ยนแม่พิมพ์สำหรับเครื่องฉีดพลาสติกขนาด 30 ตัน

วันที่	จาก	ไป	เวลา(นาที)
21 พฤศจิกายน 2545	0003-2-9-027-02	038X-0-0-D16-01	13
3 ธันวาคม 2545	082T-0-0-810-01	120H-0-0-584-Y2	14
21 กันยายน 2545	1416-0-0-030-01	AT06-0-0-030-01	16
5 กันยายน 2545	1C84-0-C-584-02	0515-2-0-080-01	17
11 ธันวาคม 2545	21S0-8-0-027-02	1C85-0-C-584-46	16
3 ตุลาคม 2545	24B9-A-0-130-16	26Z1-0-0-584-02	18
4 กันยายน 2545	3473-8-0-027-01	0003-2-9-027-02	14
10 ตุลาคม 2545	3473-8-0-027-01	4931-0-0-439-02	18
21 ธันวาคม 2545	3S16-0-0-584-01	FG33-0-0-NWO-01	19
1 ตุลาคม 2545	FG01-0-0-027-01	13B7-0-0-D16-Y2	14
14 กันยายน 2545	FG29-0-0-NWO-01	SA09-0-0-586-01	18
26 ธันวาคม 2545	FG40-0-0-ABS-02	JM10-0-0-PS1-02	13
19 ตุลาคม 2545	JM52-0-0-824-MS	JM57-0-0-974-S1	14
30 ตุลาคม 2545	JM71-0-0-920-01	SA09-0-0-586-01	16
13 พฤศจิกายน 2545	TA02-0-C-200-W1	JM69-0-0-A01-08	18
9 พฤศจิกายน 2545	TH01-0-0-030-58	TS01-0-0-395-01	17
เฉลี่ย			16
ส่วนเบี่ยงเบนมาตรฐาน			2.03

ตารางที่ ก-2 เวลาเปลี่ยนแม่พิมพ์สำหรับเครื่องฉีดพลาสติกขนาด 55 ตัน

วันที่	จาก	ไป	เวลา(นาที)
13 กันยายน 2545	0245-5-0-027-01	DK05-0-0-541-01	17
19 ธันวาคม 2545	1Y85-0-0-027-01	1F55-1-0-027-01	22
25 ตุลาคม 2545	1Z04-0-0-561-02	TP10-0-0-080-02	24
4 กันยายน 2545	23H5-0-0-586-02	4345-0-0-SM1-10	25
26 กันยายน 2545	24Z5-1-0-080-02	3923-5-0-SM2-09	20
2 ตุลาคม 2545	266B-0-C-15M-65	2707-F-9-SM1-10	24
27 ธันวาคม 2545	2703-F-0-586-W1	2709-F-0-SM1-10	23
3 ตุลาคม 2545	2709-F-0-586-R3	FG04-0-0-027-02	25
18 กันยายน 2545	2735-0-0-027-02	TP08-0-0-586-GY	19
11 ตุลาคม 2545	FG02-0-U-586-02	SS04-0-0-080-02	18
21 พฤศจิกายน 2545	JM18-0-C-ABT-MG	JM51-0-0-824-MS	20
6 ธันวาคม 2545	NH01-0-0-586-01	NH07-0-C-561-67	25
18 พฤศจิกายน 2545	NH07-0-C-561-47	JM26-0-0-LP1-02	19
11 ธันวาคม 2545	NH07-0-C-561-67	0153-0-0-027-01	23
12 กันยายน 2545	SZ01-0-0-030-02	3923-5-0-SM1-10	19
10 ตุลาคม 2545	UZ04-0-C-564-02	2P50-0-0-027-02	21
เฉลี่ย			21
ส่วนเบี่ยงเบนมาตรฐาน			2.60

ตารางที่ ก-3 เวลาเปลี่ยนแม่พิมพ์สำหรับเครื่องฉีดพลาสติกขนาด 80 ตัน

วันที่	จาก	ไป	เวลา(นาที)
6 ธันวาคม 2545	034N-0-0-PA6-01	2859-F-9-586-02	28
24 ตุลาคม 2545	034P-0-0-PA6-01	HC03-0-0-BWE-02	33
12 กันยายน 2545	1763-5-0-080-01	22Y4-0-0-080-20	32
11 ธันวาคม 2545	3597-F-9-SM2-09	3818-5-0-586-R6	27
18 กันยายน 2545	3599-F-9-586-39	TC01-0-0-584-01	33
12 กันยายน 2545	4475-0-0-027-01	1763-5-0-080-01	34
11 พฤศจิกายน 2545	4475-0-0-999-01	NK01-0-0-D07-02	31
5 กันยายน 2545	JM04-0-0-030-01	3599-F-9-586-02	33
30 ตุลาคม 2545	NH13-0-0-564-D8	NH15-0-0-561-02	31
4 พฤศจิกายน 2545	NH17-0-C-564-D8	NH20-0-C-564-I2	32
18 พฤศจิกายน 2545	NU03-0-0-586-02	NU01-0-0-027-01	35
22 พฤศจิกายน 2545	NU08-0-0-027-01	0061-2-0-518-01	33
19 กันยายน 2545	TK06-A-0-030-02	TK07-0-0-030-02	30
26 กันยายน 2545	TR38-5-0-SM1-10	TT08-0-0-586-02	29
2 ตุลาคม 2545	UN47-2-0-030-01	TY04-0-0-UBE-01	34
3 ตุลาคม 2545	UZ01-0-C-586-M3	UT02-0-0-LP1-W2	32
		เฉลี่ย	32
		ส่วนเบี่ยงเบนมาตรฐาน	2.11

ตารางที่ ก-4 เวลาเปลี่ยนแม่พิมพ์สำหรับเครื่องฉีดพลาสติกขนาด 100 ตัน

วันที่	จาก	ไป	เวลา(นาที)
13 กันยายน 2545	045Z-0-0-030-01	13A6-8-0-030-02	37
11 ธันวาคม 2545	266A-0-C-15M-65	266D-0-0-15M-65	40
1 ตุลาคม 2545	392S-0-C-PM1-10	266A-0-C-15M-65	36
3 ตุลาคม 2545	392T-0-C-PM1-10	392U-0-C-PM1-10	35
26 ธันวาคม 2545	NH09-0-C-561-64	NU05-0-0-027-02	40
13 พฤศจิกายน 2545	NH11-0-0-561-01	NH20-0-C-564-I2	33
24 ตุลาคม 2545	NU05-0-0-027-02	TH02-0-0-030-01	34
30 ตุลาคม 2545	SS09-0-0-518-02	SS08-0-0-518-02	35
19 ตุลาคม 2545	SS21-0-0-PSO-02	3S48-8-0-D17-R7	37
3 ธันวาคม 2545	TC04-A-0-19H-02	NH08-0-0-561-47	36
14 กันยายน 2545	TC10-0-0-145-02	TC09-0-0-145-02	39
11 ธันวาคม 2545	TC11-0-0-145-02	NH16-0-0-564-I2	40
18 กันยายน 2545	TK08-0-0-586-01	TY05-0-0-027-60	35
5 กันยายน 2545	TT04-0-0-TTO-36	TY01-0-0-UBE-02	35
12 กันยายน 2545	TT05-0-0-561-62	UN54-1-0-D16-46	40
21 พฤศจิกายน 2545	TY11-0-0-NA2-02	392T-0-C-PM1-10	39
		เฉลี่ย	37
		ส่วนเบี่ยงเบนมาตรฐาน	2.23

ตารางที่ ก-5 เวลาเปลี่ยนแม่พิมพ์สำหรับเครื่องฉีดพลาสติกขนาด 150 ตัน

วันที่	จาก	ไป	เวลา(นาที)
18 กันยายน 2545	2D49-0-027-52	TC05-A-0-19H-02	45
18 กันยายน 2545	317R-8-0-564-01	HK01-A-C-584-W1	48
30 พฤศจิกายน 2545	HK01-A-C-584-W1	HK02-0-C-584-W1	40
12 กันยายน 2545	HK01-A-C-PF1-09	HK02-0-C-PF1-09	44
25 ตุลาคม 2545	HK02-0-C-999-09	1259-0-0-027-02	39
30 พฤศจิกายน 2545	JM60-0-0-824-MS	JM61-0-0-A01-08	47
3 ธันวาคม 2545	JM62-0-0-810-01	TK03-0-0-500-02	42
3 ธันวาคม 2545	NK02-0-0-D07-02	SS11-0-0-E17-01	40
12 กันยายน 2545	NK02-0-0-D07-02	ST02-0-0-080-01	40
4 กันยายน 2545	NK06-0-0-99-01	SS10-0-0-080-L1	40
3 ตุลาคม 2545	PC01-0-C-586-01	JM65-0-0-PS1-02	39
6 ธันวาคม 2545	SH38-5-9-SM2-09	TG01-0-0-564-02	39
18 พฤศจิกายน 2545	SS02-0-0-PS2-02	SL01-0-C-586-01	44
11 ธันวาคม 2545	SS22-C-0-PSO-02	SS30-2-0-PSO-02	44
25 ตุลาคม 2545	TG03-0-0-564-02	SS22-C-0-PSO-02	44
19 ตุลาคม 2545	TT09-0-0-561-01	NK02-0-0-D07-02	40
		เฉลี่ย	42
		ส่วนเบี่ยงเบนมาตรฐาน	3.04

ภาคผนวก ข

เวลาทำความสะอาดกระบอกล้างพลาสติก

เนื่องจากการทำความสะอาดกระบอกลีดพลาสติก จำเป็นต้องพิจารณาทั้งชนิดและสีของวัตถุดิบที่ผลิตต่อเนื่องกัน หัวหน้าฝ่ายเทคนิคของโรงงานซึ่งเป็นผู้ที่มีความชำนาญเรื่องคุณสมบัติของพลาสติก แนะนำให้แบ่งกลุ่มชนิดและสีของวัตถุดิบ ดังนี้

1. ชนิดของเม็ดพลาสติก แบ่งได้ 4 กลุ่ม คือ

กลุ่มที่ 1 ประกอบด้วย ABS, PP, POM และ PS

กลุ่มที่ 2 ประกอบด้วย NYLON6 และ PBT

กลุ่มที่ 3 ประกอบด้วย NYLON66 และ PC

กลุ่มที่ 4 ประกอบด้วย PVC

กลุ่มที่ 1						กลุ่มที่ 2	กลุ่มที่ 3	กลุ่มที่ 4
018	172	780	15M	E17	PS6	027	030	170
019	181	810	A01	LP1	PS7	200	036	189
050	440	814	ABS	M90	PSO	294	140	439
063	500	824	ABT	NW0	SM1	340	395	737
064	561	825	AT1	PF1	SM2	518	NA2	JSR
065	564	830	AT2	PM1	SM3	541	PA6	PVC
080	584	833	AT3	PPT	SM5	999	SN1	TT0
100	586	839	AW1	PS1	SM6	18H	UBE	TT1
104	640	920	BWE	PS2	SPP	19H		TT2
130	646	940	D07	PS3	20H			TT3
145	656	974	D16	PS4	22H			TT4
171	680	11J	D17	PS5	B3S			

2. สีของวัตถุดิบ แบ่งเป็น 4 กลุ่ม คือ

Natural	01
Light I	71, 72, 73, AH, BG, D8, I2, P1, W1, W1, W3, WH
Light II	03, 04,05,08,37,40,41,43,44, 45, 46, 47, 48, 49, 50, 60, 61, 63, 64, 65, 66, 67, 68, 69, BB, D1, D2, D3, D4,D7,D9,DG,GY,L1, L2, M2, M3,MB, MG, MS, S1, S2, S3, S4,Y2,Y4,Y5,Y6
Dark	02, 07, 09, 10, 11 ,15 , 16, 20, 23, 24, 25, 26, 27, 29,30, 33, 36, 38, 39, 52, 57, 58, B1, B4, B5, B6, B7, B8, G1, I1, R1, R2, R3, R4,R5, R6, R7, R9, T1, T4, T7, V1

เมื่อแบ่งกลุ่มชนิดและสีของวัตถุดิบแล้ว จึงพิจารณาต่อว่า ชิ้นงานที่ฉีดต่อเนื่องกันนั้น ผลิต จาก กลุ่มและสีใด ไป กลุ่มและสีใดและ จัดอยู่ในการทำความสะอาดกรณีใด โดยรหัสชนิดวัตถุดิบคือรหัสตำแหน่งที่ 7-9, รหัสสีคือรหัสตำแหน่งที่ 10-12 ของรหัสผลิตภัณฑ์ เช่น ผลิตจาก 038X-0-0-D16-01 ไป 0003-2-9-027-02 เป็นการเปลี่ยน จาก กลุ่มที่ 1 Nat ไป กลุ่มที่ 2 Dark ดังนั้น ต้องใช้การทำความสะอาดกรณีที่ 2 กล่าวคือใช้เม็ดพลาสติก AW564 แทนที่ก่อนแล้วจึงตามด้วยพลาสติกที่ต้องการ

ตารางที่ ข-1 การพิจารณาขั้นตอนการทำความสะอาดสำหรับวัตถุดิบที่ผลิตต่อเนื่องกัน

จาก	ไป	การทำความสะอาด
1-Nat	1-Nat	1
	1-Light I	1
	1-Light II	1
	1-Dark	1
	2-Nat	2
	2-Light I	2
	2-Light II	2
	2-Dark	2
	3-Nat	2
	3-Light I	2
	3-Light II	2
	3-Dark	2
	4-Nat	2
	4-Light I	2
	4-Light II	2
	4-Dark	2

จาก	ไป	การทำความสะอาด
1-Light I	1-Nat	2
	1-Light I	1
	1-Light II	1
	1-Dark	1
	2-Nat	2
	2-Light I	2
	2-Light II	2
	2-Dark	2
	3-Nat	2
	3-Light I	2
	3-Light II	2
	3-Dark	2
	4-Nat	2
	4-Light I	2
	4-Light II	2
	4-Dark	2

จาก	ไป	การทำความสะอาด
1-Light II	1-Nat	3
	1-Light I	2
	1-Light II	1
	1-Dark	1
	2-Nat	2
	2-Light I	2
	2-Light II	2
	2-Dark	2
	3-Nat	2
	3-Light I	2
	3-Light II	2
	3-Dark	2
	4-Nat	2
	4-Light I	2
	4-Light II	2
	4-Dark	2

จาก	ไป	การทำความสะอาด
1-Dark	1-Nat	3
	1-Light I	3
	1-Light II	2
	1-Dark	1
	2-Nat	3
	2-Light I	3
	2-Light II	3
	2-Dark	2
	3-Nat	3
	3-Light I	3
	3-Light II	3
	3-Dark	2
	4-Nat	3
	4-Light I	3
	4-Light II	3
	4-Dark	2

ตารางที่ ข-1 การพิจารณาขั้นตอนการทำความสะอาดสำหรับวัตถุดิบที่ผลิตต่อเนื่องกัน(ต่อ)

จาก	ไป	การทำความสะอาด
2-Nat	1-Nat	2
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	1
	2-Light I	1
	2-Light II	1
	2-Dark	1
	3-Nat	4
	3-Light I	4
	3-Light II	4
	3-Dark	4
	4-Nat	2
	4-Light I	2
	4-Light II	2
	4-Dark	2

จาก	ไป	การทำความสะอาด
2-Light I	1-Nat	2
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	2
	2-Light I	1
	2-Light II	1
	2-Dark	1
	3-Nat	4
	3-Light I	4
	3-Light II	4
	3-Dark	4
	4-Nat	2
	4-Light I	2
	4-Light II	2
	4-Dark	2

จาก	ไป	การทำความสะอาด
2-Light II	1-Nat	3
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	3
	2-Light I	2
	2-Light II	1
	2-Dark	1
	3-Nat	5
	3-Light I	5
	3-Light II	4
	3-Dark	4
	4-Nat	3
	4-Light I	2
	4-Light II	2
	4-Dark	2

จาก	ไป	การทำความสะอาด
2-Dark	1-Nat	3
	1-Light I	3
	1-Light II	3
	1-Dark	2
	2-Nat	3
	2-Light I	3
	2-Light II	2
	2-Dark	1
	3-Nat	5
	3-Light I	5
	3-Light II	5
	3-Dark	4
	4-Nat	3
	4-Light I	3
	4-Light II	3
	4-Dark	2

ตารางที่ ข-1 การพิจารณาขั้นตอนการทำความสะอาดสำหรับวัตถุเคปที่ผลิตต่อเนื่องกัน(ต่อ)

จาก	ไป	การทำความสะอาด
3-Nat	1-Nat	2
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	4
	2-Light I	4
	2-Light II	4
	2-Dark	4
	3-Nat	1
	3-Light I	1
	3-Light II	1
	3-Dark	1
	4-Nat	5
	4-Light I	5
	4-Light II	5
	4-Dark	5

จาก	ไป	การทำความสะอาด
3-Light I	1-Nat	2
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	4
	2-Light I	4
	2-Light II	4
	2-Dark	4
	3-Nat	2
	3-Light I	1
	3-Light II	1
	3-Dark	1
	4-Nat	5
	4-Light I	5
	4-Light II	5
	4-Dark	5

จาก	ไป	การทำความสะอาด
3-Light II	1-Nat	3
	1-Light I	3
	1-Light II	2
	1-Dark	2
	2-Nat	5
	2-Light I	5
	2-Light II	4
	2-Dark	4
	3-Nat	3
	3-Light I	2
	3-Light II	1
	3-Dark	1
	4-Nat	5
	4-Light I	5
	4-Light II	5
	4-Dark	5

จาก	ไป	การทำความสะอาด
3-Dark	1-Nat	3
	1-Light I	3
	1-Light II	3
	1-Dark	2
	2-Nat	5
	2-Light I	5
	2-Light II	5
	2-Dark	4
	3-Nat	3
	3-Light I	3
	3-Light II	2
	3-Dark	1
	4-Nat	5
	4-Light I	5
	4-Light II	5
	4-Dark	5

ตารางที่ ข-1 การพิจารณาขั้นตอนการทำความสะอาดสำหรับวัตถุดิบที่ผลิตต่อเนื่องกัน(ต่อ)

จาก	ไป	การทำความสะอาด
4-Nat	1-Nat	2
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	4
	2-Light I	4
	2-Light II	4
	2-Dark	4
	3-Nat	4
	3-Light I	4
	3-Light II	4
	3-Dark	4
	4-Nat	1
	4-Light I	1
	4-Light II	1
	4-Dark	1

จาก	ไป	การทำความสะอาด
4-Light I	1-Nat	2
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	4
	2-Light I	4
	2-Light II	4
	2-Dark	4
	3-Nat	5
	3-Light I	5
	3-Light II	5
	3-Dark	5
	4-Nat	2
	4-Light I	1
	4-Light II	1
	4-Dark	1

จาก	ไป	การทำความสะอาด
4-Light II	1-Nat	3
	1-Light I	2
	1-Light II	2
	1-Dark	2
	2-Nat	5
	2-Light I	5
	2-Light II	4
	2-Dark	4
	3-Nat	5
	3-Light I	5
	3-Light II	5
	3-Dark	5
	4-Nat	3
	4-Light I	2
	4-Light II	1
	4-Dark	1

จาก	ไป	การทำความสะอาด
4-Dark	1-Nat	3
	1-Light I	3
	1-Light II	3
	1-Dark	2
	2-Nat	5
	2-Light I	5
	2-Light II	5
	2-Dark	4
	3-Nat	5
	3-Light I	5
	3-Light II	5
	3-Dark	5
	4-Nat	3
	4-Light I	3
	4-Light II	2
	4-Dark	1

ตารางที่ ข-2 กรณีที่จัดขึ้นงานต่อเนื่องได้ทันที

วันที่	จาก	ไป	เวลา(นาที)
4 กันยายน 2545	23H5-0-0-586-02	4345-0-0-SM1-10	0
4 กันยายน 2545	3473-8-0-027-01	0003-2-9-027-02	0
12 กันยายน 2545	1763-5-0-080-01	22Y4-0-0-080-20	0
13 กันยายน 2545	0245-5-0-027-01	DK05-0-0-541-01	0
13 กันยายน 2545	045Z-0-0-030-01	13A6-8-0-030-02	0
14 กันยายน 2545	FG29-0-0-NW0-01	SA09-0-0-586-01	0
18 กันยายน 2545	317R-8-0-564-01	HK01-A-C-584-W1	0
18 กันยายน 2545	HK01-A-C-584-W1	HK02-0-C-584-W1	0
21 กันยายน 2545	1416-0-0-030-01	AT06-0-0-030-01	0
26 กันยายน 2545	24Z5-1-0-080-02	3923-5-0-SM2-09	0
1 ตุลาคม 2545	392T-0-C-PM1-10	392U-0-C-PM1-10	0
2 ตุลาคม 2545	266B-0-C-15M-65	2707-F-9-SM1-10	0
3 ตุลาคม 2545	24B9-A-0-130-16	26Z1-0-0-584-02	0
19 ตุลาคม 2545	JM52-0-0-824-MS	JM57-0-0-974-S1	0
25 ตุลาคม 2545	1Z04-0-0-561-02	TP10-0-0-080-02	0
30 ตุลาคม 2545	JM71-0-0-920-01	SA09-0-0-586-01	0
11 พฤศจิกายน 2545	FG02-0-U-586-02	SS04-0-0-080-02	0
21 พฤศจิกายน 2545	JM18-0-C-ABT-MG	JM51-0-0-824-MS	0
3 ธันวาคม 2545	082T-0-0-810-01	120H-0-0-584-Y2	0
6 ธันวาคม 2545	NH01-0-0-586-01	NH07-0-C-561-67	0
11 ธันวาคม 2545	3597-F-9-SM2-09	3818-5-0-586-R6	0
19 ธันวาคม 2545	1Y85-0-0-027-01	1F55-1-0-027-01	0
21 ธันวาคม 2545	3S16-0-0-584-01	FG33-0-0-NW0-01	0
26 ธันวาคม 2545	FG40-0-0-ABS-02	JM10-0-0-PS1-02	0
27 ธันวาคม 2545	2703-F-0-586-W1	2709-F-0-SM1-10	0
		เฉลี่ย	0
		ส่วนเบี่ยงเบนมาตรฐาน	0

ตารางที่ ข-3 กรณีที่ใช้ AW564 แทนที่ก่อน แล้วตามด้วยพลาสติกใหม่

วันที่	จาก	ไป	เวลา(นาที)
24 ตุลาคม 2545	0098-0-0-027-01	TT04-0-0-TTO-36	15
29 พฤศจิกายน 2545	022Z-8-C-20H-02	FG27-0-0-340-02	12
4 กันยายน 2545	0245-5-0-030-01	0085-1-0-080-AH	13
6 ธันวาคม 2545	034N-0-0-PA6-01	2859-F-9-586-02	15
24 ตุลาคม 2545	034P-0-0-PA6-01	HC03-0-0-BWE-02	14
15 พฤศจิกายน 2545	1506-0-0-027-Y2	1C84-0-C-584-46	15
3 ตุลาคม 2545	2709-F-0-586-R3	FG04-0-0-027-02	14
4 พฤศจิกายน 2545	2R30-5-0-518-01	2859-F-9-586-W1	14
10 ตุลาคม 2545	3473-8-0-027-01	4931-0-0-439-02	15
11 ธันวาคม 2545	392S-0-C-PM1-10	266A-0-C-15M-65	13
12 กันยายน 2545	4475-0-0-027-01	1763-5-0-080-01	11
11 พฤศจิกายน 2545	4475-0-0-999-01	NK01-0-0-D07-02	15
25 ตุลาคม 2545	6230-0-0-030-01	NH04-0-0-189-02	12
1 ตุลาคม 2545	FG01-0-0-027-01	13B7-0-0-D16-Y2	11
5 กันยายน 2545	JM04-0-0-030-01	3599-F-9-586-02	16
26 กันยายน 2545	JM62-0-0-920-01	NH22-0-0-027-02	12
11 ธันวาคม 2545	NH07-0-C-561-67	0153-0-0-027-01	10
11 กันยายน 2545	NK03-0-0-PVC-GY	NK04-0-0-D07-R4	14
10 ตุลาคม 2545	NK06-0-0-99-01	SS10-0-0-080-L1	15
21 พฤศจิกายน 2545	SA27-0-0-140-02	SH30-5-9-586-02	13
12 กันยายน 2545	SZ01-0-0-030-02	3923-5-0-SM1-10	15
13 พฤศจิกายน 2545	TA02-0-C-200-W1	JM69-0-0-A01-08	14
14 กันยายน 2545	TY11-0-0-NA2-02	392T-0-C-PM1-10	16
3 ตุลาคม 2545	UZ01-0-C-586-M3	UT02-0-0-LP1-W2	20
10 ตุลาคม 2545	UZ04-0-C-564-02	2P50-0-0-027-02	15
		เฉลี่ย	14
		ส่วนเบี่ยงเบนมาตรฐาน	2.07

ตารางที่ ข-4 กรณีที่ใช้ AW564 ผสมผงเคมี, AW564 และ พลาสติกใหม่ ตามลำดับ

วันที่	จาก	ไป	เวลา(นาที)
21 พฤศจิกายน 2545	0003-2-9-027-02	038X-0-0-D16-01	34
21 กันยายน 2545	0259-0-0-027-02	34U4-1-0-D16-01	24
4 กันยายน 2545	1506-0-0-027-Y2	15M5-5-0-518-01	29
5 กันยายน 2545	1C84-0-C-584-02	0515-2-0-080-01	30
11 ธันวาคม 2545	21S0-8-0-027-02	1C85-0-C-584-46	33
18 กันยายน 2545	2735-0-0-027-02	TP08-0-0-586-GY	26
22 ตุลาคม 2545	340X-0-0-18H-02	33E8-5-0-027-01	24
18 กันยายน 2545	3599-F-9-586-39	TC01-0-0-584-01	28
6 ธันวาคม 2545	392P-0-C-PM1-10	3H82-5-0-518-01	34
24 ตุลาคม 2545	7M35-4-0-019-T1	DK01-0-0-030-01	33
3 ธันวาคม 2545	C522-0-0-584-02	477N-8-0-586-01	26
4 พฤศจิกายน 2545	DK06-1-0-SN1-02	FG35-0-0-080-W2	25
22 พฤศจิกายน 2545	FG26-0-0-340-02	FG23-1-0-LP1-01	27
30 ตุลาคม 2545	FG39-0-0-ABS-02	JM58-0-0-810-01	32
21 พฤศจิกายน 2545	HD02-0-0-561-02	NH16-0-0-564-I2	25
1 ตุลาคม 2545	JM07-0-0-PSO-02	TY05-0-0-027-60	27
27 ธันวาคม 2545	NH23-0-0-027-02	TG08-0-0-564-WH	29
30 พฤศจิกายน 2545	NK02-0-0-D07-02	SS11-0-0-E17-01	34
18 พฤศจิกายน 2545	NU03-0-0-586-02	NU01-0-0-027-01	29
19 ธันวาคม 2545	SA28-0-0-140-02	SY32-0-0-D16-Y2	22
30 ตุลาคม 2545	TC04-A-0-19H-02	NH08-0-0-561-47	22
9 พฤศจิกายน 2545	TC06-0-0-145-02	470S-0-D-586-01	27
19 ตุลาคม 2545	TC11-0-0-145-02	NH16-0-0-564-I2	32
9 พฤศจิกายน 2545	TH01-0-0-030-58	TS01-0-0-395-01	31
13 พฤศจิกายน 2545	TY06-0-0-UBE-02	7M35-4-0-018-D4	27
		เฉลี่ย	28
		ส่วนเบี่ยงเบนมาตรฐาน	3.62

ตารางที่ ข-5 กรณีที่ใช้ AW564, JJ4323 และ พลาสติกใหม่ ตามลำดับ

วันที่	จาก	ไป	เวลา(นาที)
19 กันยายน 2545	0353-0-0-518-01	SA34-0-0-140-02	48
1 ตุลาคม 2545	0655-5-0-999-01	TY07-0-0-UBE-02	55
3 ธันวาคม 2545	1116-5-0-030-01	ST03-0-0-027-40	46
26 กันยายน 2545	1240-0-0-518-01	SS01-1-0-140-02	52
11 กันยายน 2545	17X2-0-C-999-01	HT01-0-0-030-W1	57
4 พฤศจิกายน 2545	2B08-0-C-027-01	042S-0-0-PA6-01	51
22 ตุลาคม 2545	353D-8-0-027-01	SA35-0-0-140-02	50
3 ตุลาคม 2545	3H82-5-0-518-01	DK06-0-0-395-02	54
22 พฤศจิกายน 2545	4485-S-0-518-01	391B-0-C-036-02	38
13 กันยายน 2545	HT01-0-0-030-W1	HD03-0-C-027-01	47
30 พฤศจิกายน 2545	IR01-0-0-030-01	SS07-0-0-518-23	40
22 พฤศจิกายน 2545	NH02-0-0-170-37	TY05-0-0-027-60	41
21 ธันวาคม 2545	NH02-0-0-170-11	NU02-0-0-027-02	57
13 พฤศจิกายน 2545	NH03-0-0-170-45	FG25-0-0-340-02	36
29 พฤศจิกายน 2545	NH04-0-0-189-02	1469-1-0-518-02	44
17 ธันวาคม 2545	NK03-0-0-PVC-GY	TP11-0-0-027-02	55
12 กันยายน 2545	SG01-0-U-294-01	DK06-0-0-395-02	48
21 พฤศจิกายน 2545	TC02-A-0-19H-02	TK17-A-0-030-02	40
19 กันยายน 2545	TC03-A-0-19H-02	TK06-A-0-030-02	39
24 ตุลาคม 2545	TC05-A-0-19H-02	TS02-0-0-030-02	49
19 ตุลาคม 2545	TY03-0-0-UBE-01	TA02-0-C-200-W1	54
17 ธันวาคม 2545	UZ19-0-0-737-02	2B03-0-0-B3S-12	43
18 พฤศจิกายน 2545	UZ21-A-0-737-02	SS15-0-0-518-02	47
26 ธันวาคม 2545	UZ23-0-0-737-02	AT01-0-0-027-D1	39
30 ตุลาคม 2545	UZ28-0-0-737-02	TC04-A-0-19H-02	55
		เฉลี่ย	47
		ส่วนเบี่ยงเบนมาตรฐาน	6.55

ตารางที่ ข-6 กรณีที่ใช้ AW564 ผสมผงเคมี, AW564,JJ4323 และพลาสติกใหม่ ตามลำดับ

วันที่	จาก	ไป	เวลา(นาที)
13 กันยายน 2545	13A6-8-0-030-02	0248-1-0-518-01	69
3 ตุลาคม 2545	1506-0-0-027-Y2	0245-5-0-030-01	75
6 ธันวาคม 2545	1W98-0-0-027-02	1219-1-0-030-01	82
19 กันยายน 2545	24V6-5-0-027-52	TH02-0-0-030-01	70
17 ธันวาคม 2545	2B03-0-0-B3S-02	FG11-0-0-030-01	83
24 ตุลาคม 2545	2D49-0-0-027-33	TH01-0-0-395-01	76
29 พฤศจิกายน 2545	391B-0-C-036-02	022Z-8-C-20H-02	88
26 ธันวาคม 2545	AT02-0-0-027-L2	FG12-0-0-030-01	74
14 กันยายน 2545	DK06-0-0-395-02	40A7-0-0-027-66	79
3 ตุลาคม 2545	DK06-L-0-395-02	0456-1-0-027-BG	80
29 พฤศจิกายน 2545	FG15-0-0-395-01	NH04-0-0-189-02	73
21 ธันวาคม 2545	FG16-0-0-140-02	0153-0-0-027-01	70
22 ตุลาคม 2545	NK03-0-0-PVC-GY	HT01-0-0-030-W1	87
13 พฤศจิกายน 2545	NU05-0-0-027-02	TH02-0-0-030-01	77
17 ธันวาคม 2545	SA15-0-0-140-02	1H10-0-0-027-49	86
11 พฤศจิกายน 2545	SA16-0-0-140-02	4475-0-0-999-01	79
19 กันยายน 2545	SA34-0-0-140-02	TA02-0-C-200-W1	71
22 ตุลาคม 2545	SA35-0-0-140-02	TA04-0-C-200-Y2	74
26 กันยายน 2545	SS01-1-0-140-02	TY09-0-0-027-01	69
10 ตุลาคม 2545	SS16-A-0-395-02	1H10-0-0-027-48	75
1 ตุลาคม 2545	TH01-0-0-030-58	0655-5-0-999-01	81
9 พฤศจิกายน 2545	TK11-A-0-036-02	0245-5-0-518-01	75
3 ธันวาคม 2545	TT04-0-0-TTO-36	TY01-0-0-UBE-02	65
11 พฤศจิกายน 2545	TY07-0-0-UBE-02	0435-5-0-518-01	84
19 ตุลาคม 2545	TY11-0-0-NA2-02	2D49-0-0-027-71	73
		เฉลี่ย	77
		ส่วนเบี่ยงเบนมาตรฐาน	6.23

ภาคผนวก ค
เวลาปรับพารามิเตอร์ของเครื่องฉีดพลาสติก

ตารางที่ ค-1 เวลาปรับพารามิเตอร์ กรณีที่ไม่ต้องติดตั้งอุปกรณ์ร่วม

วันที่	รหัสชิ้นงาน	เวลา(นาที)
4 กันยายน 2545	0085-1-0-080-AH	35
3 ตุลาคม 2545	0245-5-0-030-01	34
13 กันยายน 2545	0248-1-0-518-01	30
3 พฤศจิกายน 2545	042S-0-0-PA6-01	32
11 พฤศจิกายน 2545	0435-5-0-518-01	30
3 ตุลาคม 2545	0456-1-0-027-BG	35
6 ธันวาคม 2545	1219-1-0-030-01	28
1 ตุลาคม 2545	13B7-0-0-D16-Y2	31
4 กันยายน 2545	15M5-5-0-518-01	33
12 กันยายน 2545	1763-5-0-080-01	34
6 ธันวาคม 2545	2859-F-9-586-02	28
3 พฤศจิกายน 2545	2859-F-9-586-W1	34
5 กันยายน 2545	3599-F-9-586-02	28
24 ตุลาคม 2545	HC03-0-0-BWE-02	28
13 กันยายน 2545	HD03-0-C-027-01	33
11 กันยายน 2545	HT01-0-0-030-W1	34
22 ตุลาคม 2545	HT01-0-0-030-W1	28
19 ตุลาคม 2545	NH16-0-0-564-I2	27
21 พฤศจิกายน 2545	NH16-0-0-564-I2	34
22 ตุลาคม 2545	SA35-0-0-140-02	29
19 ธันวาคม 2545	SY32-0-0-D16-Y2	35
19 ตุลาคม 2545	TA02-0-C-200-W1	31
30 ตุลาคม 2545	TC04-A-0-19H-02	34
21 พฤศจิกายน 2545	TK17-A-0-030-02	30
24 ตุลาคม 2545	TS02-0-0-030-02	30
22 พฤศจิกายน 2545	TY05-0-0-027-60	31
	เฉลี่ย	31
	ส่วนเบี่ยงเบนมาตรฐาน	2.67

ตารางที่ ค-2 เวลาปรับพารามิเตอร์ กรณีติดตั้งอุปกรณ์ร่วม

วันที่	รหัสชิ้นงาน	เวลา(นาที)
1 ตุลาคม 2545	0655-5-0-999-01	38
11 ธันวาคม 2545	266A-0-C-15M-65	39
22 พฤศจิกายน 2545	391B-0-C-036-02	50
14 กันยายน 2545	40A7-0-0-027-66	46
11 พฤศจิกายน 2545	4475-0-0-999-01	43
26 ธันวาคม 2545	AT01-0-0-027-D1	49
12 กันยายน 2545	DK06-0-0-395-02	37
3 ตุลาคม 2545	DK06-0-0-395-02	41
17 ธันวาคม 2545	FG11-0-0-030-01	40
26 ธันวาคม 2545	FG12-0-0-030-01	41
13 พฤศจิกายน 2545	FG25-0-0-340-02	37
30 ตุลาคม 2545	JM58-0-0-810-01	48
13 พฤศจิกายน 2545	JM69-0-0-A01-08	49
25 ตุลาคม 2545	NH04-0-0-189-02	44
29 พฤศจิกายน 2545	NH04-0-0-189-02	45
30 ตุลาคม 2545	NH08-0-0-561-47	43
26 กันยายน 2545	NH22-0-0-027-02	45
11 พฤศจิกายน 2545	NK01-0-0-D07-02	43
26 กันยายน 2545	SS01-1-0-140-02	49
30 พฤศจิกายน 2545	SS07-0-0-518-23	39
10 ตุลาคม 2545	SS10-0-0-080-L1	43
30 พฤศจิกายน 2545	SS11-0-0-E17-01	44
27 ธันวาคม 2545	TG08-0-0-564-WH	41
3 ธันวาคม 2545	TY01-0-0-UBE-02	48
1 ตุลาคม 2545	TY05-0-0-027-60	50
26 กันยายน 2545	TY09-0-0-027-01	48
	เฉลี่ย	44
	ส่วนเบี่ยงเบนมาตรฐาน	4.10

ภาคผนวก ง
การทดสอบความถูกต้องในการวิเคราะห์ปัญหา

การทดสอบความถูกต้องของโปรแกรมในการวิเคราะห์ปัญหา

1. ใช้ข้อมูลทดสอบจำนวน 20 รายการ ซึ่งแสดงในตารางที่ ง-1

ตารางที่ ง-1 ข้อมูลใบสั่งสินค้า 20 รายการ

เลขที่ใบสั่งสินค้า	รหัสสินค้า	กำหนดส่งสินค้า	ปริมาณ(ชิ้น)
UN2002/01/002	038X-0-0-D16-01	9-Mar	30,000
UN2002/01/003	266B-0-C-15M-65	10-Mar	30,000
UN2002/01/005	4345-0-0-SM1-10	12-Mar	50,000
UN2002/01/001	4E34-5-0-518-01	12-Mar	60,000
UN2002/01/006	1162-5-0-027-01	15-Mar	70,000
UN2002/01/004	0655-5-0-027-01	15-Mar	150,000
UN2002/01/013	2707-F-9-SM1-10	16-Mar	50,000
UN2002/01/009	392R-0-C-PM1-10	19-Mar	20,000
UN2002/01/015	AS01-0-0-561-01	19-Mar	30,000
UN2002/01/016	0435-5-0-518-01	20-Mar	25,000
UN2002/01/017	392U-0-C-PM1-10	23-Mar	6,000
UN2002/01/011	NK02-0-0-D07-02	23-Mar	200,000
UN2002/01/010	NH09-0-C-561-47	25-Mar	5,000
UN2002/01/008	JM06-0-0-PS0-02	26-Mar	1,500
UN2002/01/020	SS21-0-0-PS0-02	26-Mar	25,000
UN2002/01/014	0003-2-9-027-02	28-Mar	20,000
UN2002/01/021	DK06-L-0-395-02	30-Mar	15,000
UN2002/01/019	317R-8-0-564-01	30-Mar	25,000
UN2002/01/024	045Z-0-0-030-01	1-Apr	50,000
UN2002/01/022	HK01-A-C-PF1-09	9-Apr	200,000

2. เช็คกับ Part Master จะได้รายละเอียด ซึ่งแสดงในตารางที่ ง-2

ตารางที่ ง-2 เช็ค Part Master

รหัสสินค้า	GT	CAV	Ton m/c	Annealing	Robot	Man	Customer Code
0003-2-9-027-02	23	2	30	w			SALT
038X-0-0-D16-01	20.5	8	30				CANNON
0435-5-0-518-01	20	2	55	w			DST
045Z-0-0-030-01	30	8	100		R	M	DST
0655-5-0-027-01	21	4	55	w			SD
1162-5-0-027-01	15.56	8	55	w			TSS
266B-0-C-15M-65	21.5	4	55		R		IMTC
2707-F-9-SM1-10	21	4	55				IMTC
317R-8-0-564-01	36.35	8	150				DASCO
392R-0-C-PM1-10	30.8	4	80		R		SU
392U-0-C-PM1-10	37.5	4	100		R		SU
4345-0-0-SM1-10	19	8	55				SU
4E34-5-0-518-01	23	4	80	w			MS
AS01-0-0-561-01	23.5	4	55				ASI
DK06-L-0-395-02	43.48	4	80		R	M	DIT
HK01-A-C-PF1-09	38	16	150		R		NH
JM06-0-0-PS0-02	34	2	80		R	M	JVCM
NH09-0-C-561-47	35	4	100		R	M	NHK
NK02-0-0-D07-02	24.96	24	150				NSK
SS21-0-0-PS0-02	35.2	4	100		R	M	SSI

3. คำนวณเวลาที่ใช้ในการผลิต ซึ่งแสดงในตารางที่ ง-3

ตารางที่ ง-3 คำนวณเวลาที่ใช้ในการผลิต

รหัสสินค้า	ปริมาณ (ชิ้น)	จำนวน ชิ้น/ชั่วโมง	เวลาที่ใช้ ในการผลิต(ชม.)
0003-2-9-027-02	20,000	282	70.99
038X-0-0-D16-01	30,000	1,264	23.73
0435-5-0-518-01	25,000	324	77.16
045Z-0-0-030-01	50,000	864	57.87
0655-5-0-027-01	150,000	617	243.06
1162-5-0-027-01	70,000	1,666	42.02
266B-0-C-15M-65	30,000	603	49.77
2707-F-9-SM1-10	50,000	617	81.02
317R-8-0-564-01	25,000	713	35.06
392R-0-C-PM1-10	20,000	421	47.53
392U-0-C-PM1-10	6,000	346	17.36
4345-0-0-SM1-10	50,000	1,364	36.65
4E34-5-0-518-01	60,000	563	106.48
AS01-0-0-561-01	30,000	551	54.40
DK06-L-0-395-02	15,000	298	50.32
HK01-A-C-PF1-09	200,000	1,364	146.60
JM06-0-0-PS0-02	1,500	191	7.87
NH09-0-C-561-47	5,000	370	13.50
NK02-0-0-D07-02	200,000	3,115	64.20
SS21-0-0-PS0-02	25,000	368	67.90

4. เรียงลำดับใบสั่งสินค้าโดยพิจารณากำหนดส่งสินค้า ความสำคัญของลูกค้าและเลขที่ใบสั่งสินค้า แสดงในตารางที่ ง-4

ตารางที่ ง-4 ใบสั่งสินค้าที่ได้รับการจัดลำดับความสำคัญแล้ว

ลำดับที่	เลขที่ใบสั่งสินค้า	รหัสสินค้า	กำหนด ส่งสินค้า	ลำดับความสำคัญ ของลูกค้า	ขนาด เครื่องฉีด
1	UN2002/01/002	038X-0-0-D16-01	9-Mar	12	30
2	UN2002/01/003	266B-0-C-15M-65	10-Mar	10	55
3	UN2002/01/005	4345-0-0-SM1-10	12-Mar	4	55
4	UN2002/01/001	4E34-5-0-518-01	12-Mar	23	80
5	UN2002/01/004	0655-5-0-027-01	15-Mar	20	55
6	UN2002/01/006	1162-5-0-027-01	15-Mar	37	55
7	UN2002/01/013	2707-F-9-SM1-10	16-Mar	10	55
8	UN2002/01/009	392R-0-C-PM1-10	19-Mar	4	80
9	UN2002/01/015	AS01-0-0-561-01	19-Mar	17	55
10	UN2002/01/016	0435-5-0-518-01	20-Mar	31	55
11	UN2002/01/011	NK02-0-0-D07-02	23-Mar	3	150
12	UN2002/01/017	392U-0-C-PM1-10	23-Mar	4	100
13	UN2002/01/010	NH09-0-C-561-47	25-Mar	7	100
14	UN2002/01/008	JM06-0-0-PS0-02	26-Mar	6	80
15	UN2002/01/020	SS21-0-0-PS0-02	26-Mar	11	100
16	UN2002/01/014	0003-2-9-027-02	28-Mar	13	30
17	UN2002/01/021	DK06-L-0-395-02	30-Mar	5	80
18	UN2002/01/019	317R-8-0-564-01	30-Mar	27	150
19	UN2002/01/024	045Z-0-0-030-01	1-Apr	31	100
20	UN2002/01/022	HK01-A-C-PF1-09	9-Apr	33	150

5. แยกงานตามกลุ่มเครื่องจักรที่ใช้ในการผลิต ดังแสดงในตารางที่ ง-5

ตารางที่ ง-5 แยกงานตามกลุ่มเครื่องจักรที่ใช้ในการผลิต

ขนาด เครื่องผลิต	ลำดับที่	รหัสสินค้า	กำหนดส่งสินค้า	เวลาที่ใช้ในการผลิต	
				ชั่วโมง	นาที
30	1	038X-0-0-D16-01	9-Mar	23	44
	16	0003-2-9-027-02	28-Mar	71	-
55	2	266B-0-C-15M-65	10-Mar	49	47
	3	4345-0-0-SM1-10	12-Mar	36	40
	5	0655-5-0-027-01	15-Mar	243	4
	6	1162-5-0-027-01	15-Mar	42	2
	7	2707-F-9-SM1-10	16-Mar	81	2
	9	AS01-0-0-561-01	19-Mar	54	24
	10	0435-5-0-518-01	20-Mar	77	10
80	4	4E34-5-0-518-01	12-Mar	106	29
	8	392R-0-C-PM1-10	19-Mar	47	32
	14	JM06-0-0-PS0-02	26-Mar	7	53
	17	DK06-L-0-395-02	30-Mar	50	20
100	12	392U-0-C-PM1-10	23-Mar	17	22
	13	NH09-0-C-561-47	25-Mar	13	31
	15	SS21-0-0-PS0-02	26-Mar	67	55
	19	045Z-0-0-030-01	1-Apr	57	53
150	11	NK02-0-0-D07-02	23-Mar	64	12
	18	317R-8-0-564-01	30-Mar	35	4
	20	HK01-A-C-PF1-09	9-Apr	146	37

6. แบ่งงานให้เครื่องจักรแต่ละเครื่อง

M030-PD01		
038X-0-0-D16-01		
0003-2-9-027-02		
M055-PD01	M055-PD02	M055-PD03
266B-0-C-15M-65	4345-0-0-SM1-10	0655-5-0-027-01
2707-F-9-SM1-10	1162-5-0-027-01	
0435-5-0-518-01	AS01-0-0-561-01	
M080-PD01		M080-PD02
4E34-5-0-518-01		392R-0-C-PM1-10
		JM06-0-0-PS0-02
		DK06-L-0-395-02
M100-PD01	M100-PD02	M100-PD03
392U-0-C-PM1-10	NH09-0-C-561-47	SS21-0-0-PS0-02
	045Z-0-0-030-01	
M150-PD01		M150-PD02
NK02-0-0-D07-02		317R-8-0-564-01
		HK01-A-C-PF1-09

7. หาแผนการผลิตที่เป็นไปได้ทั้งหมด

เครื่องผลิตพลาสติก	จำนวนงาน	จำนวนแผนการผลิตที่เป็นไปได้
M030-PD01	2	2
M055-PD01	3	6
M055-PD02	3	6
M055-PD03	1	1
M080-PD01	1	1
M080-PD02	3	6
M100-PD01	1	1
M100-PD02	2	2
M100-PD03	1	1
M150-PD01	1	1
M150-PD02	2	2

8. กำหนดค่าเริ่มต้นของเครื่องจักร (ยกตัวอย่าง 5 เครื่องจักร)

เครื่องผลิตพลาสติก	งานอันดับสุดท้าย (ก่อนเริ่มวางแผนใหม่)	เริ่มวางแผน	
		วันที่	เวลา
M030-PD01	JM53-0-0-080-02	5-Mar	00.00
M055-PD01	0259-0-0-027-02	5-Mar	00.00
M055-PD02			
M055-PD03			
M080-PD01	TG05-0-0-HP1-02	5-Mar	02.00
M080-PD02			
M100-PD01	TT004-0-0-TTO-36	6-Mar	10.00
M100-PD02			
M100-PD03			
M150-PD01			
M150-PD02	SS22-C-0-PSO-02	5-Mar	00.00

M030-PD01

แผนการผลิต	ผลิตลำดับที่	รหัสสินค้า	เวลาเตรียมเครื่องจักร			เริ่มผลิต		ผลิตเสร็จ		กำหนดส่งสินค้า	วันส่งสินค้า	งานล่าช้า
			เปลี่ยนแม่พิมพ์	ทำความสะอาด	ปรับพารามิเตอร์	วันที่	เวลา	วันที่	เวลา			
แผนที่ 1	1	038X-0-0-D16-01	16	28	31	5-Mar	01.15	6-Mar	00.59	9-Mar	8-Mar	1
	2	0003-2-9-027-02	16	14	31	6-Mar	02.00	9-Mar	01.00	28-Mar	12-Mar	16
แผนที่ 2	1	0003-2-9-027-02	16	14	31	5-Mar	01.01	8-Mar	00.01	28-Mar	11-Mar	17
	2	038X-0-0-D16-01	16	28	31	8-Mar	01.16	9-Mar	01.00	9-Mar	11-Mar	-2

ดังนั้นเลือก แผนการผลิตที่ 1

	แผนที่ 1	แผนที่ 2
ส่งงานล่าช้ารวม	0	2
เวลาเตรียมเครื่องจักรรวม	136	136

M055-PD01

แผนการผลิต	ผลิตลำดับที่	รหัสสินค้า	เวลาเตรียมเครื่องจักร			เริ่มผลิต		ผลิตเสร็จ		กำหนดส่งสินค้า	วันส่งสินค้า	งานล่าช้า
			เปลี่ยนแม่พิมพ์	ทำความสะอาด	ปรับพารามิเตอร์	วันที่	เวลา	วันที่	เวลา			
แผนที่ 1	1	266B-0-C-15M-65	21	28	44	5-Mar	01.33	7-Mar	03.20	10-Mar	9-Mar	1
	2	2707-F-9-SM1-10	21	0	31	7-Mar	04.12	10-Mar	13.14	16-Mar	13-Mar	3
	3	0435-5-0-518-01	21	28	31	10-Mar	14.34	13-Mar	19.44	20-Mar	17-Mar	3
แผนที่ 2	1	266B-0-C-15M-65	21	28	44	5-Mar	01.33	7-Mar	03.20	10-Mar	9-Mar	1
	2	0435-5-0-518-01	21	14	31	7-Mar	04.26	10-Mar	9.36	20-Mar	14-Mar	6
	3	2707-F-9-SM1-10	21	14	31	10-Mar	10.42	13-Mar	19.44	16-Mar	16-Mar	0
แผนที่ 3	1	0435-5-0-518-01	21	28	31	5-Mar	01.20	8-Mar	06.30	20-Mar	11-Mar	9
	2	2707-F-9-SM1-10	21	14	31	8-Mar	07.36	11-Mar	16.38	16-Mar	14-Mar	2
	3	266B-0-C-15M-65	21	14	44	11-Mar	17.57	13-Mar	19.44	10-Mar	16-Mar	-6
แผนที่ 4	1	0435-5-0-518-01	21	28	31	5-Mar	01.20	8-Mar	06.30	20-Mar	11-Mar	9
	2	266B-0-C-15M-65	21	14	44	8-Mar	7.49	10-Mar	09.36	10-Mar	13-Mar	-3
	3	2707-F-9-SM1-10	21	0	31	10-Mar	10.28	13-Mar	19.30	16-Mar	16-Mar	0
แผนที่ 5	1	2707-F-9-SM1-10	21	14	31	5-Mar	01.06	8-Mar	10.08	16-Mar	11-Mar	5
	2	0435-5-0-518-01	21	28	31	8-Mar	11.28	11-Mar	16.38	20-Mar	15-Mar	5
	3	266B-0-C-15M-65	21	14	44	11-Mar	17.57	13-Mar	19.44	10-Mar	16-Mar	-6
แผนที่ 6	1	2707-F-9-SM1-10	21	14	31	5-Mar	01.06	8-Mar	10.08	16-Mar	11-Mar	5
	2	266B-0-C-15M-65	21	14	44	8-Mar	11.27	10-Mar	13.14	10-Mar	13-Mar	-3
	3	0435-5-0-518-01	21	14	31	10-Mar	14.20	13-Mar	19.30	20-Mar	17-Mar	3

ดังนั้น เลือกแผนการผลิตที่ 1 หรือ 2

	แผนที่ 1	แผนที่ 2	แผนที่ 3	แผนที่ 4	แผนที่ 5	แผนที่ 6
ส่งงานล่าช้ารวม	0	0	6	3	6	3
เวลาเตรียมเครื่องจักรรวม	225	225	225	211	225	211

M080-PD02

แผนการผลิต	ผลิตลำดับที่	รหัสสินค้า	เวลาเตรียมเครื่องจักร			เริ่มผลิต		ผลิตเสร็จ		กำหนดส่งสินค้า	รับส่งสินค้า	งานสำเร็จ
			เปลี่ยนแม่พิมพ์	ทำความสะอาด	ปรับพารามิเตอร์	วันที่	เวลา	วันที่	เวลา			
แผนที่ 1	1	392R-0-C-PM1-10	32	14	44	5-Mar	03.30	7-Mar	03.02	19-Mar	9-Mar	10
	2	JM06-0-0-PS0-02	32	0	44	7-Mar	04.18	7-Mar	12.11	26-Mar	9-Mar	17
	3	DK06-L-0-395-02	32	14	44	7-Mar	13.41	9-Mar	16.01	30-Mar	11-Mar	19
แผนที่ 2	1	392R-0-C-PM1-10	32	14	44	5-Mar	03.30	7-Mar	03.02	19-Mar	9-Mar	10
	2	DK06-L-0-395-02	32	14	44	7-Mar	04.32	9-Mar	06.52	30-Mar	10-Mar	20
	3	JM06-0-0-PS0-02	32	14	44	9-Mar	08.22	9-Mar	16.15	26-Mar	11-Mar	15
แผนที่ 3	1	JM06-0-0-PS0-02	32	14	44	5-Mar	03.30	5-Mar	11.23	26-Mar	6-Mar	20
	2	DK06-L-0-395-02	32	14	44	5-Mar	12.53	7-Mar	15.23	30-Mar	9-Mar	21
	3	392R-0-C-PM1-10	32	14	44	7-Mar	16.43	9-Mar	16.15	19-Mar	12-Mar	7
แผนที่ 4	1	JM06-0-0-PS0-02	32	14	44	5-Mar	03.30	5-Mar	11.23	26-Mar	6-Mar	20
	2	392R-0-C-PM1-10	32	0	44	5-Mar	12.39	7-Mar	12.11	19-Mar	10-Mar	9
	3	DK06-L-0-395-02	32	14	44	7-Mar	13.41	9-Mar	16.01	30-Mar	11-Mar	19
แผนที่ 5	1	DK06-L-0-395-02	32	47	44	5-Mar	04.03	7-Mar	06.23	30-Mar	8-Mar	22
	2	392R-0-C-PM1-10	32	14	44	7-Mar	07.53	9-Mar	07.25	19-Mar	11-Mar	8
	3	JM06-0-0-PS0-02	32	0	44	9-Mar	08.41	9-Mar	16.34	26-Mar	11-Mar	15
แผนที่ 6	1	DK06-L-0-395-02	32	47	44	5-Mar	04.03	7-Mar	06.23	30-Mar	8-Mar	22
	2	JM06-0-0-PS0-02	32	14	44	7-Mar	07.53	7-Mar	15.46	26-Mar	9-Mar	17
	3	392R-0-C-PM1-10	32	0	44	7-Mar	17.02	9-Mar	16.34	19-Mar	12-Mar	7

ดังนั้น เลือกแผนการผลิตที่ 1 หรือ 4

	แผนที่ 1	แผนที่ 2	แผนที่ 3	แผนที่ 4	แผนที่ 5	แผนที่ 6
ส่งงานสำเร็จรวม	0	0	0	0	0	0
เวลาเตรียมเครื่องจักรรวม	256	270	270	256	289	289

M100-PDO1

แผนการผลิต	ผลิตลำดับที่	รหัสสินค้า	เวลาเตรียมเครื่องจักร			เริ่มผลิต		ผลิตเสร็จ		กำหนดส่งสินค้า	วันส่งสินค้า	งานส่ง
			เปลี่ยนแม่พิมพ์	ทำความสะอาด	ปรับพารามิเตอร์	วันที่	เวลา	วันที่	เวลา			
แผนที่ 1	1	NH09-0-C-561-47	37	28	44	6-Mar	11.49	7-Mar	01.20	25-Mar	8-Mar	17
	2	045Z-0-0-030-01	37	14	44	7-Mar	02.55	9-Mar	12.48	1-Apr	11-Mar	21
แผนที่ 2	1	045Z-0-0-030-01	37	77	44	6-Mar	12.38	8-Mar	22.31	1-Apr	10-Mar	22
	2	NH09-0-C-561-47	37	14	44	9-Mar	00.06	9-Mar	13.37	25-Mar	11-Mar	14

ดังนั้น เลือกแผนการผลิตที่ 1

	แผนที่ 1	แผนที่ 2
ส่งงานล่าช้ารวม	0	0
เวลาเตรียมเครื่องจักรรวม	204	253

M150-PDO2

แผนการผลิต	ผลิตลำดับที่	รหัสสินค้า	เวลาเตรียมเครื่องจักร			เริ่มผลิต		ผลิตเสร็จ		กำหนดส่งสินค้า	วันส่งสินค้า	งานล่วง
			เปลี่ยนแม่พิมพ์	ทำความสะอาด	ปรับพารามิเตอร์	วันที่	เวลา	วันที่	เวลา			
แผนที่ 1	1	317R-8-0-564-01	42	28	31	5-Mar	01.41	6-Mar	12.45	30-Mar	6-Mar	24
	2	HK01-A-C-PF1-09	42	0	44	6-Mar	14.11	12-Mar	16.48	9-Apr	12-Mar	28
แผนที่ 2	1	HK01-A-C-PF1-09	42	0	44	5-Mar	01.26	11-Mar	04.03	9-Apr	11-Mar	29
	2	317R-8-0-564-01	42	28	31	11-Mar	05.44	12-Mar	16.48	30-Mar	12-Mar	18

ดังนั้น เลือกแผนการผลิตที่ 1 หรือ 2

	แผนที่ 1	แผนที่ 2
ส่งงานล่าช้ารวม	0	0
เวลาเตรียมเครื่องจักรรวม	187	187

ภาคผนวก จ
เวลาส่งงานล่าช้า

ตารางที่ จ-1 เวลาส่งงานล่าช้าของแผนการผลิตที่ออกโดยผู้วางแผน เดือนกุมภาพันธ์

รหัสสินค้า	กำหนดส่งสินค้า	วันส่งสินค้า	จำนวนวันส่งงานล่าช้า
2709-F-0-586-R3	15-Feb-03	18-Feb-03	3
13B6-0-0-D16-33	19-Feb-03	22-Feb-03	3
2037-0-C-027-02	1-Mar-03	5-Mar-03	4
0245-5-0-027-01	3-Mar-03	5-Mar-03	2
0655-5-0-027-01	15-Feb-03	16-Feb-03	1
0435-5-0-518-01	20-Feb-03	22-Feb-03	2
4347-0-0-SM1-10	1-Mar-03	8-Mar-03	7
392R-0-C-PM1-10	19-Feb-03	22-Feb-03	3
392K-0-C-PM1-10	5-Mar-03	8-Mar-03	2
JM06-0-0-PS0-02	26-Feb-03	28-Feb-03	2
TC08-A-0-19H-02	10-Feb-03	12-Feb-03	2
SA21-0-0-586-01	12-Feb-03	14-Feb-03	2
TC05-A-0-19H-02	14-Feb-03	19-Feb-03	5
392S-0-C-PM1-10	10-Feb-03	14-Feb-03	4
HK01-A-C-PF1-09	8-Feb-03	11-Feb-03	3
		รวม	45

ตารางที่ จ-2 เวลาส่งงานล่าช้าของแผนการผลิตที่ออกโดยผู้วางแผน เดือนมีนาคม

รหัสสินค้า	กำหนดส่งสินค้า	วันส่งสินค้า	จำนวนวันส่งงานล่าช้า
FG42-0-0-586-01	9-Mar-03	11-Mar-03	2
SA09-0-0-586-01	26-Mar-03	29-Mar-03	3
TP11-0-0-027-02	11-Mar-03	14-Mar-03	3
2735-0-0-027-02	16-Mar-03	18-Mar-03	2
FG07-0-0-586-02	18-Mar-03	21-Mar-03	3
FG03-0-U-586-02	18-Mar-03	21-Mar-03	3
4347-0-0-999-02	16-Mar-03	22-Mar-03	6
DK06-L-0-395-02	10-Mar-03	13-Mar-03	3
392M-0-C-PM1-10	20-Mar-03	26-Mar-03	6
FG06-0-0-586-02	18-Mar-03	23-Mar-03	5
TC05-A-0-19H-02	28-Mar-03	31-Mar-03	3
045Z-0-0-030-01	3-Apr-03	9-Apr-03	6
TC04-A-0-19H-002	18-Mar-03	22-Mar-03	4
TC04-A-0-19H-02	4-Apr-03	10-Apr-03	6
SS30-1-0-PS0-02	2-Apr-03	6-Apr-03	4
HK01-A-C-999-09	19-Mar-03	26-Mar-03	7
		รวม	66

ตารางที่ จ-3 เวลาส่งงานล่าช้าของแผนการผลิตที่ออกโดยโปรแกรม เดือนกุมภาพันธ์

รหัสสินค้า	กำหนดส่งสินค้า	วันส่งสินค้า	จำนวนวันส่งงานล่าช้า
4347-0-0-SM1-10	1-Mar-03	12-Mar-03	11
0655-5-0-027-01	15-Feb-03	22-Feb-03	7
41F6-8-C-D07-02	28-Feb-03	4-Mar-03	4
SA21-0-0-586-01	12-Feb-03	13-Feb-03	1
TC05-A-0-19H-02	14-Feb-03	16-Feb-03	2
TC04-A-0-19H-02	10-Feb-03	12-Feb-03	2
TP10-0-0-080-02	12-Feb-03	13-Feb-03	1
3599-F-9-586-02	7-Feb-03	8-Feb-03	1
392S-0-C-PM1-10	10-Feb-03	9-Feb-03	1
HK01-A-C-PF1-09	8-Feb-03	12-Feb-03	4
SS22-C-0-PS0-02	28-Feb-03	2-Mar-03	2
		รวม	36

ตารางที่ จ-4 เวลาส่งงานล่าช้าของแผนการผลิตที่ออกโดยโปรแกรม เดือนมีนาคม

รหัสสินค้า	กำหนดส่งสินค้า	วันส่งสินค้า	จำนวนวันส่งงานล่าช้า
SA09-0-0-586-01	26-Mar-03	29-Mar-03	3
UZ18-0-0-564-W3	28-Mar-03	30-Mar-03	2
TP11-0-0-027-02	10-Mar-03	13-Mar-03	3
4347-0-0-999-02	19-Mar-03	25-Mar-03	6
NH06-1-0-561-44	20-Mar-03	27-Mar-03	7
1763-5-0-080-01	27-Mar-03	29-Mar-03	2
33E8-5-0-027-01	27-Mar-03	31-Mar-03	2
TC08-A-0-19H-02	28-Mar-03	8-Apr-03	8
TC04-A-0-19H-B5	4-Apr-03	11-Apr-03	7
HK01-A-C-999-09	19-Mar-03	25-Mar-03	6
		รวม	46

ภาคผนวก จ
แผนการผลิตเดือนกุมภาพันธ์และเดือนมีนาคม

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (MO30-PDO1)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
038X-0-0-D16-01	60,000	2/3/03 9:02	2/5/03 8:29	47	-
JM53-0-0-080-02	30,000	2/5/03 9:29	2/7/03 7:52	60	-
JM49 0-0-PS1-02	8,000	2/7/03 8:52	2/8/03 9:33	60	-
3473-8-0-027-01	40,000	2/8/03 10:48	2/10/03 3:51	75	-
0003-2-9-027-02	20,000	2/10/03 4:38	2/13/03 3:37	47	-
SA09-0-0-586-01	50,000	2/13/03 4:52	2/18/03 8:19	75	-
AT06-0-0-030-01	40,000	2/18/03 9:33	2/21/03 17:48	74	-
1416-0-0-030-01	65,000	2/21/03 18:35	2/23/03 13:13	47	-
TS01-0-0-395-01	50,000	2/23/03 14:00	2/24/03 23:45	47	-
1C84-0-C-584-46	40,000	2/25/03 0:46	2/26/03 9:38	61	-
Total				593	0

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M055-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
266B-0-C-15M-65	50,000	2/1/03 18:11	2/5/03 5:08	65	-
1162-5-0-027-01	70,000	2/5/03 6:14	2/7/03 0:15	66	-
UZ04-0-C-564-02	70,000	2/7/03 1:34	2/12/03 17:22	79	-
13B6-0-0-D16-33	20,000	2/12/03 18:27	2/13/03 10:47	65	-
0435-5-0-518-01	25,000	2/13/03 12:07	2/16/03 17:17	80	-
UZ18-0-0-564-W3	160,000	2/16/03 18:36	2/20/03 13:57	79	-
4347-0-0-SM1-10	480,000	2/20/03 14:49	3/10/03 9:29	52	11
Total				486	11

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M055-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
23H5-0-0-586-02	60,000	2/6/03 1:09	2/7/03 21:36	52	-
2709-F-0-586-R3	6,500	2/7/03 22:28	2/8/03 4:12	52	-
0655-5-0-027-01	150,000	2/8/03 5:32	2/18/03 8:35	80	7
NH07-0-C-561-64	3,000	2/18/03 9:54	2/18/03 19:10	79	-
NH07-0-C-561-67	4,000	2/18/03 20:15	2/19/03 8:36	65	-
JM26-0-0-LP1-02	20,000	2/19/03 9:41	2/21/03 17:14	65	-
0353-0-0-518-01	80,000	2/21/03 18:34	2/23/03 9:40	80	-
41F6-8-C-D07-02	400,000	2/23/03 10:46	3/1/03 13:22	66	4
8Y32-0-0-D16-Y2	3,800	3/1/03 14:28	3/2/03 12:27	66	-
SA36-0-0-172-AII	40,000	3/2/03 13:33	3/3/03 8:04	66	-
UZ04-0-D-564-B6	12,000	3/3/03 9:09	3/4/03 5:31	65	-
0245-5-0-030-01	20,000	3/4/03 6:51	3/5/03 0:36	80	-
UZ04-0-D-564-Y6	12,000	3/5/03 1:55	3/5/03 22:17	79	-
Total				895	11

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M055-PD03)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
4345-0-0-SM1-10	50,000	2/3/03 22:38	2/5/03 11:17	66	-
2707-F-9-SM1-10	50,000	2/5/03 12:09	2/8/03 21:10	52	-
UZ04-0-C-564-73	70,000	2/8/03 22:15	2/13/03 21:05	65	-
AS01-0-0-561-01	30,000	2/13/03 22:25	2/16/03 4:49	80	-
NH07-0-C-561-47	5,000	2/16/03 5:54	2/16/03 21:20	65	-
TP13-0-0-500-02	12,000	2/16/03 22:12	2/17/03 19:30	52	-
2P50-0-0-027-02	70,000	2/17/03 20:36	2/19/03 15:49	66	-
0353-0-0-027-01	80,000	2/19/03 17:09	2/21/03 8:15	80	-
2037-0-C-027-02	60,000	2/21/03 9:20	2/22/03 21:36	65	-
UZ04-0-D-564-73	30,000	2/22/03 23:09	2/25/03 2:05	93	-
4347-0-0-586-39	6,000	2/25/03 2:57	2/25/03 7:59	52	-
0245-5-0-027-01	20,000	2/25/03 9:19	2/26/03 3:04	80	-
UZ04-0-D-564-B4	10,000	2/26/03 4:23	2/26/03 21:22	79	-
1Z04-0-0-561-02	30,000	2/26/03 22:14	2/28/03 10:11	52	-
UZ04-0-D-564-B7	12,000	2/28/03 11:16	3/1/03 7:38	65	-
0655-5-0-027-01	150,000	3/1/03 8:58	3/8/03 3:00	80	-
Total				1092	0

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M080-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
4E34-5-0-518-01	60,000	2/5/03 12:50	2/9/03 23:19	91	-
392P-0-C-PM1-10	50,000	2/10/03 0:49	2/14/03 23:39	90	-
JM06-0-0-PS0-02	1,500	2/15/03 0:55	2/15/03 8:47	76	-
4475-0-0-027-01	60,000	2/15/03 10:18	2/20/03 6:02	91	-
392K-0-C-PM1-10	30,000	2/20/03 7:32	2/21/03 18:01	90	-
392M-0-C-PM1-10	55,000	2/21/03 19:17	2/23/03 17:49	76	-
22Y4-0-0-080-20	70,000	2/23/03 18:52	2/25/03 1:55	63	-
034P-0-0-PA6-01	26,000	2/25/03 3:26	2/27/03 8:36	91	-
Total				668	0

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M080-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
DK06-L-0-395-02	32,000	2/6/03 1:51	2/10/03 13:12	90	-
392R-0-C-PM1-10	50,000	2/10/03 14:42	2/12/03 14:14	90	-
392M-0-C-561-W1	55,000	2/12/03 15:30	2/16/03 4:48	76	-
DK06-L-0-SN1-02	32,000	2/16/03 6:18	2/20/03 17:39	90	-
NK01-0-0-D07-02	100,000	2/20/03 18:56	2/22/03 2:59	77	-
HA01-0-0-D16-01	130,000	2/22/03 4:43	2/25/03 4:41	104	-
SS27-0-0-PS0-02	12,000	2/25/03 5:44	2/26/03 15:43	63	-
22Y4-0-0-D16-01	60,000	2/26/03 17:14	2/27/03 19:51	91	-
22Y4-0-0-643-02	50,000	2/27/03 20:54	2/28/03 19:05	63	-
Total				744	0

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M100-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
TC04-A-0-19H-02	6,000	2/5/03 14:31	2/7/03 22:04	82	-
SA21-0-0-586-01	25,000	2/7/03 23:40	2/10/03 13:24	96	1
TC05-A-0-19H-02	6,000	2/10/03 14:46	2/12/03 22:19	82	2
392T-0-C-PM1-10	25,000	2/12/03 23:54	2/15/03 21:21	95	-
392U-0-C-PM1-10	25,000	2/15/03 22:42	2/18/03 23:02	81	-
3925-5-0-586-39	6,000	2/19/03 0:10	2/19/03 5:38	68	-
2703-F-0-586-W1	8,500	2/19/03 7:14	2/19/03 14:07	96	-
15M5-5-0-518-01	60,000	2/19/03 15:29	2/22/03 12:56	82	-
AS02-0-0-561-B1	60,000	2/22/03 14:18	2/27/03 21:37	82	-
2703-F-0-SM2-09	10,000	2/27/03 22:45	2/28/03 6:51	68	-
Total				832	3

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M100-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
TC05-A-0-19H-02	6,000	2/7/03 13:22	2/9/03 20:55	82	-
TP10-0-0-080-02	10,000	2/9/03 22:17	2/11/03 1:19	82	1
TC04-A-0-19H-02	6,000	2/11/03 2:41	2/11/03 12:05	82	2
SS04-0-0-080-02	40,000	2/11/03 13:27	2/14/03 8:55	82	-
3599-F-9-586-39	2,600	2/14/03 10:03	2/14/03 20:05	68	-
045Z-0-0-030-01	100,000	2/14/03 21:54	2/18/03 9:50	109	-
SK02-0-0-027-01	15,000	2/18/03 11:45	2/19/03 7:26	115	-
TY02-0-0-UBE-02	2,000	2/19/03 9:21	2/20/03 8:48	115	-
TY01-0-0-UBE-01	2,000	2/20/03 10:24	2/20/03 16:34	96	-
NH08-0-0-561-47	3,000	2/20/03 18:09	2/21/03 1:47	95	-
NH08-0-0-561-64	3,000	2/21/03 3:08	2/21/03 10:46	81	-
NH08-0-0-561-67	3,000	2/21/03 12:07	2/21/03 22:24	81	-
120H-0-0-584-Y2	20,000	2/21/03 23:45	2/23/03 6:37	81	-
3925-5-0-SM1-10	35,000	2/23/03 7:45	2/25/03 0:16	68	-
3599-F-9-SM1-10	8,000	2/25/03 1:24	2/26/03 4:34	68	-
TT04-0-0-TT3-T4	4,000	2/26/03 5:56	2/26/03 16:59	82	-
TT04-0-0-TT4-D9	4,000	2/26/03 18:21	2/27/03 5:24	82	-
Total				1469	3

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M100-PD03)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
3599-F-9-586-02	10,000	2/4/03 21:41	2/6/03 7:38	68	1
392S-0-C-PM1-10	2,500	2/6/03 8:59	2/6/03 16:30	81	1
NH09-0-C-561-64	3,000	2/6/03 18:05	2/7/03 2:11	95	-
NH09-0-C-561-67	3,000	2/7/03 3:32	2/7/03 11:38	81	-
SZ01-0-0-030-02	100,000	2/7/03 13:00	2/10/03 10:04	82	-
TC02-A-0-19H-02	5,000	2/10/03 11:59	2/12/03 8:44	115	-
FG01-0-0-027-01	160,000	2/12/03 10:20	2/19/03 15:10	96	-
477N-8-0-586-01	60,000	2/19/03 16:32	2/22/03 12:12	82	-
DK04-0-0-541-01	50,000	2/22/03 13:34	2/25/03 13:44	82	-
24Z5-1-0-080-02	60,000	2/25/03 20:06	3/1/03 12:04	82	-
Total				864	2

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M150-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
HK01-A-C-PF1-09	200,000	1/31/03 3:45	2/9/03 12:49	87	4
317R-8-0-564-01	300,000	2/9/03 14:31	2/24/03 5:07	102	-
SS30-1-0-PS0-02	15,000	2/24/03 6:21	2/28/03 2:57	74	-
Total				263	4

แผนการผลิตเดือนกุมภาพันธ์ จากโปรแกรมวางแผนการผลิต (M150-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
HK02-0-C-584-W1	200,000	2/1/03 6:34	2/19/03 14:36	87	-
NK02-0-0-D07-02	200,000	2/19/03 15:50	2/22/03 8:02	74	-
SS22-C-0-PS0-02	12,000	2/22/03 9:29	2/28/03 2:30	87	2
Total				248	2

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M030-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
1W98-0-0-027-02	50,000	2/26/03 10:39	2/28/03 6:03	61	-
JM72-0-0-080-02	6,200	2/28/03 7:17	3/1/03 7:12	74	-
FG42-0-0-586-01	6,000	3/1/03 8:27	3/3/03 8:36	75	-
038X-0-0-D16-01	75,000	3/3/03 9:23	3/5/03 20:42	47	-
FG42-0-0-586-01	6,000	3/5/03 21:29	3/7/03 21:38	47	-
0003-2-9-027-02	15,000	3/7/03 22:39	3/10/03 3:53	61	-
SA09-0-0-586-01	150,000	3/10/03 5:08	3/25/03 15:30	75	3
340X-0-0-18H-02	5,000	3/25/03 16:44	3/26/03 21:17	74	-
1C84-0-C-584-46	30,000	3/26/03 22:32	3/27/03 23:11	75	-
1C85-0-C-584-46	30,000	3/27/03 23:58	3/28/03 22:32	47	-
			Total	636	3

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M055-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
1259-0-0-027-02	20,000	3/10/03 10:35	3/11/03 15:54	66	-
23H5-0-0-586-02	80,000	3/11/03 17:00	3/14/03 4:16	66	-
FG07-0-0-586-02	6,000	3/14/03 5:08	3/14/03 22:44	52	-
FG09-0-0-586-02	6,000	3/14/03 23:49	3/15/03 22:02	65	-
FG03-0-U-586-02	4,000	3/15/03 22:54	3/16/03 18:39	52	-
DK05-0-0-541-01	45,000	3/16/03 19:59	3/19/03 17:26	80	-
JM26-0-0-LP1-02	25,000	3/19/03 18:45	3/22/03 16:12	79	-
4A68-1-0-586-01	30,000	3/22/03 17:32	3/24/03 9:49	80	-
NH01-0-0-586-01	10,000	3/24/03 10:41	3/25/03 3:40	52	-
UZ18-0-0-564-W3	133,600	3/25/03 4:45	3/28/03 9:02	65	2
CH01-0-0-561-01	6,000	3/28/03 10:08	3/28/03 21:15	66	-
UZ04-0-D-564-Y5	5,000	3/28/03 22:20	3/29/03 6:49	65	-
27G7-0-0-027-02	70,000	3/29/03 7:55	3/31/03 11:14	66	-
8Y32-0-0-D16-Y2	3,500	3/31/03 12:34	4/1/03 8:49	80	-
2P50-0-0-027-02	130,000	4/1/03 7:55	4/4/03 18:10	66	-
Total				1000	2

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M055-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
TP11-0-0-027-02	36,000	3/5/03 23:23	3/10/03 8:56	66	3
FG08-0-0-586-02	6,000	3/10/03 10:02	3/11/03 11:02	66	-
4A69-1-0-586-01	20,000	3/11/03 12:22	3/12/03 13:51	80	-
UZ03-0-D-564-41	35,000	3/12/03 20:43	3/15/03 16:14	52	-
40A8-0-0-027-02	8,000	3/15/03 17:33	3/16/03 21:57	79	-
266B-0-C-15M-65	30,000	3/16/03 23:30	3/19/03 1:16	93	-
1Y85-0-0-027-01	40,000	3/19/03 2:22	3/22/03 13:42	66	-
2037-0-C-027-02	60,000	3/22/03 14:47	3/24/03 3:03	65	-
1F55-1-0-027-01	40,000	3/24/03 4:23	3/25/03 17:25	80	-
TK01-0-0-027-02	20,000	3/25/03 18:17	3/27/03 5:47	52	-
127T-0-0-N12-01	12,000	3/27/03 7:07	3/28/03 4:25	80	-
Total				779	3

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M055-PD03)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
TP08-0-0-586-GY	8,000	3/8/03 4:06	3/9/03 4:47	66	-
2735-0-0-027-02	60,000	3/9/03 5:53	3/11/03 8:49	66	-
4347-0-0-999-02	230,000	3/11/03 9:41	3/23/03 14:01	52	6
FG04-0-0-027-02	30,000	3/23/03 14:53	3/25/03 10:52	52	-
2530-0-C-027-02	60,000	3/25/03 11:44	3/26/03 20:55	52	-
1F55-1-0-518-01	40,000	3/26/03 22:15	3/28/03 11:17	80	-
IG01-0-0-586-01	100,000	3/28/03 12:23	4/1/03 18:37	66	-
SK04-0-0-027-01	20,000	4/1/03 19:43	4/3/03 4:22	66	-
			Total	500	6

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M080-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
392K-0-C-PM1-10	90,000	2/27/03 10:06	3/3/03 17:34	90	-
1763-5-0-080-01	15,000	3/3/03 19:05	3/4/03 7:55	91	-
NH05-1-0-561-44	20,000	3/4/03 8:58	3/7/03 23:23	63	-
392M-0-C-PM1-10	85,000	3/8/03 0:39	3/13/03 21:44	76	-
DK06-L-0-395-02	36,000	3/13/03 23:14	3/20/03 9:24	90	-
NH06-1-0-561-44	10,000	3/20/03 10:55	3/24/03 3:35	91	7
392R-0-C-PM1-10	45,000	3/24/03 4:51	3/28/03 15:48	76	-
2859-F-9-586-39	18,000	3/28/03 16:51	3/30/03 7:44	63	-
TY04-0-0-UBE-01	2,500	3/30/03 9:15	3/30/03 16:40	91	-
034N-0-0-PA6-01	26,000	3/30/03 17:43	4/2/03 2:54	63	-
Total				794	7

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M080-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
FG06-0-0-586-02	160,000	2/28/03 20:08	3/25/03 12:44	63	-
1763-5-0-080-01	15,000	3/25/03 14:15	3/27/03 4:19	91	2
33E8-5-0-027-01	30,000	3/27/03 5:36	3/27/03 16:01	77	2
1763-6-0-080-02	15,000	3/27/03 17:18	3/29/03 7:22	77	-
2859-F-9-SM1-10	10,000	3/29/03 8:25	3/30/03 6:01	63	-
Total				371	4

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M100-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
TP10-0-0-080-02	30,000	2/28/03 7:59	3/3/03 17:06	68	-
TC02-A-0-19H-02	12,000	3/3/03 18:28	3/8/03 5:52	82	-
392T-0-C-PM1-10	20,000	3/8/03 7:27	3/10/03 15:00	95	-
FG01-0-0-027-01	100,000	3/10/03 16:36	3/15/03 4:37	96	-
045Z-0-0-030-01	300,000	3/15/03 6:45	3/29/03 17:58	128	-
Total				469	0

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M100-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
NH19-0-0-564-D8	4,500	2/27/03 6:46	3/1/03 3:08	82	-
NH16-0-0-564-I2	3,000	3/1/03 4:16	3/2/03 14:18	68	-
TC04-A-0-19H-02	5,000	3/2/03 15:40	3/7/03 8:38	82	-
TC03-A-0-19H-02	12,000	3/7/03 9:46	3/14/03 1:06	68	-
TC05-A-0-19H-02	55,000	3/14/03 2:14	4/5/03 15:55	68	8
Total				368	8

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M100-PD03)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
TC01-0-0-584-01	40,000	3/1/03 13:40	3/2/03 22:29	96	-
NH16-0-0-564-D8	8,500	3/2/03 23:37	3/7/03 0:02	68	-
NH19-0-0-564-I2	5,300	3/7/03 1:10	3/9/03 5:26	68	-
392U-0-C-PM1-10	10,000	3/9/03 6:47	3/10/03 11:43	81	-
NH11-0-0-561-01	2,000	3/10/03 13:19	3/11/03 14:00	96	-
266A-0-C-15M-65	20,000	3/11/03 15:21	3/14/03 10:29	81	-
2703-F-0-586-Y4	2,200	3/14/03 11:37	3/14/03 13:24	68	-
3S48-8-0-D17-R7	150,000	3/14/03 14:32	3/18/03 21:07	68	-
TC04-A-0-19H-B5	60,000	3/18/03 22:29	4/8/03 8:39	82	7
Total				708	7

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M150-PD01)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
317R-8-0-564-01	60,000	2/28/03 4:39	3/3/03 16:48	102	-
TT09-0-0-561-01	8,000	3/3/03 18:15	3/6/03 1:48	87	-
NK06-0-0-171-01	100,000	3/6/03 3:15	3/19/03 6:04	87	-
SS30-1-0-PS0-02	35,000	3/19/03 7:18	3/28/03 7:21	74	-
Total				350	0

แผนการผลิตเดือนมีนาคม จากโปรแกรมวางแผนการผลิต (M150-PD02)

Product Code	Quantity (Pcs.)	Start Date	Finish Date	Setup Time (Min.)	Lateness (Day)
HK01-A-C-999-09	800,000	2/28/03 4:11	3/23/03 6:25	101	6
SS22-D-0-PS0-02	10,000	3/23/03 8:06	3/26/03 8:56	101	-
SS22-B-0-PS0-02	18,000	3/26/03 10:23	3/31/03 15:56	87	-
Total				289	6

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M030-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time (Min)				Start Date	Finished Date	Delivery Date	Leadness (Day)
			Change Mold	Clean M/C	Parameter	Total				
JM58-0-0-810-01							2/3/03 8:15			
036X-0-0-D16-01	60,000	10-Feb-03	16	0	31	47	2/3/03 9:02	2/5/03 8:29	07-Feb-03	
JM53-0-0-060-02	30,000	15-Feb-03	16	0	44	60	2/5/03 9:29	2/7/03 7:52	08-Feb-03	
JM49-0-0-PS1-02	6,000	15-Feb-03	16	0	44	60	2/7/03 8:52	2/8/03 9:33	09-Feb-03	
3473-8-0-027-01	40,000	25-Feb-03	16	28	31	75	2/8/03 10:48	2/10/03 3:51	13-Feb-03	
0003-2-9-027-02	20,000	28-Feb-03	16	0	31	47	2/10/03 4:38	2/13/03 3:37	16-Feb-03	
SA09-0-0-586-01	50,000	28-Feb-03	16	28	31	75	2/13/03 4:52	2/18/03 8:19	20-Feb-03	
AT06-0-0-030-01	40,000	28-Feb-03	16	14	44	74	2/18/03 9:33	2/21/03 17:48	24-Feb-03	
1416-0-0-030-01	65,000	3-Mar-03	16	0	31	47	2/21/03 18:35	2/23/03 13:15	26-Feb-03	
TS01-0-0-395-01	50,000	3-Mar-03	16	0	31	47	2/23/03 14:00	2/24/03 23:45	27-Feb-03	
1C84-0-C-564-46	40,000	11-Mar-03	16	14	31	61	2/25/03 0:46	2/26/03 9:36	28-Feb-03	
					Total	593			Total	0

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M080-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
HC04-0-0-019-T1							1/30/03 11:19			
4E34-5-0-518-01	60,000	12-Feb-03	32	47	31	110	1/30/03 13:09	2/6/03 6:03	19-Feb-03	
392P-0-C-PM1-10	50,000	19-Feb-03	32	14	44	90	2/8/03 7:33	2/12/03 19:54	15-Feb-03	
392R-0-C-PM1-10	50,000	19-Feb-03	32	0	44	76	2/12/03 21:10	2/19/03 13:23	22-Feb-03	3
4475-0-0-027-01	60,000	28-Feb-03	32	47	31	110	2/19/03 15:13	2/24/03 10:57	27-Feb-03	
HA01-0-0-D16-01	130,000	5-Mar-03	32	14	31	77	2/24/03 12:14	2/27/03 12:12	1-Mar-03	
392M-0-C-561-W1	55,000	5-Mar-03	32	28	31	91	2/27/03 13:43	3/3/03 3:01	5-Mar-03	
392K-0-C-PM1-10	30,000	5-Mar-03	32	14	31	77	3/3/03 4:18	3/4/03 14:47	6-Mar-03	2
SS27-0-0-PS0-02	12,000	10-Mar-03	32	14	31	77	3/4/03 16:04	3/6/03 2:03	9-Mar-03	
						Total	708		Total	5

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M080-PD02)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
HA01-0-0-D16-01							2/5/03 0:21			
DK06-L-0-SN1-02	32,000	15-Feb-03	32	14	44	90	2/5/03 1:51	2/10/03 19:24	13-Feb-03	
DK06-L-0-395-02	32,000	23-Feb-03	32	14	31	77	2/10/03 20:41	2/17/03 8:07	21-Feb-03	
NK01-0-0-D07-02	100,000	25-Feb-03	32	14	31	77	2/17/03 9:24	2/21/03 3:45	24-Feb-03	
392M-0-C-PM1-10	55,000	26-Feb-03	32	14	44	90	2/21/03 5:15	2/24/03 16:46	25-Feb-03	
JM06-0-0-PS0-02	1,500	26-Feb-03	32	14	31	77	2/24/03 16:05	2/25/03 1:57	26-Feb-03	2
22Y4-0-0-060-20	70,000	10-Mar-03	32	28	44	104	2/25/03 3:41	2/26/03 10:44	1-Mar-03	
22Y4-0-0-643-02	50,000	10-Mar-03	32	14	31	77	2/26/03 12:01	2/27/03 14:36	2-Mar-03	
22Y4-0-0-D16-01	60,000	10-Mar-03	32	47	31	110	2/27/03 16:28	2/28/03 19:05	2-Mar-03	
034P-0-0-PA6-01	26,000	15-Mar-03	32	28	31	91	2/28/03 20:36	3/3/03 1:46	6-Mar-03	
						Total	793		Total	2

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M100-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time (Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
266D-0-0-15M-65							2/6/03 13:09			
599-F-9-586-02	10,000	10-Feb-03	37	14	31	52	2/6/03 14:31	2/8/03 0:26	9-Feb-03	-
392U-0-C-PM1-10	25,000	10-Feb-03	37	14	31	82	2/6/03 1:50	2/9/03 20:02	10-Feb-03	-
SA21-0-0-586-01	25,000	12-Feb-03	37	47	31	115	2/9/03 21:57	2/12/03 11:41	14-Feb-03	2
NH09-0-C-561-67	3,000	25-Feb-03	37	14	44	95	2/12/03 13:16	2/12/03 21:22	15-Feb-03	-
SZ01-0-0-030-02	100,000	26-Feb-03	37	14	31	82	2/12/03 22:44	2/15/03 19:46	19-Feb-03	-
TC02-A-0-19H-02	5,000	27-Feb-03	37	14	31	82	2/15/03 21:10	2/17/03 17:55	20-Feb-03	-
3599-F-9-586-39	2,600	1-Mar-03	37	28	31	96	2/17/03 19:31	2/18/03 5:33	22-Feb-03	-
3925-5-0-586-39	6,000	3-Mar-03	37	0	44	81	2/16/03 6:54	2/16/03 12:22	22-Feb-03	-
NH08-0-0-561-47	3,000	7-Mar-03	37	14	44	95	2/16/03 13:57	2/16/03 21:35	22-Feb-03	-
NH08-0-0-561-64	3,000	7-Mar-03	37	14	31	82	2/16/03 22:57	2/19/03 6:35	23-Feb-03	-
NH08-0-0-561-67	3,000	7-Mar-03	37	14	31	82	2/15/03 7:57	2/19/03 16:14	23-Feb-03	-
AS02-0-0-561-B1	60,000	8-Mar-03	37	28	31	96	2/15/03 19:50	2/25/03 3:09	26-Feb-03	-
						Total	1,070		Total	2

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M100-PD02)

Product Code	Quantity (Pcs)	Due Date	Set up Time (Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
3925-0-C-PM1-10							2/7/03 12:00			
TC04-A-0-19H-02	6,000	10-Feb-03	37	14	44	95	2/7/03 13:35	2/6/03 7:22	9-Feb-03	-
TC08-A-0-19H-02	6,000	10-Feb-03	37	0	44	81	2/6/03 6:43	2/10/03 16:16	12-Feb-03	2
TC05-A-0-19H-02	6,000	14-Feb-03	37	0	31	68	2/10/03 17:24	2/15/03 11:54	19-Feb-03	5
SS04-0-0-080-02	40,000	26-Feb-03	37	14	31	82	2/15/03 13:16	2/18/03 8:44	21-Feb-03	-
22T-0-C-PM1-10	25,000	1-Mar-03	37	14	31	82	2/16/03 10:06	2/21/03 7:33	24-Feb-03	-
FG01-0-0-027-01	160,000	1-Mar-03	37	47	31	115	2/21/03 9:26	2/25/03 13:06	26-Feb-03	-
045Z-0-0-030-01	100,000	5-Mar-03	37	14	31	82	2/25/03 14:30	3/2/03 2:26	4-Mar-03	-
DK04-0-0-541-01	50,000	10-Mar-03	37	14	44	95	3/2/03 4:01	3/5/03 9:11	9-Mar-03	-
TT04-0-0-TT3-T4	4,000	22-Mar-03	37	14	44	95	3/5/03 10:46	3/5/03 21:49	10-Mar-03	-
TT04-0-0-TT4-D9	4,000	22-Mar-03	37	28	31	96	3/5/03 23:25	3/6/03 10:26	10-Mar-03	-
						Total	691		Total	7

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M100-PD03)

Product Code	Quantity (Pcs)	Due Date	Set up Time (Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
IM87-0-0-PSO-02							2/4/03 20:33			
2425-1-0-080-02	60,000	8-Feb-03	37	14	31	82	2/4/03 21:55	2/7/03 0:01	6-Feb-03	-
TP10-0-0-080-02	10,000	9-Feb-03	37	14	31	82	2/7/03 1:23	2/6/03 4:25	9-Feb-03	-
3925-0-C-PM1-10	10,000	10-Feb-03	37	14	31	82	2/6/03 5:47	2/12/03 14:49	14-Feb-03	4
NH09-0-C-561-64	3,000	25-Feb-03	37	28	31	96	2/12/03 16:25	2/13/03 0:31	16-Feb-03	-
TC05-A-0-19H-02	6,000	27-Feb-03	37	28	31	96	2/13/03 2:07	2/15/03 9:40	20-Feb-03	-
SK02-0-0-027-01	15,000	4-Mar-03	37	47	44	128	2/15/03 11:46	2/16/03 7:29	19-Feb-03	-
15M5-5-0-518-01	60,000	4-Mar-03	37	14	31	82	2/16/03 8:51	2/19/03 6:18	22-Feb-03	-
2703-F-0-586-W1	8,500	4-Mar-03	37	28	44	109	2/19/03 8:07	2/19/03 15:00	23-Feb-03	-
477N-8-0-586-01	60,000	5-Mar-03	37	28	31	96	2/19/03 16:36	2/22/03 12:16	25-Feb-03	-
TY01-0-0-UBE-01	2,000	6-Mar-03	37	14	31	82	2/22/03 13:38	2/22/03 19:48	25-Feb-03	-
TY02-0-0-UBE-02	2,000	6-Mar-03	37	0	31	68	2/22/03 20:56	2/23/03 20:23	25-Feb-03	-
120H-0-0-584-Y2	20,000	10-Mar-03	37	14	31	82	2/23/03 21:45	2/25/03 4:37	26-Feb-03	-
3925-5-0-SM1-10	35,000	15-Mar-03	37	28	44	109	2/25/03 6:26	2/26/03 22:57	1-Mar-03	-
3599-F-9-SM1-10	8,000	17-Mar-03	37	0	31	66	2/27/03 0:05	2/26/03 3:15	2-Mar-03	-
2703-F-0-SM2-09	10,000	17-Mar-03	37	14	31	82	2/26/03 4:37	2/28/03 12:43	2-Mar-03	-
						Total	1,160		Total	4

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M150-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Leadness (Day)
			Change Mold	Clean M/C	Parameter	Total				
SS22-B-0-PS0-02							1/31/03 2:18			
NK02-0-0-D07-02	200,000	24-Feb-03	42	14	31	87	1/31/03 3:45	2/2/03 19:57	5-Feb-03	
HK02-0-C-584-W	200,000	28-Feb-03	42	47	44	133	2/2/03 22:10	2/21/03 5:59	24-Feb-03	
SS30-1-0-PS0-02	15,000	3-Mar-03	42	28	44	114	2/21/03 7:53	2/25/03 4:29	1-Mar-03	
						Total	334		Total	0

แผนการผลิตเดือนกุมภาพันธ์ จากผู้วางแผนการผลิต (M150-PD02)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Leadness (Day)
			Change Mold	Clean M/C	Parameter	Total				
TK02-0-0-080-02							2/1/03 5:20			
HK01-A-C-PF1-09	200,000	8-Feb-03	42	14	44	100	2/1/03 7:00	2/7/03 9:36	11-Feb-03	3
317R-6-0-564-01	300,000	28-Feb-03	42	47	31	120	2/7/03 11:36	2/22/03 2:12	25-Feb-03	
SS22-C-0-PS0-02	12,000	28-Feb-03	42	14	44	100	2/22/03 3:52	2/25/03 19:16	26-Feb-03	
						Total	320		Total	3

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M030-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time (Min)				Start Date	Finished Date	Delivery Date	Lead time (Day)
			Change Mold	Clean M/C	Parameter	Total				
1C84-0-C-584-46							2/26/03 9:38			
1W98-0-0-027-02	50,000	05-Mar-03	16	14	31	61	2/26/03 10:39	2/28/03 6:03	03-Mar-03	
1M72-0-0-080-02	6,200	07-Mar-03	15	14	44	74	2/28/03 7:17	3/4/03 20:17	06-Mar-03	
FG42-0-0-586-02	6,000	09-Mar-03	16	14	31	61	3/4/03 21:18	3/9/03 9:48	11-Mar-03	
FG42-0-0-586-01	6,000	14-Mar-03	16	47	31	94	3/9/03 11:22	3/11/03 11:31	13-Mar-03	
038X-0-0-D16-01	75,000	17-Mar-03	16	14	31	61	3/11/03 12:32	3/13/03 23:51	15-Mar-03	
SA09-0-0-586-01	150,000	26-Mar-03	16	14	31	61	3/14/03 0:52	3/26/03 19:42	29-Mar-03	
0003-2-9-027-02	15,000	01-Apr-03	16	14	44	74	3/26/03 20:56	3/29/03 2:10	31-Mar-03	
340X-0-0-18H-02	5,000	02-Apr-03	16	28	44	88	3/29/03 3:38	3/30/03 8:11	02-Apr-03	
1C85-0-C-584-46	30,000	10-Apr-03	16	28	31	75	3/30/03 9:26	3/31/03 8:00	03-Apr-03	
1C84-0-C-584-46	30,000	10-Apr-03	16	0	31	47	3/31/03 8:47	4/1/03 9:26	04-Apr-03	
					Total	696			Total	5

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M055-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
4347-0-0-SM1-10							3/10/03 9:29			
TP11-0-0-027-02	36,000	11-Mar-03	21	0	31	52	3/10/03 10:21	3/11/03 14:44	14-Mar-03	3
1259-0-0-027-02	20,000	15-Mar-03	21	14	31	66	3/11/03 15:50	3/12/03 1:09	14-Mar-03	
2735-0-0-027-02	60,000	16-Mar-03	21	0	44	65	3/12/03 2:14	3/14/03 5:10	16-Mar-03	2
DK05-0-0-541-01	45,000	20-Mar-03	21	47	44	112	3/14/03 7:02	3/17/03 4:29	20-Mar-03	
40A8-0-0-027-02	8,000	23-Mar-03	21	14	31	65	3/17/03 5:35	3/18/03 9:59	21-Mar-03	-
FG04-0-0-027-02	30,000	25-Mar-03	21	0	31	52	3/18/03 10:51	3/20/03 6:50	23-Mar-03	-
1Y85-0-0-027-01	40,000	26-Mar-03	21	47	31	99	3/20/03 6:29	3/23/03 19:49	25-Mar-03	-
1F55-1-0-027-01	40,000	03-Apr-03	21	0	44	65	3/23/03 20:54	3/25/03 9:56	28-Mar-03	-
SK04-0-0-027-01	20,000	08-Apr-03	21	0	31	52	3/25/03 15:48	3/26/03 19:27	30-Mar-03	
127T-0-0-N12-01	12,000	10-Apr-03	21	14	31	66	3/26/03 20:33	3/27/03 17:51	31-Mar-03	-
2P50-0-0-027-02	130,000	30-Apr-03	21	14	31	65	3/27/03 15:57	3/31/03 3:12	03-Apr-03	-
					Total	761			Total	5

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M055-PD02)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
UZ04-0-D-564-Y5							3/5/03 22:17			
TP08-0-0-566-GY	6,000	11-Mar-03	21	28	31	60	3/5/03 23:37	3/7/03 0:16	10-Mar-03	-
23H5-0-0-586-02	80,000	16-Mar-03	21	14	31	66	3/7/03 1:24	3/11/03 9:34	14-Mar-03	
FG07-0-0-566-02	6,000	18-Mar-03	21	0	44	65	3/11/03 10:39	3/17/03 0:09	21-Mar-03	3
FG03-0-U-566-02	4,000	18-Mar-03	21	0	44	65	3/17/03 1:14	3/17/03 20:59	21-Mar-03	3
FG09-0-0-586-02	6,000	19-Mar-03	21	0	31	52	3/17/03 21:51	3/18/03 8:41	19-Mar-03	-
FG08-0-0-566-02	6,000	19-Mar-03	21	0	31	52	3/18/03 9:33	3/18/03 21:16	19-Mar-03	
4A69-1-0-566-01	20,000	22-Mar-03	21	47	31	99	3/18/03 22:55	3/20/03 6:24	21-Mar-03	-
1Z03-0-D-564-41	35,000	24-Mar-03	21	14	31	66	3/20/03 7:30	3/22/03 1:10	24-Mar-03	
266B-0-C-15M-65	30,000	25-Mar-03	21	14	44	79	3/22/03 2:29	3/23/03 11:35	25-Mar-03	
4A68-1-0-566-01	30,000	27-Mar-03	21	26	31	60	3/23/03 12:55	3/25/03 5:12	27-Mar-03	-
NH01-0-0-566-01	10,000	27-Mar-03	21	0	44	65	3/25/03 5:17	3/25/03 23:16	27-Mar-03	
2037-0-C-027-02	60,000	01-Apr-03	21	14	31	66	3/26/03 0:22	3/27/03 12:38	30-Mar-03	-
TK01-0-0-027-02	20,000	04-Apr-03	21	0	31	52	3/27/03 13:30	3/29/03 1:00	02-Apr-03	
2TG7-0-0-027-02	70,000	04-Apr-03	21	0	44	65	3/29/03 2:55	3/31/03 5:24	03-Apr-03	
UZ04-0-D-564-Y5	5,000	04-Apr-03	21	47	44	112	3/31/03 7:16	3/31/03 15:45	03-Apr-03	
6Y32-0-0-D16-Y2	3,500	05-Apr-03	21	14	31	66	3/31/03 15:51	4/1/03 13:06	03-Apr-03	-
					Total	1,130			Total	6

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M055-PD03)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
0655-5-0-027-01							3/8/03 3:00			
2530-0-C-027-02	60,000	12-Mar-03	21	14	31	66	3/8/03 4:06	3/10/03 14:26	12-Mar-03	-
4347-0-0-999-02	230,000	16-Mar-03	21	14	44	79	3/10/03 15:45	3/19/03 3:50	22-Mar-03	6
JM26-0-0-LP1-02	25,000	23-Mar-03	21	14	31	66	3/19/03 4:56	3/22/03 2:23	23-Mar-03	
UZ18-0-0-564-W3	133,600	28-Mar-03	21	47	31	99	3/22/03 4:02	3/25/03 8:19	27-Mar-03	-
1F55-1-0-518-01	40,000	30-Mar-03	21	28	44	93	3/25/03 9:52	3/26/03 22:54	30-Mar-03	-
IG01-0-0-586-01	100,000	02-Apr-03	21	14	44	79	3/27/03 0:13	3/31/03 6:27	02-Apr-03	
CH01-0-0-561-01	6,000	02-Apr-03	21	14	31	66	3/31/03 7:33	3/31/03 18:40	02-Apr-03	
					Total	546			Total	6

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M080-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
034P-0-0-PA6-01							2/27/03 8:36			
DK06-L-0-395-02	36,000	10-Mar-03	32	14	31	77	2/27/03 9:07	3/10/03 0:27	13-Mar-03	3
1763-5-0-080-01	15,000	18-Mar-03	32	47	44	123	3/10/03 1:11	3/10/03 14:01	13-Mar-03	-
NH06-1-0-561-44	10,000	20-Mar-03	32	28	31	91	3/10/03 14:32	3/12/03 9:45	15-Mar-03	-
NH05-1-0-561-44	20,000	20-Mar-03	32	0	31	63	3/12/03 10:16	3/16/03 0:41	19-Mar-03	-
392K-0-C-PM1-10	90,000	20-Mar-03	32	14	44	90	3/16/03 1:25	3/18/03 20:56	20-Mar-03	-
392M-0-C-PM1-1	85,000	20-Mar-03	32	0	31	63	3/18/03 21:27	3/23/03 1:52	26-Mar-03	6
392R-0-C-PM1-10	45,000	30-Mar-03	32	0	31	63	3/23/03 2:23	3/27/03 13:20	29-Mar-03	-
2859-F-9-586-39	18,000	02-Apr-03	32	28	44	104	3/27/03 14:04	3/29/03 4:57	01-Apr-03	-
034N-0-0-PA6-01	26,000	16-Apr-03	32	28	31	91	3/29/03 5:28	3/31/03 14:39	03-Apr-03	-
						Total	765		Total	9

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M080-PD02)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
22Y4-0-0-643-02							2/28/03 19:05			
FG06-0-0-585-02	160,000	18-Mar-03	32	14	31	77	2/28/03 20:22	3/21/03 23:12	23-Mar-03	5
1763-5-0-080-01	15,000	02-Apr-03	32	47	31	110	3/22/03 1:02	3/23/03 15:06	26-Mar-03	-
33E6-5-0-027-01	30,000	04-Apr-03	32	14	44	90	3/23/03 16:36	3/24/03 3:01	27-Mar-03	-
TY04-0-0-UBE-01	2,500	09-Apr-03	32	14	44	90	3/24/03 4:31	3/24/03 11:56	28-Mar-03	-
2859-F-9-SM1-10	10,000	17-Apr-03	32	14	31	77	3/24/03 13:13	3/25/03 10:49	30-Mar-03	-
1763-6-0-060-02	15,000	18-Apr-03	32	14	31	77	3/25/03 12:06	3/25/03 13:23	31-Mar-03	-
						Total	521		Total	5

แผนการผลิตเดือนมีนาคม จากฝ่ายแผนการผลิต (M100-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
2703-F-0-SM2-09							2/28/03 6:51			
TP10-0-0-080-02	30,000	11-Mar-03	37	14	31	82	2/28/03 8:13	3/3/03 17:20	06-Mar-03	
392U-0-C-PM1-10	10,000	20-Mar-03	37	14	31	82	3/3/03 18:42	3/4/03 23:38	07-Mar-03	
392T-0-C-PM1-10	20,000	20-Mar-03	37	0	44	81	3/5/03 0:59	3/7/03 8:32	11-Mar-03	
2703-F-0-586-Y4	2,200	24-Mar-03	37	28	31	96	3/7/03 10:08	3/7/03 11:55	10-Mar-03	
TC05-A-0-19H-02	55,000	28-Mar-03	37	14	31	82	3/7/03 13:17	3/28/03 16:33	31-Mar-03	3
3546-6-0-D17-R7	150,000	02-Apr-03	37	26	31	96	3/26/03 20:09	3/31/03 16:19	02-Apr-03	
						Total	519		Total	3

แผนการผลิตเดือนมีนาคม จากฝ่ายแผนการผลิต (M100-PD02)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
TT04-0-0-TT4-D9							2/27/03 5:24			
NH16-0-0-564-12	3,000	17-Mar-03	37	14	31	82	2/27/03 6:46	2/28/03 16:46	02-Mar-03	
NH19-0-0-564-12	5,300	17-Mar-03	37	0	31	66	2/28/03 17:56	3/2/03 22:12	05-Mar-03	
NH19-0-0-564-D8	4,500	17-Mar-03	0	14	44	58	3/2/03 23:10	3/4/03 19:32	07-Mar-03	
NH16-0-0-564-D.	8,500	17-Mar-03	37	0	31	68	3/4/03 20:40	3/6/03 21:05	10-Mar-03	
266A-0-C-15M-65	20,000	25-Mar-03	37	14	31	82	3/6/03 22:27	3/11/03 17:35	14-Mar-03	
NH11-0-0-561-01	2,000	30-Mar-03	37	47	31	115	3/11/03 19:30	3/12/03 20:11	15-Mar-03	
FG01-0-0-027-01	100,000	30-Mar-03	37	14	44	95	3/12/03 21:46	3/17/03 9:47	20-Mar-03	
045Z-0-0-030-01	300,000	03-Apr-03	37	14	31	82	3/17/03 11:09	4/7/03 9:39	09-Apr-03	6
						Total	650		Total	6

แผนการผลิตเดือนมีนาคม จากฝ่ายแผนการผลิต (M100-PD03)

Product Code	Quantity (Pcs)	Due Date	Set up Time(Min)				Start Date	Finished Date	Delivery Date	Lateness (Day)
			Change Mold	Clean M/C	Parameter	Total				
24Z5-1-0-080-02							3/1/03 12:04			
TC01-0-0-584-01	40,000	18-Mar-03	37	47	44	126	3/1/03 14:12	3/2/03 23:01	05-Mar-03	
TC02-A-0-19H-02	12,000	18-Mar-03	37	14	44	95	3/3/03 0:36	3/7/03 12:00	10-Mar-03	
TC03-A-0-19H-02	12,000	18-Mar-03	37	0	31	66	3/7/03 13:08	3/12/03 0:32	15-Mar-03	
TC04-A-0-19H-02	5,000	18-Mar-03	37	0	31	66	3/12/03 1:40	3/19/03 5:23	22-Mar-03	4
TC04-A-0-19H-B5	60,000	04-Apr-03	37	28	44	109	3/19/03 7:12	4/7/03 6:02	10-Apr-03	6
						Total	468		Total	10

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M150-PD01)

Product Code	Quantity (Pcs)	Due Date	Set up Time (Min)				Start Date	Finished Date	Delivery Date	Leadness (Day)
			Change Mold	Clean M/C	Parameter	Total				
SS30-1-0-PS0-02							2/28/03 2:57			
317R-8-0-564-01	60,000	11-Mar-03	42	47	31	120	2/28/03 4:57	3/3/03 17:06	07-Mar-03	
1T09-0-0-561-01	8,000	23-Mar-03	42	14	44	100	3/3/03 18:46	3/6/03 2:18	09-Mar-03	
NK06-0-0-171-01	100,000	25-Mar-03	42	14	44	100	3/6/03 3:59	3/19/03 6:46	22-Mar-03	
SS30-1-0-PS0-02	35,000	02-Apr-03	42	14	44	100	3/19/03 8:28	4/4/03 15:16	06-Apr-03	4
						Total			Total	4

แผนการผลิตเดือนมีนาคม จากผู้วางแผนการผลิต (M150-PD02)

Product Code	Quantity (Pcs)	Due Date	Set up Time (Min)				Start Date	Finished Date	Delivery Date	Leadness (Day)
			Change Mold	Clean M/C	Parameter	Total				
SS22-C-0-PS0-02							2/28/03 2:30			
HK01-A-C-999-09	800,000	19-Mar-03	42	14	44	100	2/28/03 4:10	3/24/03 14:35	26-Mar-03	7
SS22-D-0-PS0-02	10,000	30-Mar-03	42	14	44	100	3/24/03 16:15	3/27/03 17:05	29-Mar-03	
SS22-B-0-PS0-02	18,000	05-Apr-03	42	0	44	66	3/27/03 18:31	4/2/03 0:04	04-Apr-03	
						Total			Total	7

ภาคผนวก ช
ชุดคำสั่งโปรแกรม

```

frmLUVProduct - 1
Option Explicit
Private rs As ADODB.Recordset
Private bCancel As Boolean
Public openner As String
Private tranno As Double
Private strtranno As String
Private strBasicSQL As String
Private strCriteriaSQL As String
Private strOrderSQL As String
Private o
Private Sub cmdOK_Click()
Dim oProduct As cProduct
If openner = "Product" Then
If TypeOf o Is cProduct Then
Set oProduct = o
frmPlanner.txtProduct.Text = If(IsNull(oProduct.prod_code), "",oProduct.prod_code)
Set frmPlanner.oProduct = o
frmLUVProduct.Hide
End If
ElseIf openner = "mProduct" Then
Set oProduct = o
frmStartTime.txtProd.Text = If(IsNull(oProduct.prod_code), "",oProduct.prod_code)
Set frmStartTime.oProduct = o
frmLUVProduct.Hide
ElseIf openner = "Machine" Then
Dim oMachine As cMachine
If TypeOf o Is cMachine Then
Set oMachine = o
frmStartTime.txtMachinecode.Text = If(IsNull(oMachine.machine_code), "",oMachine.machine_code)
Set frmStartTime.o
Machine = o
frmLUVProduct.Hide
End If
End If
end sub
Private Sub Command1_Click()
'lookupLastest
If (openner = "Product") Or (openner = "mProduct") Then
searchproduct
ElseIf openner = "Machine" Then
SearchMachine
End If
end sub
Private Sub ListView1_ItemClick(ByVal Item As MSCOMCTLLib.ListItem)
If TypeOf Item.Tag Is cProduct Then
Set o = Item.Tag
ElseIf TypeOf Item.Tag Is cMachine Then
Set o = Item.Tag
End If
end sub
Private Sub searchproduct()
If Not (rs Is Nothing) Then
rs.Close
Set rs = Nothing

```

```

End If
Set rs = New ADODB.Recordset
Dim lstmp As ListItem
Dim oProduct As cProduct
strBasicSQL = "select a.prod_code, a.prod_name, a.cav," & _
' a.ct, a.machine_size, a.annealing," & _
' a.robot, a.man, a.cust_code," & _
' b.cust_priority ' & _
' from product a, customer b ' & _
' where a.cust_code = b.cust_code '
If Len(txtName.Text) > 0 Then
strBasicSQL = strBasicSQL & " and prod_code like '%" & txtName.Text & '%" or prod_name like '%"
frmLUVProduct - 2 & txtName.Text & '%"
End If
strBasicSQL = strBasicSQL & "order by prod_code" rs.Open strBasicSQL, db
ListView1.ListItems.Clear
While Not rs.EOF
Set lstmp = ListView1.ListItems.Add(, CStr(rs.Fields!prod_code))
With lstmp
.Text = rs.Fields!prod_code
.SubItems(1) = IIf(IsNull(rs.Fields!prod_name), "", rs.Fields!prod_name)
End With
Set oProduct = New cProduct
oProduct.Initialize rs.Fields!prod_code, IIf(IsNull(rs.Fields!prod_name), "", rs.Fields!prod_name)
oProduct.cav = rs.Fields!cav
oProduct.ct = rs.Fields!ct
oProduct.machine_size = rs.Fields!machine_size
oProduct.annealing = IIf(IsNull(rs.Fields!annealing), "", rs.Fields!annealing)
oProduct.robot = IIf(IsNull(rs.Fields!robot), "", rs.Fields!robot)
oProduct.man = IIf(IsNull(rs.Fields!man), "", rs.Fields!man)
oProduct.cust_code = IIf(IsNull(rs.Fields!cust_code), "", rs.Fields!cust_code)
Set lstmp.Tag = oProduct rs.MoveNext
Wend
end sub

```

```

Private Sub SearchMachine()
If Not (rs Is Nothing) Then rs.Close
Set rs = Nothing
End If
Set rs = New ADODB.Recordset
Dim lstmp As ListItem
Dim oMachine As cMachine
strBasicSQL = "select machine_no, machine_name, machine_size " & _ " from Tmachine "
If Len(txtName.Text) > 0 Then
strBasicSQL = strBasicSQL & " where machine_no like '%" & txtName.Text & '%" or
machine_name like '%" & txtName.Text & '%"
End If
strBasicSQL = strBasicSQL & "order by machine_no" rs.Open strBasicSQL, db
ListView1.ListItems.Clear
While Not rs.EOF
Set lstmp = ListView1.ListItems.Add(, CStr(rs.Fields!machine_no))
With lstmp
.Text = rs.Fields!machine_no

```

```

.SubItems(1) = IIf(IsNull(rs.Fields!machine_name), "", rs.Fields!machine_name)
End With
Set oMachine = New cMachine
oMachine.SetValue rs Set lstmp.Tag = oMachine rs.MoveNext
Wend
end sub

```

```

frmMachine - 1
Option Explicit
frmMain - 1
Private Sub mMachine_Click()
Unload Me
frmStartTime.Show
end sub

```

```

Private Sub mnuOrder_Click()
Unload Me
frmPlanner.Show
end sub

```

```

frmMold - 1
Option Explicit
frmPlanner - 1
Option Explicit
Private rs As ADODB.Recordset
Private bCancel As Boolean
Private tranno As Double
Private strtranno As String
Private strBasicSQL As String
Private strCriteriaSQL As String
Private strOrderSQL As String
Public oProduct As cProduct
Public colTransact As Collection
Public colold As Collection
Public vjob As Collection
Const MODULENAME = "frmPlanner"
Private Enum FormState
fmNEW
fmSAVED
fmEDIT
fmSEARCH
End Enum
Private Enum colData
colMin = -1
colID
colTreatment
colDuration
colColor
colStatus
colMax
End Enum
Dim fmState As FormState
Dim bDirtyForm As Boolean
' History
' 2002-03-10 Created

```

```

Function isDirtyStatus() As Boolean isDirtyStatus = bDirtyForm
end function

' History
' 2002-03-10 Created Sub setDirtyStatus(ByVal newDirtyForm As Boolean) bDirtyForm = newDirtyForm
' lblDirty.Caption = IIf(newDirtyForm, "**", "")
end sub

' 2002-01-28 Created
'

Private Function getModelFormState() As FormState
getModelFormState = fmState
end function

' History
' 200X-XX-XX Created
'

Private Function getNameModelFormState(ByVal fm As FormState) As String
Dim result As String
Select Case fm
Case fmSEARCH
result = "SEARCH"
Case fmSAVED
result = "SAVED"
Case fmEDIT
result = "EDIT"
Case fmNEW
result = "NEW"
Case Else
result = "UNKNOWN"
End Select
getNameModelFormState = result
end function

' History
' 200X-XX-XX Created
frmPlanner - 2
'

Private Sub setModelFormState(ByVal fm As FormState)
Const METHODNAME = "setModelFormState" On Error GoTo ErrHandler
fmState = fm
ExitMethod:
Exit Sub
ErrorHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end sub

Private Sub setVisualizeFormState(ByVal fm As FormState)
'Const METHODNAME = "setVisualizeFormState"
'On Error GoTo ErrHandler
'
' Select Case fm
' Case fmNEW
' lockGrid True

```

```

' Case fmSAVED
' lockGrid True
' Case fmEDIT
' lockGrid False
' Case fmSEARCH
' lockGrid True
' Case Else
' ' nothing
' End Select
ExitMethod:
Exit Sub
ErrorHandler:
' ShowError MODULENAME, METHODNAME
Resume ExitMethod
end sub

Sub setupGrid(ByVal basic As String, ByVal criteria As String, ByVal orderby As String)
Const METHODNAME = "setupGrid"
Dim sql As String
sql = basic
If Len(criteria) > 0 Then
sql = sql + " and " + criteria
End If
sql = sql + " " + orderby
' logger.debugLog MODULENAME, METHODNAME, "sql: " & sql
If Not (rs Is Nothing) Then rs.Close
Set rs = Nothing
End If
Set rs = New ADODB.Recordset rs.Open sql, db, adOpenDynamic, adLockOptimistic
' Set TDBGGrid1.datasource = rs TDBGGrid1.ReBindtranno = -1bCancel = False
end sub

' Desc
' list last tranno
' History
Function lookupLastest() As Long strBasicSQL = "select tranno, orderno,product_code, qty,duedate " & _
"from transact " & _"order by tranno "
frmPlanner - 3
strCriteriaSQL = ""
strOrderSQL = ""
setupGrid strBasicSQL, strCriteriaSQL, strOrderSQL
Me.Show
' prepareReturn lookupLastest = tranno
end function

Private Sub btnGen_Click()
Dim i As Integer
deletetmp
deletemachinetime
deletemachinejob
Dim o As New cOrder
For i = 1 To ListView1.ListItems.Count
Set o = ListView1.ListItems.Item(i).Tag
insertToTmp o
Next i

```

```
findSolution
```

```
end sub
```

```
Private Sub btnLoad_Click()
```

```
'Load data
```

```
Dim oord As cOrder
```

```
Dim ctxord As cOrderContext
```

```
Dim vData As New Collection
```

```
Set ctxord = New cOrderContext
```

```
Dim i As Integer
```

```
Dim phr As Double
```

```
Dim pt As Double
```

```
Dim lsttmp As ListItem
```

```
Set vData = ctxord.GetData("select a.orderno, a.prod_code, a.duedate, a.qty,
```

```
b.cav, b.ct from torderdata a, product b where a.prod_code = b.prod_code")
```

```
If vData.Count > 0 Then
```

```
ListView1.ListItems.Clear
```

```
For i = 1 To vData.Count
```

```
Set oord = vData.Item(i)
```

```
Set lsttmp = ListView1.ListItems.Add(, , CStr(oord.orderno))
```

```
With lsttmp
```

```
phr = (oord.cav * 0.9) / oord.ct
```

```
pt = oord.Quality / phr
```

```
.SubItems(1) = oord.productcode
```

```
.SubItems(2) = oord.Quality
```

```
.SubItems(3) = IIf(IsNull(oord.duedate), "", oord.duedate)
```

```
.SubItems(4) = phr
```

```
.SubItems(5) = pt
```

```
End With
```

```
Set lsttmp.Tag = oord
```

```
Next i
```

```
End If
```

```
end sub
```

```
Private Sub btnSave_Click()
```

```
'doSave
```

```
savetolist
```

```
end sub
```

```
Private Sub Command1_Click() 'lookupLastest
```

```
If Not (rs Is Nothing) Then rs.Close
```

```
Set rs = Nothing
```

```
End If
```

```
Set rs = New ADODB.Recordset
```

```
Dim lsttmp As ListItem
```

```
Dim oord As cOrder
```

```
Dim phr As Double
```

```
Dim pt As Double
```

```
Dim ctxCorder As cOrderContext
```

```
Set ctxCorder = New cOrderContext
```

```
Dim colorder As New Collection
```

```
frmPlanner - 4
```

```
strBasicSQL = "select a.tranno, a.orderno, a.product_code, a.qty, a.duedate,b.cav, b.ct " & _
```

```
"from transact a, product b " & _ "where a.prod_code = b.prod_code " & _ "order by tranno "
```

```
Set colorder = ctxCorder.GetData(strBasicSQL)
```



```

Dim i As Integer
ListView1.ListItems.Clear
For i = 1 To colorder.Count
Set lsttmp = ListView1.ListItems.Add(, CStr(rs.Fields!orderno))
With lsttmp
Set oord = New cOrder
Set oord = colorder.Item(i)
plr = (oord.cav * 0.9) / oord.ct
pt = oord.Quality / plr
.SubItems(1) = oord.productcode
.SubItems(2) = oord.Quality
.SubItems(3) = IIf(IsNull(oord.duedate), "", oord.duedate)
.SubItems(4) = plr
.SubItems(5) = pt
End With
' Set oOrd = New cOrder
' oOrd.Initialize CStr(rs.Fields!orderno), rs.Fields!product_code, rs.Fields!qty, IIf(IsNull(rs.Fields!duedate), 0, rs.Fields!duedate)
' oOrd.cav = oProduct.cav
' oOrd.ct = oProduct.ct
Set lsttmp.Tag = oord
Next i
end sub

Private Sub doSave()
'Save information
Const METHODNAME = "doSave"
On Error GoTo ErrHandler
Dim iRow As Long
Dim strSQL As String
Dim strStatus As String
Dim parainstert(4) As Variant
Dim paraUpdate(4) As Variant
Dim cmd As ADODB.Command
Dim lAffected As Long
Dim ctxSystabcontext As SysTabContext
Set ctxSystabcontext = New SysTabContext
Dim strlasttranno As String
Dim lasttranno As Double
' force any changed made to the current row
' to be saved to data source, or XArray
'TDGrid1.Update
' Find last tranno
If tranno = 0 Then
strlasttranno = ctxSystabcontext.GetKeyValue("tranno")
tranno = CDb1(strlasttranno)
tranno = tranno + 1
insertValue
Else
updateValue
End If
ExitMethod:
Exit Sub
ErrorHandler:
ShowError MODULENAME, METHODNAME
Resume

```

```

logger fetalLog MODULENAME, METHODNAME, "sql=[" & strSQL & "]"
logger fetalLog MODULENAME, METHODNAME, "paraInsert=[" & showArray(paraInsert) & "]"
logger fetalLog MODULENAME, METHODNAME, "paraUpdate=[" & showArray(paraUpdate) & "]"
Resume ExitMethod
end sub

frmPlanner ~ 5
Private Sub Command3_Click()
lookupLastest
end sub

Private Sub btnSaveTran_Click()
doSave
end sub

Private Sub cmdListQueue_Click()
frmLUVProduct.openner = "Planner"
frmLUVProduct.Show
end sub

Private Sub ListView1_ItemClick(ByVal Item As MSCometLib.ListItem)
If TypeOf Item.Tag Is cOrder Then
Dim o As cOrder
Set o = Item.Tag
txtTranno.Text = o.orderno
txtOrder.Text = o.productcode
txtQty.Text = o.Quantity
DTPicker2.value = o.duedate
End If
end sub

Private Sub insertValue()
Dim i As Integer
Dim o As cOrder
Dim strSQL As String
Dim parainsert(5) As Variant
Dim cmd As ADODB.Command
Dim lAffected As Long
Dim strParameter As String
For i = 1 To ListView1.ListItems.Count
Set cmd = New ADODB.Command
Set o = ListView1.ListItems.Item(i).Tag
strSQL = "insert into transact (tranno, orderno,product_code, qty, duedate) "
strParameter = " values (" & CStr(tranno) & "," & o.orderno & "," & o.productcode & "," & o.Quantity & "," & o.duedate & ")"
strSQL = strSQL & strParameter
cmd.ActiveConnection = db
cmd.CommandText = strSQL
cmd.Execute
Next i
strSQL = "update systab set [value] =" & tranno & " where keyvalue ='tranno'"
cmd.CommandText = strSQL
cmd.Execute
setscreen
end sub

```

```

Private Sub savetolist()
Dim o As cOrder
Set o = New cOrder
Dim lsttmp As ListItem
o.orderno = txtOrder.Text
o.productcode = txtProduct.Text
o.Quality = txtQty.Text
o.duedate = DTPicker2.value
o.cav = oProduct.cav
o.ct = oProduct.ct
o.phr = (3600 * o.cav * 0.9) / (o.ct)
o.processtime = (o.Quality / o.phr)
If hasOrder(o.orderno) Then
alert "?????_????????????????????????????????????????"
'Case update
Else
Set lsttmp = ListView1.ListItems.Add(, , o.orderno)
With lsttmp
.Text = o.orderno
.SubItems(1) = o.productcode
.SubItems(2) = o.Quality
.SubItems(3) = IIf(IsNull(o.duedate), "", o.duedate)
.SubItems(4) = CStr(o.phr)
frmPlanner - 6
.SubItems(5) = CStr(o.processtime)
End With
End If
Set lsttmp.Tag = o
end sub

Private Sub updateValue()
'Update
end sub

Private Sub setscreen()
txtTranno.Text = tranno
end sub

Private Sub findSolution()
'Find Solution
Dim colMachine As Collection
Dim i As Integer
Dim j As Integer
Dim o As cTmpTransact
Dim oMachine As cMachine
Dim oMachineTime As cMachineTime
Dim ctxOmachineTime As cMachineTimeContext
findpriority
Set colMachine = findAllmachine
' insert all machine in the table
For i = 1 To colMachine.Count
Set oMachine = colMachine.Item(i)
Set ctxOmachineTime = New cMachineTimeContext
ctxOmachineTime.updateDB ("insert into tmachine_time (machine_code,machine_size," & "sumtime)
values (" & oMachine.machine_code & "," & oMachine.machine_size & ",0)")

```

```

Next i
putJob
findpermutation
settime
findoptimum
end sub

Private Function hasOrder(ByVal orderno As String) As Boolean
hasOrder = False
end function

Private Sub insertToTmp(ByVal o As cOrder)
'insert into tmptable
Dim i As Integer
Dim strSQL As String
Dim parainst(5) As Variant
Dim cmd As ADODB.Command
Dim lAffected As Long
Dim strParameter As String
Set cmd = New ADODB.Command
strSQL = "insert into tmptransact (orderno,product_code, qty,duedate,processtime) "
strParameter = " values ('" & o.orderno & "','" & o.productcode & "','" & o.Quality & "','" & o.duedate & "','" & o.processtime & "')"
strSQL = strSQL & strParameter
cmd.ActiveConnection = db
cmd.CommandText = strSQL
cmd.Execute
end sub

Private Sub findpriority()
' Select from tmptransact order by due_date,custcode,ordernumber
Dim otmp As cTmpTransact
Dim ctxOtmp As cTmpTransactContext
Dim lstmp2 As ListItem
Dim phr As Double
Dim processtime As Double
Dim i As Integer
Set colTransact = New Collection
strBasicSQL = "select product.cav as cav, product.ct as ct, product.cust_code as cust_code ,customer.cust_priority as cust_priority, " & _
"tmptransact.duedate as duedate, tmptransact.orderno as orderno, " & _frmPlanner - 7
"tmptransact.processtime as processtime, product.prod_code as prod_code, " & _
"tmptransact.qty as qty, product.machine_size as machine_size " & _
"from tmptransact,customer, product " & _
"where product.cust_code = customer.cust_code " & _
"and tmptransact.product_code = product.prod_code " & _
"order by product.machine_size,tmptransact.duedate asc, customer.cust_priority desc, tmptransact.orderno asc"
Set ctxOtmp = New cTmpTransactContext
Set colTransact = ctxOtmp.GetData(strBasicSQL)
For i = 1 To colTransact.Count
Set otmp = New cTmpTransact
Set otmp = colTransact.Item(i)
Set lstmp2 = ListView2.ListItems.Add(, , otmp.orderno)
With lstmp2
phr = (3600 * otmp.cav * 0.9) / (otmp.ct)
processtime = (otmp.qty / phr)
otmp.processtime = processtime

```

```

.Text = otmp.orderno
.SubItems(1) = otmp.prod_code
.SubItems(2) = otmp.qty
.SubItems(3) = CStr(otmp.duedate)
.SubItems(4) = ""
.SubItems(5) = CStr(processtime)
.SubItems(6) = otmp.cust_priority
.SubItems(7) = otmp.machine_size
End With
Set lstmp2.Tag = otmp
Next i
end sub

Private Sub deletetmp()
Dim ctxTmp As cTmpTransactContext
Set ctxTmp = New cTmpTransactContext
Dim strSQL As String
strSQL = "delete from tmptransact"
ctxTmp.updateDB (strSQL)
end sub

Private Sub deletemachinetime()
Dim ctxTmp As cTmpTransactContext
Set ctxTmp = New cTmpTransactContext
Dim strSQL As String
strSQL = "delete from Tmachine_time"
ctxTmp.updateDB (strSQL)
end sub

Private Sub deletemachinejob()
Dim ctxTmp As cTmpTransactContext
Set ctxTmp = New cTmpTransactContext
Dim strSQL As String
strSQL = "delete from tmachine_job"
ctxTmp.updateDB (strSQL)
end sub

Private Function findAllmachine() As Collection
Dim ctxMachine As cMachineContext
Set ctxMachine = New cMachineContext
Dim vcollection As Collection
Set vcollection = ctxMachine.GetData("select * from tmachine")
Set findAllmachine = vcollection
end function

Private Sub putJob()
Dim ctxMachineTime As cMachineTimeContext
Dim ctxMachinejob As cMachineJobContext
Dim o As cTmpTransact
Dim colTime As Collection
Dim otime As cMachineTime
Dim ojob As cMachineJob
Set ctxMachineTime = New cMachineTimeContext
Set ctxMachinejob = New cMachineJobContext
Dim strSQL As String
Dim i As Integer

```

```

frmPlanner - 8
Dim lsttmp As ListItem
For i = 1 To ListView2.ListItems.Count
Set o = ListView2.ListItems.Item(i).Tag
strSQL = "select distinctrow machine_code, min (sumtime) as sumtime " & _
" from tmachine_time where machine_size = " & o.machine_size & "" & _
" group by machine_code order by min(sumtime) asc,machine_code asc"Set colTime = ctxMachineTime.GetData(strSQL)
If colTime.Count > 0 Then
Set otime = colTime.Item(1)
strSQL = "insert into tmachine_job (machine_code, prod_code,machine_size) " & _"values (" & otime.machine_code & ", " & o.prod_code & ", " & _
o.machine_size & ")"
Set lsttmp = ListView3.ListItems.Add(, otime.machine_code)
With lsttmp
.Text = otime.machine_code
.SubItems(1) = o.prod_code
End With
If ctxMachinejob.updateDB(strSQL) Then
strSQL = "update tmachine_time set sumtime = sumtime + " & o.processtime & _" where machine_code = " & otime.machine_code & _
""ctxMachineTime.updateDB (strSQL)
End If
End If
Next i
end sub

Private Sub findpermutation()
'Permutation
'select all from tmachine_job order by machine_size
'and then making permutation of job into table tpermutation
Dim oMachinejob As cMachineJob
Dim ctxMachinejob As cMachineJobContext
Set ctxMachinejob = New cMachineJobContext
Dim oMachine As cMachine
Dim ctxMachine As cMachineContext
Set ctxMachine = New cMachineContext
Dim vmachine As New Collection
Dim colresult As Collection
Set colresult = New Collection
Set vmachine = ctxMachine.GetData("select * from Tmachine")
Dim i As Integer
Dim j As Integer
Dim k As Integer
Set vjob = New Collection
ListView4.ListItems.Clear
For i = 1 To vmachine.Count
Set oMachine = vmachine.Item(i)
Set vjob = ctxMachinejob.GetData("select a.machine_code, a.prod_code,b.annealing, b.robot,b.man " & _
" from Tmachine_job a, product b " & _
" where a.machine_code = " & oMachine.machine_code & "" & _
" and a.prod_code = b.prod_code")
If vjob.Count > 0 Then
Set colresult = Pnr(vjob.Count)
Dim lsttmp As ListItem
For j = 1 To colresult.Count
Dim ctxPermutation As cPermutationContext
Dim oPermutation As cPermutation

```

```

Set ctxPermutation = New cPermutationContext
Dim oAny
For Each oAny In colresult
For k = LBound(oAny) To UBound(oAny)
Set oPermutation = New cPermutation
Set oMachinejob = oAny(k)
oPermutation.machine_code = oMachinejob.machine_code
oPermutation.planno = CStr(j)
frmPlanner - 9
oPermutation.itemno = CStr(k)
oPermutation.prod_code = oMachinejob.prod_code
Set lsttmp = ListView4.ListItems.Add(, , CStr(oPermutation.planno))
With lsttmp
.Text = oMachinejob.machine_code
.SubItems(1) = oPermutation.planno
.SubItems(2) = oPermutation.itemno
.SubItems(3) = oPermutation.prod_code
End With
Set lsttmp.Tag = oPermutation
Next k
Next oAny
Next j
End If
Next i
end sub

```

```

Private Function Pnr(iDigit As Integer) As Collection
Dim colNew As Collection
Dim v() As cMachineJob
Dim u() As cMachineJob
Dim oAny
Dim i As Integer
Dim j As Integer
Dim iBuffer As cMachineJob
Dim iResult() As cMachineJob
If iDigit > 0 Then
Set colNew = New Collection
If iDigit = 1 Then
ReDim iResult(1 To iDigit) As cMachineJob
Set iResult(1) = vjob.Item(1)
colNew.Add iResult
Set Pnr = colNew
Else
Set colold = Pnr(iDigit - 1)
ReDim u(1 To iDigit - 1) As cMachineJob
ReDim v(1 To iDigit) As cMachineJob
For Each oAny In colold
u = oAny
For i = LBound(u) To UBound(u)
Set v(i) = u(i)
Next i
Set v(iDigit) = vjob.Item(iDigit)
colNew.Add v
For j = UBound(v) To LBound(v) + 1 Step -1
' just swap

```

```

Set iBuffer = v(j)
Set v(j) = v(j - 1)
Set v(j - 1) = iBuffer
colNew.Add v
Next j
Next oAny
Set Pnr = colNew
End If
Else
Set Pnr = Nothing
End If
Set colNew = Nothing
Set colold = Nothing
end function

Private Sub settime()
'find time
end sub

Private Sub findoptimum()
'find best way
end sub

frmProduct - 1
Option Explicit
frmStartTime - 1
Option Explicit
Public oMachine As cMachine
Public oProduct As cProduct
Private Sub btnSave_Click()
'check that insert all the machine in the table
If Not addall Then
alert "?????????????"
Else
insertstarttable
End If
end sub

Private Sub cmdListQueue_Click()
frmLUVProduct.openner = "Machine"
frmLUVProduct.Show
end sub

Private Sub Command1_Click()
frmLUVProduct.openner = "mProduct"
frmLUVProduct.Show
end sub

Private Sub Command2_Click()
'Keep to list
Dim x As Date
x = dtstarttime.value
'savetolist
Date.Add
end sub

```



```

Private Sub Command3_Click()
'Do save
end sub

Private Sub savetolist()
Dim o As cStart
Set o = New cStart
Dim lsttmp As ListItem
o.machinecode = txtMachinecode.Text
o.prod_code = txtProd.Text
o.startdatetime = CDate(FormatDateTime(dtstarttime.value, vbLongDate) & " " &
FormatDateTime(dtstarttime.value, vbShortTime))
If hasmachine(o.machinecode) Then
alert "?????_??????????????"
Else
'insertTmp
Set lsttmp = ListView1.ListItems.Add(, o.machinecode)
With lsttmp
.Text = o.machinecode
.SubItems(1) = o.prod_code
.SubItems(2) = o.startdatetime
End With
End If
Set lsttmp.Tag = o
end sub

Private Function addall() As Boolean
'check
end function

Private Sub insertstarttable()
'insert starttable
'insert into tmptable
Dim i As Integer
Dim strSQL As String
Dim cmd As ADODB.Command
Dim lAffected As Long
Dim strParameter As String
Dim oSetup As cStart
For i = 1 To ListView1.ListItems.Count
Set oSetup = New cStart
Set oSetup = ListView1.ListItems.Item(i).Tag
Set cmd = New ADODB.Command
frmStartTime - 2
If hasmachine(oSetup.machinecode) Then
strSQL = "update tsetup set startdate = " & oSetup.startdatetime & ", " & _
"lastproduct = " & oSetup.prod_code & " " & _
"where machineno = " & oSetup.machinecode & ""
cmd.ActiveConnection = db
cmd.CommandText = strSQL
cmd.Execute
Else
strSQL = "insert into tsetup (machineno, startdate, lastproduct) "
'strParameter = " values (" & osetup.orderno & ", " & osetup.productcode & ", " & o.
Quality & ", " & o.duedate & ", " & o.processtime & ")"

```

```

strSQL = strSQL & strParameter
cmd.ActiveConnection = db
cmd.CommandText = strSQL
cmd.Execute
End If
Next i
end sub

Private Function hasmachine(ByVal strmachinecode As String) As Boolean
'check
Dim ctxstart As cStartContext
Dim vcol As Collection
Dim strSQL As String
Set ctxstart = New cStartContext
strSQL = " select * from tsetup where machineno ='" & strmachinecode & "'"
Set vcol = ctxstart.GetData(strSQL)
If vcol.Count > 0 Then
hasmachine = True
Else
hasmachine = False
End If
end function

business = 1
Option Explicit
Const MODULENAME = "business"
Public Sub ShowError(ByVal strModule As String, ByVal strMethod As String)
Dim strMsgbox As String
strMsgbox = _
"Error occured at " & strModule & ", " & strMethod & vbCrLf & _
"Err.Number : " & Err.Number & vbCrLf & _
Err.Description
' 2002-02-24
alert strMsgbox, vbCritical
If Not (logger Is Nothing) Then
logger.fatalLog strModule, strMethod, Err.Number & " " & Err.Description
End If
end sub

Public Function alert(strMessage As String, Optional buttons As VbMsgBoxStyle = vbOKOnly) As VbMsgBoxResult
Dim result As VbMsgBoxResult
result = MsgBox(strMessage, buttons, ALERTTITLESTRING)
alert = result
end function

Public Sub PrepareContext()
' Set ctxClient = New OClientContext
' Set ctxProvince = New OProvinceContext
' Set ctxPostal = New OPostalContext
' Set ctxDoctor = New ODoctorContext
' Set ctxDiagonose = New ODiagonoseContext
' Set ctxQueueMaster = New OQueueMasterContext
' Set ctxQueueDetail = New OQueueDetailContext
' Set ctxColor = New OColorContext
end sub

```

```

' Description
' Responded for connection database.
' History
Public Sub PrepareConnection()
Const METHODNAME = "PrepareConnection"
On Error GoTo ErrHandler
Set db = New ADODB.Connection
Dim strDBConType As String
Dim strCon As String
mIniUtil.GetProfileItem "Settings", "DBConnection.Type", strInitFullPath,
strDBConType
' DBConnection.Type = ODBC | OLE DB
,
' DBConnection.Type=OLE DB
' DBConnection.Provider=Microsoft.Jet.OLEDB.4.0
' DBConnection.Data Source=D:\piek\queue\db.mdb
,
' DBConnection.Type=ODBC
' DBConnection.DSN=queue
,
Select Case strDBConType
Case "OLE DB"
Dim provider As String
Dim datasource As String
mIniUtil.GetProfileItem "Settings", "DBConnection.Provider", strInitFullPath,
provider
mIniUtil.GetProfileItem "Settings", "DBConnection.Data Source", strInitFullPath,
datasource
strCon = "Provider=" & provider & ";Data Source=" & datasource
Case "ODBC"
Dim dsn As String
mIniUtil.GetProfileItem "Settings", "DBConnection.DSN", strInitFullPath, dsn
strCon = "DSN=" & dsn
Case Else
business - 2
logger.fatalLog MODULENAME, METHODNAME, "invalid DBConnection.Type : [" & strDBConType & "]"
End Select
db.CursorLocation = adUseClient
db.Open strCon
ExitMethod:
Exit Sub
ErrHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end sub

converter - 1
Option Explicit
Const MODULENAME = "converter"
' Default value for conversion to data failed
Const epochDate = #1/1/1900# ' 01-Jan-1900
' Create : 2002-01-19
' Description :
' Convert to currency
' return value :

```

```

' x as currency, if conversion is successful
' default, if x is null, nothing, empty, empty string
' convertFail, conversion failed
Public Function toCurrency( _ByVal x As Variant, _Optional ByVal default As Currency = 0, _Optional ByVal convertFail As Currency = 0) As
Currency
Const METHODNAME = "toCurrency"
Dim iVarType As VbVarType
Dim result As Currency
On Error GoTo ErrHandler
' Sorry but Basic has no short-circuit logic evaluation
' vb can't evaluate
' IsNull(x) Or IsEmpty(x) or (x is nothing)
' if x is nothing, this expression will raise error
iVarType = VarType(x)
Select Case iVarType
Case vbNull, vbEmpty
result = default
GoTo ExitMethod
Case vbObject
If x Is Nothing Then
result = default
GoTo ExitMethod
End If
Case vbString
If Len(Trim(x)) = 0 Then
result = default
GoTo ExitMethod
End If
End Select
On Error Resume Next
result = CCur(x)
' if "Type mismatch" error occurred
' then conversion is failed
If Err.Number = 13 Then
result = convertFail
GoTo ExitMethod
End If
On Error GoTo ErrHandler
ExitMethod:
toCurrency = result
Exit Function
ErrHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end function

' Create : 2002-01-19
' Description :
' Convert to date
' return value :
' x as date, if conversion is successful
' default, if x is null, nothing, empty, empty string
' convertFail, conversion failed
Public Function toDate( _ByVal x As Variant, _Optional ByVal default As Date = ePocDate, _
Optional ByVal convertFail As Date = ePocDate) As Date

```

```

Const METHODNAME = "toDate"
Dim iVarType As VbVarType
converter - 2
Dim result As Date
On Error GoTo ErrHandler
' Sorry but Basic has no short-circuit logic evaluation
' vb can't evaluate
' IsNull(x) Or IsEmpty(x) or (x is nothing)
' if x is nothing, this expression will raise error
iVarType = VarType(x)
Select Case iVarType
Case vbNull, vbEmpty
result = default
GoTo ExitMethod
Case vbObject
If x Is Nothing Then
result = default
GoTo ExitMethod
End If
Case vbString
If Len(Trim(x)) = 0 Then
result = default
GoTo ExitMethod
End If
End Select
On Error Resume Next
result = CDate(x)
' if "Type mismatch" error occurred
' then conversion is failed
If Err.Number = 13 Then
result = convertFail
GoTo ExitMethod
End If
On Error GoTo ErrHandler
ExitMethod:
toDate = result
Exit Function
ErrHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end function

' Create : 2002-01-19
' Description :
' Convert to long
' return value :
' x as long, if conversion is successful
' default, if x is null, nothing, empty, empty string
' convertFail, conversion failed
Public Function toLong( _
ByVal x As Variant, _
Optional ByVal default As Long = 0, _
Optional ByVal convertFail As Long = 0) As Long
Const METHODNAME = "toLong"
Dim iVarType As VbVarType

```

```

Dim result As Long
On Error GoTo ErrHandler
' Sorry but Basic has no short-circuit logic evaluation
' vb can't evaluate
' IsNull(x) Or IsEmpty(x) or (x is nothing)
' if x is nothing, this expression will raise error
iVarType = VarType(x)
Select Case iVarType
Case vbNull, vbEmpty
result = default
GoTo ExitMethod
Case vbObject
If x Is Nothing Then
result = default
GoTo ExitMethod
End If
Case vbString
If Len(Trim(x)) = 0 Then
result = default
converter - 3
GoTo ExitMethod
End If
End Select
On Error Resume Next
result = CLng(x)
' if "Type mismatch" error occurred
' then conversion is failed
If Err.Number = 13 Then
result = convertFail
GoTo ExitMethod
End If
On Error GoTo ErrHandler
ExitMethod:
toLong = result
Exit Function
ErrorHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end function

' Create : 2002-01-19
' Description :
' Convert to String
'
' Note:
' This toString() method had been rename to toStringValue()
' because toString() function had implemented
' into many class modules
'
' return value :
' x as String, if conversion is successful
' default, if x is null, nothing, empty
' convertFail, conversion failed
Public Function toStringValue(ByVal x As Variant, Optional ByVal default As String = "", Optional B
yVal convertFail As String = "") As String

```

```

Const METHODNAME = "toString"
Dim iVarType As VbVarType
Dim result As String
On Error GoTo ErrHandler
' Sorry but Basic has no short-circuit logic evaluation
' vb can't evaluate
' IsNull(x) Or IsEmpty(x) or (x is nothing)
' if x is nothing, this expression will raise error
iVarType = VarType(x)
Select Case iVarType
Case vbNull, vbEmpty
result = default
GoTo ExitMethod
Case vbObject
If x Is Nothing Then
result = default
GoTo ExitMethod
End If
Case vbString
' intentional no check empty String
' This is toStringValue() function! :)
' empty String is just perfect String
result = x
GoTo ExitMethod
End Select
On Error Resume Next
result = CStr(x)
' if "Type mismatch" error occurred
' then conversion is failed
If Err.Number = 13 Then
result = convertFail
GoTo ExitMethod
End If
On Error GoTo ErrHandler
converter - 4
ExitMethod:
toStringValue = result
Exit Function
ErrHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end function

' Create : 2001-12-31
' Description :
' convert null to empty string
' for display purpose
Public Function NullEmptyString(ByVal x As Variant) As String
NullEmptyString = toStringValue(x)
end function

' Create : 31-Dec-2001
' Description :
' enclosed with single quote,
' for construct SQL statement

```

```

' INSERT/UPDATE/SELECT (WHERE) statement
' History
' 2001-12-31 Piek Created
' 2002-01-19 Piek Rewrite based on toStringValue function
' This method similar to toStringValue
' but has many thing difference,
' - if string conversion failed, this will flag an error and continue.
' - if conversion successful, output will enclosed with single quote
', on the other hand, NULL has not.
' 2002-01-21 Piek Rewrite new concept
Public Function toDBString(ByVal x As Variant, Optional ByVal EmptyStringBeNull
As Boolean = False)
As String
Const METHODNAME = "toDBString"
Const default = "NULL"
Const convertFail = "NULL"
Const NULLString = "NULL"
Dim iVarType As VbVarType
Dim result As String
On Error GoTo ErrHandler
' Sorry but Basic has no short-circuit logic evaluation
' vb can't evaluate
' IsNull(x) Or IsEmpty(x) or (x is nothing)
' if x is nothing, this expression will raise error
iVarType = VarType(x)
Select Case iVarType
Case vbNull, vbEmpty
result = default
GoTo ExitMethod
Case vbObject, vbDataObject
If x Is Nothing Then
result = default
GoTo ExitMethod
End If
Case vbString
' intentional no check empty String
' This is toStringValue() function! :)
' empty String is just perfect String
If EmptyStringBeNull And Len(x) = 0 Then
result = NULLString
Else
result = "'" & x & "'"
End If
GoTo ExitMethod
End Select
On Error Resume Next
result = "'" & CStr(x) & "'"
' if "Type mismatch" error occurred
' then conversion is failed
If Err.Number = 13 Then
converter = 5
' This should not happed!
ShowError MODULENAME, METHODNAME
result = convertFail
GoTo ExitMethod

```



```

End If
On Error GoTo ErrHandler
ExitMethod:
toDBString = result
Exit Function
ErrHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end function

Public Function toDirectDBString( _ByVal x As Variant, _Optional ByVal EmptyStringBeNull As Boolean = False) As Variant
Const METHODNAME = "toDBString"
Const convertFail = ""
Dim iVarType As VbVarType
Dim result As Variant
On Error GoTo ErrHandler
' Sorry but Basic has no short-circuit logic evaluation
'' vb can't evaluate
' IsNull(x) Or IsEmpty(x) or (x is nothing)
' if x is nothing, this expression will raise error
iVarType = VarType(x)
Select Case iVarType
Case vbNull, vbEmpty
result = Null
GoTo ExitMethod
Case vbObject
If x Is Nothing Then
result = Null
GoTo ExitMethod
End If
Case vbString
' intentional no check empty String
' This is toStringValue() function! :)
' empty String is just perfect String
If EmptyStringBeNull And Len(x) = 0 Then
result = Null
Else
result = x
End If
GoTo ExitMethod
End Select
On Error Resume Next
result = "" & CStr(x) & ""
' if "Type mismatch" error occurred
' then conversion is failed
If Err.Number = 13 Then
' This should not happed!
ShowError MODULENAME, METHODNAME
result = convertFail
GoTo ExitMethod
End If
On Error GoTo ErrHandler
ExitMethod:
toDirectDBString = result
Exit Function

```

```

ErrorHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
converter - 6
end function

dateconverter - 1
Option Explicit
Const MODULENAME = "dateconverter"
Const DATESEP = "-"
Const TIMESEP = ":"
' Piek
' Convert from date to string 'YYYY-MM-DD'
' Created : 18-Jan-2002
Function DateToYYYYMMDD(dDate As Date, Optional BuddaEra As Boolean = False,
Optional ZeroPad As Boolean = False) As String
Dim iDay As Integer, iMonth As Integer, iYear As Integer
Dim strDay As String, strMonth As String, strYear As String
iDay = DatePart("d", dDate)
iMonth = DatePart("m", dDate)
iYear = DatePart("yyyy", dDate)
If BuddaEra Then
If iYear < 2500 Then iYear = iYear + 543
End If
strYear = iYear
If Not ZeroPad Then
strMonth = iMonth
strDay = iDay
Else
strMonth = Format(iMonth, "00")
strDay = Format(iDay, "00")
End If
DateToYYYYMMDD = strYear & DATESEP & strMonth & DATESEP & strDay
end function

Function DateToYYYYMMDDHHMMSS(dDate As Date, Optional BuddaEra As Boolean =
False, Optional ZeroPad
As Boolean = False) As String
DateToYYYYMMDDHHMMSS = _
DateToYYYYMMDD(dDate, BuddaEra, ZeroPad) & " " & _DateToHHMMSS(dDate, ZeroPad)
end function

' Piek
' Convert from date to string 'HH:MM:SS'
' Created : 18-Jan-2002
Function DateToHHMMSS(dDate As Date, Optional ZeroPad As Boolean = True) As String
Dim iHour As Integer, iMinute As Integer, iSecond As Integer
Dim strHour As String, strMinute As String, strSecond As String
iHour = DatePart("h", dDate)
iMinute = DatePart("n", dDate)
iSecond = DatePart("s", dDate)
If Not ZeroPad Then
strHour = iHour
strMinute = iMinute

```



```

Public Const MSG_CONFIRMSYSTEMEXIT = "??????_????????????????????"
Public Const MSG_CONFIGNOTFOUND = "??? config.ini ??????" & vbCrLf &
"????????????_?"
Public Const ALERTTITLESTRING As String = "Planner System"
Public Const CW_ID = 600
Public Const CW_QUEUEENO = 1000
Public Const CW_PROVINCENAME = 1800
Public Const CW_NAME = 1800
Public Const CW_SURNAME = 2100
Public Const CW_DATE = 1800
Public Const CW_DURATION = 1000
Public Const CW_COLOR = 1000
Public Const CW_TREATMENT = 2100
Public Const CM_ID = "ID"
Public Const CM_NAME = "???"
Public Const CM_SURNAME = "?????"
Public Const CM_QUEUEENO = "?????"
Public Const CM_DATE = "?????"
Public Const CM_COLOR = "?"
Public Const CM_DURATION = "?????"
Public Const CM_TREATMENT = "?????"
Public Const CM_PROVINCENAME = "?_?"
Public bSystemMenuHooked As Boolean
Planner_main - 1
Option Explicit
Const MODULENAME = "Planner_main"
Public Sub main()
Const METHODNAME = "main"
On Error GoTo ErrHandler
strInitFullPath = App.Path & "config.ini"
Set logger = New OLogger
logger.readConfig strInitFullPath
If Not checkConfig(strInitFullPath) Then
Exit Sub
End If
Dim strAlertTitle As String
Dim strHook As String
With mIniUtil
.GetProfileItem "Settings", "Alert.Title", strInitFullPath, strAlertTitle
.GetProfileItem "Settings", "Menu", strInitFullPath, strHook
End With
bSystemMenuHooked = (toLong(strHook) = 1)
'test
frmMain.Show
frmMain.Caption = frmMain.Caption & " v" & App.Major & "." & App.Minor & "." &
App.Revision
logger.infoLog MODULENAME, METHODNAME, "start application"
frmMain.MousePointer = vbHourglass
PrepareConnection
frmMain.MousePointer = vbDefault
frmMain.MousePointer = vbHourglass
PrepareContext
frmMain.MousePointer = vbDefault
ExitMethod:
Exit Sub

```

```

ErrorHandler:
ShowError MODULENAME, METHODNAME
Resume ExitMethod
end sub

Private Function checkConfig(ByVal str As String) As Boolean
Const METHODNAME = "main"
Dim result As Boolean
Dim fso As FileSystemObject
Set fso = New FileSystemObject
If Not fso.FileExists(str) Then
alert MSG_CONFIGNOTFOUND, vbCritical
End If
result = (fso.FileExists(str))
Set fso = Nothing
ExitMethod:
checkConfig = result
end function

Utils - 1
Option Explicit
Const MODULENAME = "Utils"
Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal
lpClassName As String, B
yVal lpWindowName As String) As Long
Declare Function GetDesktopWindow Lib "user32" () As Long
Declare Function GetWindow Lib "user32" (ByVal hWnd As Long, ByVal wCmd As Long)
As Long
Public Const GWL_ID = (-12)
Public Const GW_HWNDNEXT = 2
Public Const GW_CHILD = 5
Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hWnd As
Long, ByVal lpClass
Name As String, ByVal nMaxCount As Long) As Long
Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hWnd
As Long, ByVal lpStr
ing As String, ByVal cch As Long) As Long
Declare Function GetParent Lib "user32" (ByVal hWnd As Long) As Long
' 2002-03-12 Add About in System menu
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal
lpPrevWndFunc As Long,
ByVal hWnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As
Long) As Long
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hWnd
As Long, ByVal nInde
x As Long, ByVal dwNewLong As Long) As Long
Declare Function AppendMenu Lib "user32" Alias "AppendMenuA" (ByVal hMenu As
Long, ByVal wFlags As
Long, ByVal wIDNewItem As Long, ByVal lpNewItem As String) As Long
Declare Function GetSystemMenu Lib "user32" (ByVal hWnd As Long, ByVal bRevert
As Long) As Long
' SetWindowLong : message from the window menu ( or system or control menu )
Public Const WM_SYSCOMMAND = &H112
' AppendMenu : contains a pointer to a null-terminated string.
Public Const MF_STRING = &H0

```

```

' the identifier of the new menu item (About)
Public Const SC_ABOUT = 1
' SetWindowLong : sets a new address for the window procedure
Public Const GWL_WNDPROC = -4
Global lpPrevWndProc As Long
Global gWH As Long
' 2002-03-12
' although this routine use only in mdiMain,
' but AddressOf only use with procedure/function in standard module
Function WindowProc(ByVal hw As Long, ByVal uMsg As Long, ByVal wParam As Long,
ByVal lParam As Long) As Long
' We need to trap the WM_SYSCOMMAND message to determine when the user has
clicked on our
' new menu item
' the message is stored in uMsg
' our new menu item is identified as SC_ABOUT
If uMsg = WM_SYSCOMMAND Then
Select Case wParam And &HFFFF&
Case SC_ABOUT
' frmAbout.Show vbModal
End Select
End If
' Always call the original handler when you are done
WindowProc = CallWindowProc(lpPrevWndProc, hw, uMsg, wParam, lParam)
end function

' Created : 2002-01-18
' Description :
' Truncate time portion from the given date
' Return value :
' if x is Date, returns x as Date without time portion data
' ,otherwise returns Null.
Public Function TruncateTime(ByVal x As Variant) As Variant
If IsNull(x) Then
TruncateTime = Null
ElseIf VarType(x) <> vbDate Then
TruncateTime = Null
Else
TruncateTime = DateSerial(Year(x), Month(x), Day(x))
End If
end function

' Created : 2002-01-18
Utils - 2
' Description :
' Center the specified form within the screen
Sub CenterForm(f As Form)
f.Move (Screen.Width - f.Width) \ 2, (Screen.Height - f.Height) \ 2
end sub

' Created : 2002-01-18
' Description :
' Return the filename portion of a full pathname
Function ExtractFile(ByVal PathName As String) As String

```

```

Dim f As String
Dim n As Integer
f = PathName
' remove string content from left to the right most '^'
Do
n = InStr(f, '^')
If n > 0 Then f = Right(f, Len(f) - n)
Loop While n > 0
ExtractFile = f
end function

' Created : 2002-01-18
' Description :
' Return the basename portion of a full pathname
Function ExtractBase(ByVal PathName As String) As String
Dim f As String
Dim n As Integer
f = ExtractFile(PathName)
n = InStr(f, ".")
If n > 0 Then f = Left(f, n - 1)
ExtractBase = f
end function

' Created : 2002-01-18
' Description :
' Return the directory path portion of a full pathname
Function ExtractPath(ByVal PathName As String) As String
Dim f As String
Dim n As Integer
f = PathName
Do
n = InStr(f, '^')
If n > 0 Then f = Right$(f, Len(f) - n)
Loop While n > 0
ExtractPath = Left(PathName, Len(PathName) - Len(f))
end function

,

' Description :
' Disable/enable text widget, and set background, foreground color.
' Parameters :
' obj : user interface component
' bEnabled : how to render control, false as disabled, true as enabled
,

' History
' ???-??-?? Piek Created
' 2002-01-19 Piek Put comment
Public Sub setTextEnable(obj As Object, ByVal bEnabled As Boolean)
Const METHODNAME = "setTextEnable"
On Error GoTo ErrHandler
obj.Enabled = bEnabled
obj.TabStop = bEnabled
Utils - 3
If TypeOf obj Is VB.Label Then
' obj.BackColor = IIf(bEnabled, vbButtonFace, vbButtonFace)
' do nothing

```

```

Else
obj.BackColor = IIf(bEnabled, vbWindowBackground, vbButtonFace)
End If
obj.ForeColor = IIf(bEnabled, vbWindowText, vbGrayText)
Exit Sub
ErrorHandler:
' Object doesn't support this property or method
' The Widget may not has ForeColor property
If Err.Number = 438 Then
Resume Next
Else
ShowError MODULENAME, METHODNAME
End If
end sub

' Description :
' Render visual component as enabled/disabled.
' Use in form when user save data,
' all textboxes should be unchangeable (gray out) but can got focus.
' Parameters :
' obj : user interface component
' bLocked : how to render visualize control, false as disabled, true as enabled
' setBackColor : if false then did not set BackColor ( for RadioButton )
'
' History
' 2001-12-31 Pick Created
' 2002-01-19 Pick Add setBackColor parameter
'

Public Sub LockControl(obj As Object, ByVal bLocked As Boolean, Optional
setBackColor As Boolean =
True)
Const METHODNAME = "LockControl"
On Error Resume Next
obj.Locked = bLocked
If Err.Number = 438 Then
obj.Enabled = Not bLocked
End If
If setBackColor Then
obj.BackColor = IIf(Not bLocked, vbWindowBackground, vbButtonFace)
End If
ExitMethod:
Exit Sub
ErrorHandler:
' Object doesn't support this property or method
' The Widget may not has ForeColor property
If Err.Number = 438 Then
Resume Next
Else
ShowError MODULENAME, METHODNAME
End If
end sub

' Description :
' text objects want to highlight all text.
' Application Example :

```



```

' When data validate failed, move focus to the component
',highlight text in the box.
'
' History
' ?????-??-?? Piek Created
'
Public Sub selectAll(obj As Object)
obj.SelStart = 0
obj.SelLength = Len(obj.Text)
end sub

' Description :
' This function will return date in format DD/MM/YYYY
' Input value :
' Literal data string as D/M/YYYY or D/MM/YYYY or DD/M/YYYY or DD/MM/YYYY
' Return value :
Utils - 4
' if invalid format, it return ""
' otherwise returns DD/MM/YYYY
' History
' ?????-??-?? ????? Create
' 2002-01-19 Piek Comment
)

Public Function CheckDateFormat(ByVal pDate As String, ByVal pType As String) As
String
On Error GoTo ErrHandler
Dim i As Integer
Dim j As Integer
Dim strTmpDate As String
Dim temp(0 To 2) As String
j = 0
strTmpDate = Trim$(pDate)
i = InStr(pDate, "_")
If i > 0 Then
CheckDateFormat = ""
Exit Function
End If
pDate = Trim$(pDate)
i = InStr(pDate, "/")
If i <= 0 Then
CheckDateFormat = ""
Else
Do Until i = 0
temp(j) = Left$(pDate, i - 1)
pDate = Mid$(pDate, i + 1, Len(pDate) - i)
i = InStr(pDate, "/")
j = j + 1
If j > 2 Then
CheckDateFormat = ""
Exit Function
End If
Loop
temp(j) = pDate
If (pType = "T") Then
If ((Val(temp(2)) - 543) < 1900) Then

```

```

CheckDateFormat = ""
Exit Function
End If
End If
If Trim$(temp(0)) = "" Or Trim$(temp(1)) = "" Or Trim$(temp(2)) = "" Or _
Len(Trim$(temp(0))) > 2 Or Len(Trim$(temp(1))) > 2 Or Len(Trim$(temp(2))) <> 4
Then
CheckDateFormat = ""
Else
Select Case Val(temp(1))
Case 1, 3, 5, 7, 8, 10, 12
If Val(temp(0)) < 1 Or Val(temp(0)) > 31 Then
CheckDateFormat = ""
Else
CheckDateFormat = SetDateFormat(strTmpDate)
End If
Case 4, 6, 9, 11
If Val(temp(0)) < 1 Or Val(temp(0)) > 30 Then
CheckDateFormat = ""
Else
CheckDateFormat = SetDateFormat(strTmpDate)
End If
Case 2
If (pType = "T") Then
temp(2) = str$(Val(temp(2)) - 543)
End If
If Right$(Trim$(temp(2)), 2) = "00" Then
If Val(temp(2)) Mod 400 = 0 Then
If Val(temp(0)) < 1 Or Val(temp(0)) > 29 Then
CheckDateFormat = ""
Else
CheckDateFormat = SetDateFormat(strTmpDate)
End If
End If
Else
If Val(temp(0)) < 1 Or Val(temp(0)) > 28 Then
CheckDateFormat = ""
Else
Utils - 5
CheckDateFormat = SetDateFormat(strTmpDate)
End If
End If
Else
If Val(temp(2)) Mod 4 = 0 Then
If Val(temp(0)) < 1 Or Val(temp(0)) > 29 Then
CheckDateFormat = ""
Else
CheckDateFormat = SetDateFormat(strTmpDate)
End If
End If
Else
If Val(temp(0)) < 1 Or Val(temp(0)) > 28 Then
CheckDateFormat = ""
Else
CheckDateFormat = SetDateFormat(strTmpDate)
End If
End If
End If

```

```

End If
Case Else
CheckDateFormat = ""
End Select
End If
End If
Exit Function
ErrorHandler:
MsgBox Err.Number & " " & Err.Description, vbCritical
end function

Public Function SetDateFormat(strDate As String) As String
'change date format from x/x/xxxx to 0x/0x/xxxx
Dim i As Integer
Dim strTmp As String
strTmp = Trim$(strDate)
SetDateFormat = ""
i = InStr(strTmp, "/")
SetDateFormat = SetDateFormat & Format$(Left$(strTmp, i - 1), "00") & "/"
strTmp = Right(strTmp, Len(strTmp) - i)
i = InStr(strTmp, "/")
SetDateFormat = SetDateFormat & Format$(Left$(strTmp, i - 1), "00") & "/"
strTmp = Right(strTmp, Len(strTmp) - i)
SetDateFormat = SetDateFormat & Format$(Left$(strTmp, Len(strTmp)), "0000")
end function

' Description :
' Returns true if the specific year is a leap year.
' Input value :
' iYear : year (Gregorian calendar)
' History
' 2002-01-19 Piek Create
'

Public Function isLeapYear(ByVal iYear As Integer) As Boolean
isLeapYear = False
If (iYear Mod 400) = 0 Then
isLeapYear = True
ElseIf (iYear Mod 100) = 0 Then
isLeapYear = False
ElseIf (iYear Mod 4) = 0 Then
isLeapYear = True
End If
end function

' Description :
' Returns maximum day ( base 1 ) for the given month and year
' Input value :
' iYear : year (Gregorian calendar)
' iMonth : month (base 1, Jan = 1, .. , Dec = 12 )
' History
' 2002-01-19 Piek Create
'

Public Function getMaxDay(ByVal iYear As Integer, ByVal iMonth As Integer) As
Integer
Utils - 6

```

```

Dim bLeapYear As Boolean
Select Case iMonth
Case 1, 3, 5, 7, 8, 10, 12
getMaxDay = 31
Case 4, 6, 9, 11
getMaxDay = 30
Case 2
getMaxDay = IIf(isLeapYear(iYear), 29, 28)
End Select
end function

' Description:
' Get window handler without specific an exact title.
' Reference
' (Q147659) HOWTO: Get a Window Handle Without Specifying an Exact Title
' History
' 2002-02-22 Created
'

Public Function FindWindowLike(hwndStart As Long, WindowText As String) As Long
Dim hWnd As Long
Dim sw As String ' hold window title
Dim sc As String ' hold classname
Dim chwnd As Long ' hwnd of first child window
If hwndStart = 0 Then
hwndStart = GetDesktopWindow()
hWnd = GetWindow(hwndStart, GW_CHILD)
Else
hWnd = GetWindow(hwndStart, GW_CHILD)
End If
Do Until hWnd = 0
sw = Space(255)
sw = Left(sw, GetWindowText(hWnd, sw, 255))
sc = Space(255)
sc = Left(sc, GetClassName(hWnd, sc, 255))
If sw Like WindowText Then
FindWindowLike = hWnd
' Debug.Print "Window Found: "
' Debug.Print " Window Text : " & sw
' Debug.Print " Window Class : " & sc
' Debug.Print " Window Handle : " & CStr(hwnd)
End If
' if has first child
' recursive check
' if found then return result
' end if
chwnd = GetWindow(hWnd, GW_CHILD)
If chwnd <> 0 Then
chwnd = FindWindowLike(chwnd, WindowText)
If chwnd <> 0 Then
FindWindowLike = chwnd
End If
End If
hWnd = GetWindow(hWnd, GW_HWNDNEXT)
Loop
' Debug.Print "Window Handle : " & CStr(hwndStart)

```

```

' Debug.Print "Loop through : " & c
' Debug.Print
end function

' Description:
' Return true if has any window title contain 'Microsoft Visual Basic'
' History
' 2002-02-22 Created
Public Function isVBIDERunning() As Boolean
If FindWindowLike(0, "**Microsoft Visual Basic*") > 0 Then
isVBIDERunning = True
Else
Utils - 7
isVBIDERunning = False
End If
end function

' Description:
' Show content of array, for debugging parameter query.
,
' History
' 2002-03-03 Created
,
Public Function showArray(x As Variant) As String
If Not IsArray(x) Then
Exit Function
End If
Dim i As Integer
For i = LBound(x) To UBound(x)
showArray = showArray & "[" & i & "]" & TypeName(x(i)) & ": [" & x(i) & "]" &
vbCrLf
Next i
end function

xarray - 1
Option Explicit
' Created : 2002-01-12
' Purpose : copy data from recordset into xArrayDB
' Parameters
,
' iColStart : offset xarraydb column , dimension start value
' iColAppend : appended xarraydb column ( for keep other information such as
record status
,
Public Function loadXArray(rs As ADODB.Recordset, Optional iColStart As Integer
= 0, Optional iColA
ppend As Integer = 0) As XArrayDB
Dim oResult As XArrayDB
Set oResult = New XArrayDB
oResult.Clear
Dim iDBCcolMax As Integer
Dim iDBCcolCount As Integer
Dim iRow As Long
iDBCcolMax = rs.Fields.Count
iRow = 0

```

```

oResult.ReDim 0, -1, iColStart, iColStart + iDBCcolMax + iColAppend
oResult.Clear
Dim str As String
Do While Not rs.EOF
oResult.AppendRows 1
For iDBCcolCount = 0 To iDBCcolMax - 1
str = dbToString(rs.Fields(iDBCcolCount).value)
oResult.value(iRow, iDBCcolCount + iColStart) = str
Next iDBCcolCount
rs.MoveNext
iRow = iRow + 1
Loop
Set loadXArray = oResult
end function

Public Function dbToString(x As Variant) As String
Select Case VarType(x)
Case vbEmpty
dbToString = ""
Case vbNull
dbToString = ""
Case vbInteger, vbLong, vbByte
dbToString = x
Case vbSingle, vbDouble
dbToString = x
Case vbCurrency
dbToString = x
Case vbDecimal
dbToString = x
Case vbDate
dbToString = x
Case vbString
dbToString = x
Case vbBoolean
dbToString = x
Case Else
End Select
end function

' 2002-02-10 Created
' 2002-02-17 Beware LOV data is Empty
*
' Parameter
' bEmpty : if true, then xdb just create structure, has no data ( in case of
user cancel )
' xdb : (Output) result
' Description
' copy data in current record in recordset into XArray DB
' useful in LOV form
' Note:
xarray - 2
' * In case of bEmpty is true, why xdb must create, why not just Null ?
Sub recordsetToXArray(tdbg As TDBGGrid, rs As ADCDB.Recordset, xdb As XArrayDB,
Optional bEmpty As Boolean = False)

```

```

If xdb Is Nothing Then
Set xdb = New XArrayDB
End If
xdb.ReDim 0, 0, 0, tdbg.Columns.Count - 1
xdb.Clear
' 2002-02-17
If Not bEmpty And Not rs.EOF And Not rs.BOF Then
Dim i As Integer
For i = 0 To tdbg.Columns.Count - 1
xdb(0, i) = rs.Fields(i).value
Next i
End If
end sub

cMachine - 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarmachine_code As String 'local copy
Private mvarmachine_name As String 'local copy
Private mvarmachine_size As String 'local copy
Private mvarlstJob As Collection 'local copy
Public Property Set lstJob(ByVal vData As Collection)
'used when assigning an Object to the property, on the left side of a Set
statement.
'Syntax: Set x.lstJob = Form1
Set mvarlstJob = vData
end property

Public Property Get lstJob() As Collection
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.lstJob
Set lstJob = mvarlstJob
end property

Public Property Let machine_size(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.machine_size = 5
mvarmachine_size = vData
end property

Public Property Get machine_size() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machine_size
machine_size = mvarmachine_size
end property

Public Property Let machine_name(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.machine_name = 5
mvarmachine_name = vData
end property

Public Property Get machine_name() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machine_name

```

```

machine_name = mvarmachine_name
end property

```

```

Public Property Let machine_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.machine_code = 5
mvarmachine_code = vData
end property

```

```

Public Property Get machine_code() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machine_code
machine_code = mvarmachine_code
end property

```

```

Public Sub setValue(ByVal rs As ADODB.Recordset)
mvarmachine_code = rs.Fields!machine_no
mvarmachine_name = rs.Fields!machine_name
mvarmachine_size = rs.Fields!machine_size
end sub

```

```

cMachineContext - 1
Option Explicit
Public Function GetData(ByVal strSQL As String) As Collection
Const METHODNAME = "GetData"
On Error GoTo ErrHandler
Dim rs As ADODB.Recordset
Dim obj As cMachine
Dim vcollection As Collection
Set vcollection = New Collection
Set rs = New ADODB.Recordset
rs.CursorLocation = adUseClient
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
Do While Not rs.EOF
Set obj = New cMachine
obj.setValue rs
vcollection.Add obj
rs.MoveNext
Loop
rs.Close
rs.ActiveConnection = Nothing
Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
ErrHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

```

```

cMachineJob - 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarmachine_code As String 'local copy
Private mvarprod_code As String 'local copy

```



```
'local variable(s) to hold property value(s)
Private mvarannealing As String 'local copy
Private mvarrobot As String 'local copy
Private mvarman As String 'local copy
Public Property Let man(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.man = 5
mvarman = vData
end property
```

```
Public Property Get man() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.man
man = mvarman
end property
```

```
Public Property Let robot(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.robot = 5
mvarrobot = vData
end property
```

```
Public Property Get robot() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.robot
robot = mvarrobot
end property
```

```
Public Property Let annealing(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.annealing = 5
mvarannealing = vData
end property
```

```
Public Property Get annealing() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.annealing
annealing = mvarannealing
end property
```

```
Public Property Let prod_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.prod_code = 5
mvarprod_code = vData
end property
```

```
Public Property Get prod_code() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.prod_code
prod_code = mvarprod_code
end property
```

```
Public Property Let machine_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.machine_code = 5
```

```

mvarmachine_code = vData
end property

cMachineJob - 2
Public Property Get machine_code() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machine_code
machine_code = mvarmachine_code
end property

Public Sub setValue(ByVal rs As ADODB.Recordset)
mvarmachine_code = rs.Fields!machine_code
mvarprod_code = rs.Fields!prod_code
mvarannealing = rs.Fields!annealing
mvarrobot = rs.Fields!robot
mvarman = rs.Fields!man
end sub

cMachineJobContext - 1
Option Explicit
Public Function GetData(ByVal strSQL As String) As Collection
Const METHODNAME = "GetData"
On Error GoTo ErrHandler
Dim rs As ADODB.Recordset
Dim obj As cMachineJob
Dim vcollection As Collection
Set vcollection = New Collection
Set rs = New ADODB.Recordset
rs.CursorLocation = adUseClient
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
Do While Not rs.EOF
Set obj = New cMachineJob
obj.setValue rs
vcollection.Add obj
rs.MoveNext
Loop
rs.Close
rs.ActiveConnection = Nothing
Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
ErrHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

Public Function updateDB(ByVal strSQL As String) As Boolean
Const METHODNAME = "updateDb"
Dim result As Boolean
On Error GoTo ErrHandler
Dim cmd As New ADODB.Command
Dim lAffected As Long
cmd.ActiveConnection = db
cmd.CommandType = adCmdText

```

```

cmd.CommandText = strSQL
cmd.Execute !Affected
If !Affected = 0 Then
result = False
Else
result = True
End If
ExitMethod:
Set cmd = Nothing
updateDB = result
Exit Function
ErrorHandler:
result = False
Resume ExitMethod
end function

cMachineTime - 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarmachine_code As String 'local copy
Private mvarsuntime As Double 'local copy
Public Property Let sumtime(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.sumtime = 5
mvarsuntime = vData
end property

Public Property Get sumtime() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.sumtime
sumtime = mvarsuntime
end property

Public Property Let machine_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.machine_code = 5
mvarmachine_code = vData
end property

Public Property Get machine_code() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machine_code
machine_code = mvarmachine_code
end property

Public Sub setValue(ByVal rs As ADODB.Recordset)
mvarmachine_code = rs.Fields!machine_code
mvarsuntime = rs.Fields!sumtime
end sub

cMachineTimeContext - 1
Option Explicit
Public Function GetData(ByVal strSQL As String) As Collection
Const METHODNAME = "GetData"
On Error GoTo ErrorHandler

```

```

Dim rs As ADODB.Recordset
Dim obj As cMachineTime
Dim vcollection As Collection
Set vcollection = New Collection
Set rs = New ADODB.Recordset
rs.CursorLocation = adUseClient
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
Do While Not rs.EOF
Set obj = New cMachineTime
obj.setValue rs
vcollection.Add obj
rs.MoveNext
Loop
rs.Close
rs.ActiveConnection = Nothing
Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
ErrorHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

Public Function updateDB(ByVal strSQL As String) As Boolean
Const METHODNAME = "updateDb"
Dim result As Boolean
On Error GoTo ErrorHandler
Dim cmd As New ADODB.Command
Dim lAffected As Long
cmd.ActiveConnection = db
cmd.CommandType = adCmdText
cmd.CommandText = strSQL
cmd.Execute lAffected
If lAffected = 0 Then
result = False
Else
result = True
End If
ExitMethod:
Set cmd = Nothing
updateDB = result
Exit Function
ErrorHandler:
result = False
Resume ExitMethod
end function

cOrder - 1
Option Explicit
Private msOrderNo As String
Private msProductCode As String
Private msQuality As String
Private mdDueDate As Date
'local variable(s) to hold property value(s)

```

```

Private mvarcav As Double 'local copy
Private mvarct As Double 'local copy
Private mvarprocesstime As Double 'local copy
Private mvarphr As Double 'local copy
Public Property Let phr(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.phr = 5
mvarphr = vData
end property

```

```

Public Property Get phr() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.phr
phr = mvarphr
end property

```

```

Public Property Let processtime(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.processtime = 5
mvarprocesstime = vData
end property

```

```

Public Property Get processtime() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.processtime
processtime = mvarprocesstime
end property

```

```

Public Property Let ct(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.ct = 5
mvarct = vData
Debug.Print "ct was set to : " & vData
end property

```

```

Public Property Get ct() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.ct
ct = mvarct
Debug.Print "ct was read as : " & mvarct
end property

```

```

Public Property Let cav(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.cav = 5
mvarcav = vData
end property

```

```

Public Property Get cav() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.cav
cav = mvarcav
end property

```

```

Public Property Get orderno() As String
orderno = msOrderNo
end property

Public Property Let orderno(v As String)
msOrderNo = v
end property

Public Property Get productcode() As String
productcode = msProductCode
end property

Public Property Let productcode(v As String)
msProductCode = v
end property

Public Property Get Quality() As String
Quality = msQuality
end property

Public Property Let Quality(v As String)
msQuality = v
end property

Public Property Get duedate() As Date
duedate = mdDueDate
end property

Public Property Let duedate(v As Date)
mdDueDate = v
end property

Public Sub Initialize(OrdNo As String, OrdProdCode As String, OrdQty As String,
OrdDueDate As Date)
msOrderNo = OrdNo
msProductCode = OrdProdCode
msQuality = OrdQty
mdDueDate = OrdDueDate
end sub

Public Sub setValue(ByVal rs As ADODB.Recordset)
msOrderNo = rs.Fields!orderno
msProductCode = rs.Fields!prod_code
msQuality = rs.Fields!qty
duedate = rs.Fields!duedate
mvarcav = rs.Fields!cav
mvarct = rs.Fields!ct
end sub

cOrderContext - 1
Option Explicit
Public Function GetData(ByVal strSQL As String) As Collection
Const METHODNAME = "GetData"
On Error GoTo ErrHandler
Dim rs As ADODB.Recordset

```

```

Dim obj As cOrder
Dim vcollection As Collection
Set vcollection = New Collection
Set rs = New ADODB.Recordset
rs.CursorLocation = adUseClient
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
Do While Not rs.EOF
Set obj = New cOrder
obj.setValue rs
vcollection.Add obj
rs.MoveNext
Loop
rs.Close
rs.ActiveConnection = Nothing
Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
ErrorHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

cPermutation - 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarmachine_code As String 'local copy
Private mvarplanno As String 'local copy
Private mvaritemno As String 'local copy
Private mvarchg As Double 'local copy
Private mvarclean As Double 'local copy
Private mvaradj As Double 'local copy
Private mvarstartmanu As Date 'local copy
Private mvarendmanu As Date 'local copy
Private mvardeadline As Date 'local copy
Private mvardeliver As Date 'local copy
Private mvarlate As Double 'local copy
'local variable(s) to hold property value(s)
Private mvarprod_code As String 'local copy
Public Property Let prod_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.prod_code = 5
mvarprod_code = vData
end property

Public Property Get prod_code() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.prod_code
prod_code = mvarprod_code
end property

Public Property Let late(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.late = 5
mvarlate = vData

```

end property

Public Property Get late() As Double

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.late

late = mvarlate

end property

Public Property Let deliver(ByVal vData As Date)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.deliver = 5

mvardeliver = vData

end property

Public Property Get deliver() As Date

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.deliver

deliver = mvardeliver

end property

Public Property Let deadline(ByVal vData As Date)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.deadline = 5

mvardeadline = vData

end property

Public Property Get deadline() As Date

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.deadline

deadline = mvardeadline

end property

cPermutation - 2

Public Property Let endmanu(ByVal vData As Date)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.endmanu = 5

mvarendmanu = vData

end property

Public Property Get endmanu() As Date

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.endmanu

endmanu = mvarendmanu

end property

Public Property Let startmanu(ByVal vData As Date)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.startmanu = 5

mvarstartmanu = vData

end property

Public Property Get startmanu() As Date

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.startmanu

startmanu = mvarstartmanu

end property

Public Property Let adj(ByVal vData As Double)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.adj = 5

mvaradj = vData

end property

Public Property Get adj() As Double

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.adj

adj = mvaradj

end property

Public Property Let clean(ByVal vData As Double)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.clean = 5

mvarclean = vData

end property

Public Property Get clean() As Double

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.clean

clean = mvarclean

end property

Public Property Let chg(ByVal vData As Double)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.chg = 5

mvarchg = vData

end property

Public Property Get chg() As Double

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.chg

chg = mvarchg

end property

Public Property Let itemno(ByVal vData As String)

cPermutation - 3

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.itemno = 5

mvaritemno = vData

end property

Public Property Get itemno() As String

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.itemno

itemno = mvaritemno

end property

Public Property Let planno(ByVal vData As String)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.planno = 5

mvarplanno = vData

```
end property
```

```
Public Property Get planno() As String
```

```
'used when retrieving value of a property, on the right side of an assignment.
```

```
'Syntax: Debug.Print X.planno
```

```
planno = mvarplanno
```

```
end property
```

```
Public Property Let machine_code(ByVal vData As String)
```

```
'used when assigning a value to the property, on the left side of an assignment.
```

```
'Syntax: X.machine_code = 5
```

```
mvarmachine_code = vData
```

```
end property
```

```
Public Property Get machine_code() As String
```

```
'used when retrieving value of a property, on the right side of an assignment.
```

```
'Syntax: Debug.Print X.machine_code
```

```
machine_code = mvarmachine_code
```

```
end property
```

```
Public Sub setValue(ByVal rs As ADODB.Recordset)
```

```
mvaradj = rs.Fields!adj
```

```
mvarchg = rs.Fields!chg
```

```
mvarclean = rs.Fields!clean
```

```
mvardeadline = rs.Fields!deadline
```

```
mvardeliver = rs.Fields!deliver
```

```
mvarendmanu = rs.Fields!endmanu
```

```
mvaritemno = rs.Fields!itemno
```

```
mvarlate = rs.Fields!late
```

```
mvarmachine_code = rs.Fields!machine_code
```

```
mvarplanno = rs.Fields!planno
```

```
mvarprod_code = rs.Fields!prod_code
```

```
mvarstartmanu = rs.Fields!startmanu
```

```
end sub
```

```
cPermutationContext - 1
```

```
Option Explicit
```

```
Public Function GetData(ByVal strSQL As String) As Collection
```

```
Const METHODNAME = "GetData"
```

```
On Error GoTo ErrHandler
```

```
Dim rs As ADODB.Recordset
```

```
Dim obj As cPermutation
```

```
Dim vcollection As Collection
```

```
Set vcollection = New Collection
```

```
Set rs = New ADODB.Recordset
```

```
rs.CursorLocation = adUseClient
```

```
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
```

```
Do While Not rs.EOF
```

```
Set obj = New cPermutation
```

```
obj.setValue rs
```

```
vcollection.Add obj
```

```
rs.MoveNext
```

```
Loop
```

```
rs.Close
```

```
rs.ActiveConnection = Nothing
```

```

Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
ErrorHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

Public Function updateDB(ByVal strSQL As String) As Boolean
Const METHODNAME = "updateDb"
Dim result As Boolean
On Error GoTo ErrorHandler
Dim cmd As New ADODB.Command
Dim lAffected As Long
cmd.ActiveConnection = db
cmd.CommandType = adCmdText
cmd.CommandText = strSQL
cmd.Execute lAffected
If lAffected = 0 Then
result = False
Else
result = True
End If
ExitMethod:
Set cmd = Nothing
updateDB = result
Exit Function
ErrorHandler:
result = False
Resume ExitMethod
end function

cProduct - 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarprod_code As String 'local copy
Private mvarprod_name As String 'local copy
'local variable(s) to hold property value(s)
Private mvarcav As Double 'local copy
Private mvarct As Double 'local copy
Private mvarmachine_size As Double 'local copy
Private mvarannealing As String 'local copy
Private mvarman As String 'local copy
Private mvarrobot As String 'local copy
'local variable(s) to hold property value(s)
Private mvarcust_code As String 'local copy
Public Property Let cust_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.cust_code = 5
mvarcust_code = vData
end property

Public Property Get cust_code() As String
'used when retrieving value of a property, on the right side of an assignment.

```

```
'Syntax: Debug.Print X.cust_code
cust_code = mvarcust_code
end property
```

```
Public Property Let robot(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.robot = 5
mvarrobot = vData
end property
```

```
Public Property Get robot() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.robot
robot = mvarrobot
end property
```

```
Public Property Let man(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.man = 5
mvarman = vData
end property
```

```
Public Property Get man() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.man
man = mvarman
end property
```

```
Public Property Let annealing(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.annealing = 5
mvarannealing = vData
end property
```

```
Public Property Get annealing() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.annealing
annealing = mvarannealing
end property
```

```
Public Property Let machine_size(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
cProduct - 2
'Syntax: X.machine_size = 5
mvarmachine_size = vData
end property
```

```
Public Property Get machine_size() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machine_size
machine_size = mvarmachine_size
end property
```

```
Public Property Let ct(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
```

```
'Syntax: X.ct = 5
```

```
mvarct = vData
```

```
end property
```

```
Public Property Get ct() As Double
```

```
'used when retrieving value of a property, on the right side of an assignment.
```

```
'Syntax: Debug.Print X.ct
```

```
ct = mvarct
```

```
end property
```

```
Public Property Let cav(ByVal vData As Double)
```

```
'used when assigning a value to the property, on the left side of an assignment.
```

```
'Syntax: X.cav = 5
```

```
mvarcav = vData
```

```
end property
```

```
Public Property Get cav() As Double
```

```
'used when retrieving value of a property, on the right side of an assignment.
```

```
'Syntax: Debug.Print X.cav
```

```
cav = mvarcav
```

```
end property
```

```
Public Property Let prod_name(ByVal vData As String)
```

```
'used when assigning a value to the property, on the left side of an assignment.
```

```
'Syntax: X.prod_name = 5
```

```
mvarprod_name = vData
```

```
end property
```

```
Public Property Get prod_name() As String
```

```
'used when retrieving value of a property, on the right side of an assignment.
```

```
'Syntax: Debug.Print X.prod_name
```

```
prod_name = mvarprod_name
```

```
end property
```

```
Public Property Let prod_code(ByVal vData As String)
```

```
'used when assigning a value to the property, on the left side of an assignment.
```

```
'Syntax: X.prod_code = 5
```

```
mvarprod_code = vData
```

```
end property
```

```
Public Property Get prod_code() As String
```

```
'used when retrieving value of a property, on the right side of an assignment.
```

```
'Syntax: Debug.Print X.prod_code
```

```
prod_code = mvarprod_code
```

```
end property
```

```
Public Sub Initialize(productcode As String, productname As String)
```

```
mvarprod_code = productcode
```

```
mvarprod_name = productname
```

```
end sub
```

```
eProduct - 3
```

```
Public Sub setValue(ByVal rs As ADODB.Recordset)
```

```
mvarannealing = rs.Fields!annealing
```

```
mvarcav = rs.Fields!cav
```

```

mvarct = rs.Fields!ct
mvarcust_code = rs.Fields!cust_code
mvarmachine_size = rs.Fields!machine_size
mvarman = rs.Fields!man
mvarprod_code = rs.Fields!prod_code
mvarprod_name = rs.Fields!prod_name
mvarrobot = rs.Fields!robot
end sub

cProductContext - 1
Option Explicit
Public Function GetData(ByVal strSQL As String) As Collection
Const METHODNAME = "GetData"
On Error GoTo ErrHandler
Dim rs As ADODB.Recordset
Dim obj As cProduct
Dim vcollection As Collection
Set vcollection = New Collection
Set rs = New ADODB.Recordset
rs.CursorLocation = adUseClient
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
Do While Not rs.EOF
Set obj = New cProduct
obj.SetValue rs
vcollection.Add obj
rs.MoveNext
Loop
rs.Close
rs.ActiveConnection = Nothing
Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
ErrHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

cStart - 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarmachinecode As String 'local copy
Private mvarstartdate As Date 'local copy
Private mvarprod_code As String 'local copy
Public Property Let prod_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.prod_code = 5
mvarprod_code = vData
end property

Public Property Get prod_code() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.prod_code
prod_code = mvarprod_code
end property

```

```
Public Property Let startdatetime(ByVal vData As Date)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.startdatetime = 5
mvarstartdatetime = vData
end property
```

```
Public Property Get startdatetime() As Date
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.startdatetime
startdatetime = mvarstartdatetime
end property
```

```
Public Property Let machinecode(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.machinecode = 5
mvarmachinecode = vData
end property
```

```
Public Property Get machinecode() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machinecode
machinecode = mvarmachinecode
end property
```

```
Public Sub setValue(ByVal rs As ADODB.Recordset)
mvarmachinecode = rs.Fields!machineno
mvarprod_code = rs.Fields!prod_code
mvarstartdatetime = rs.Fields!startdate
end sub
```

```
cStartContext - 1
Option Explicit
Public Function GetData(ByVal strSQL As String) As Collection
Const METHODNAME = "GetData"
On Error GoTo ErrHandler
Dim rs As ADODB.Recordset
Dim obj As cStart
Dim vcollection As Collection
Set vcollection = New Collection
Set rs = New ADODB.Recordset
rs.CursorLocation = adUseClient
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
Do While Not rs.EOF
Set obj = New cStart
obj.setValue rs
vcollection.Add obj
rs.MoveNext
Loop
rs.Close
rs.ActiveConnection = Nothing
Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
```

```

ErrorHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

Public Function updateDB(ByVal strSQL As String) As Boolean
Const METHODNAME = "updateDb"
Dim result As Boolean
On Error GoTo ErrorHandler
Dim cmd As New ADODB.Command
Dim lAffected As Long
cmd.ActiveConnection = db
cmd.CommandType = adCmdText
cmd.CommandText = strSQL
cmd.Execute lAffected
If lAffected = 0 Then
result = False
Else
result = True
End If
ExitMethod:
Set cmd = Nothing
updateDB = result
Exit Function
ErrorHandler:
result = False
Resume ExitMethod
end function

cTmpTransact = 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarorderno As String 'local copy
Private mvarprod_code As String 'local copy
Private mvarqty As String 'local copy
Private mvarduedate As Date 'local copy
Private mvarprocesstime As Double 'local copy
Private mvarcav As Double 'local copy
Private mvarct As Double 'local copy
Private mvarcust_priority As String 'local copy
Private mvarcust_code As String 'local copy
'local variable(s) to hold property value(s)
Private mvarmachine_size As String 'local copy
Public Property Let machine_size(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.machine_size = 5
mvarmachine_size = vData
end property

Public Property Get machine_size() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.machine_size
machine_size = mvarmachine_size
end property

```



```
Public Property Let cust_code(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.cust_code = 5
mvarcust_code = vData
end property
```

```
Public Property Get cust_code() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.cust_code
cust_code = mvarcust_code
end property
```

```
Public Property Let cust_priority(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.cust_priority = 5
mvarcust_priority = vData
end property
```

```
Public Property Get cust_priority() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.cust_priority
cust_priority = mvarcust_priority
end property
```

```
Public Property Let ct(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.ct = 5
mvarct = vData
end property
```

```
Public Property Get ct() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.ct
ct = mvarct
end property
```

```
Public Property Let cav(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
cTmpTransact = 2
'Syntax: X.cav = 5
mvarcav = vData
end property
```

```
Public Property Get cav() As Double
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.cav
cav = mvarcav
end property
```

```
Public Property Let processtime(ByVal vData As Double)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.processtime = 5
mvarprocesstime = vData
end property
```

Public Property Get processtime() As Double

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.processtime

processtime = mvarprocesstime

end property

Public Property Let duedate(ByVal vData As Date)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.duedate = 5

mvarduedate = vData

end property

Public Property Get duedate() As Date

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.duedate

duedate = mvarduedate

end property

Public Property Let qty(ByVal vData As String)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.qty = 5

mvarqty = vData

end property

Public Property Get qty() As String

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.qty

qty = mvarqty

end property

Public Property Let prod_code(ByVal vData As String)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.prod_code = 5

mvarprod_code = vData

end property

Public Property Get prod_code() As String

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.prod_code

prod_code = mvarprod_code

end property

Public Property Let orderno(ByVal vData As String)

'used when assigning a value to the property, on the left side of an assignment.

'Syntax: X.orderno = 5

cTmpTransact - 3

mvarorderno = vData

end property

Public Property Get orderno() As String

'used when retrieving value of a property, on the right side of an assignment.

'Syntax: Debug.Print X.orderno

orderno = mvarorderno

end property

```

Public Sub setValue(ByVal rs As ADODB.Recordset)
mvarcav = rs.Fields!cav
mvarct = rs.Fields!ct
mvarcust_code = rs.Fields!cust_code
mvarcust_priority = rs.Fields!cust_priority
mvarduedate = rs.Fields!duedate
mvarmachine_size = rs.Fields!machine_size
mvarorderno = rs.Fields!orderno
mvarprocesstime = rs.Fields!processtime
mvarprod_code = rs.Fields!prod_code
mvarqty = rs.Fields!qty
end sub

cTmpTransactContext - 1
Option Explicit
Public Function GetData(ByVal strSQL As String) As Collection
Const METHODNAME = "GetData"
On Error GoTo ErrHandler
Dim rs As ADODB.Recordset
Dim obj As cTmpTransact
Dim vcollection As Collection
Set vcollection = New Collection
Set rs = New ADODB.Recordset
rs.CursorLocation = adUseClient
rs.Open strSQL, db, adOpenStatic, adLockReadOnly
Do While Not rs.EOF
Set obj = New cTmpTransact
obj.setValue rs
vcollection.Add obj
rs.MoveNext
Loop
rs.Close
rs.ActiveConnection = Nothing
Set GetData = vcollection
getDataExit:
Set rs = Nothing
Exit Function
ErrHandler:
ShowError "getdata", "getdata"
Resume getDataExit
end function

Public Function updateDB(ByVal strSQL As String) As Boolean
Const METHODNAME = "updateDb"
Dim result As Boolean
On Error GoTo ErrHandler
Dim cmd As New ADODB.Command
Dim lAffected As Long
cmd.ActiveConnection = db
cmd.CommandType = adCmdText
cmd.CommandText = strSQL
cmd.Execute lAffected
If lAffected = 0 Then
result = False
Else

```

```

result = True
End If
ExitMethod:
Set cmd = Nothing
updateDB = result
Exit Function
ErrorHandler:
result = False
Resume ExitMethod
end function

IniUtil - 1
Option Explicit
Private Declare Function GetPrivateProfileSection Lib "kernel32" Alias
"GetPrivateProfileSectionA"
(ByVal lpAppName As String, ByVal lpReturnedString As String, ByVal nSize As
Long, ByVal lpFileName As String) As Long
Private Declare Function GetPrivateProfileString Lib "kernel32" Alias
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As
String, ByVal lpReturn
edString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long
Private Declare Function WritePrivateProfileSection Lib "kernel32" Alias
"WritePrivateProfileSectionA" (ByVal lpAppName As String, ByVal lpString As String, ByVal lpFileName As String) As Long
Private Declare Function WritePrivateProfileString Lib "kernel32" Alias
"WritePrivateProfileStringA
" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As Long
' Function Name: GetProfileItem
' Parameters: strSectionName - section name in the ini file
' strKeyName - key name within the section
' strFileName - ini file name
' strValue - return value
' Return: nonzero - success
' zero - value is empty or an error encountered
' Description: This function retrieve the string from the specified section
' in the initialization file
Public Function GetProfileItem(ByVal strSectionName As String, ByVal strKeyName As String, ByVal strFileName As String, strValue As String) As
Integer
Dim success As Long
Dim nSize As Long
Dim ret As String
ret = Space$(2048)
nSize = 2048
success = GetPrivateProfileString(strSectionName, strKeyName, "", ret, nSize, strFileName)
If success Then
strValue = Left$(ret, success)
GetProfileItem = success
Else
strValue = ""
GetProfileItem = 0
End If
end function

' Function Name: SaveProfileItem
' Parameters: strSectionName - section name in the ini file
' strKeyName - key name within the section

```

```

' strFileName - ini file name
' strValue - value to be written
' Return: nonzero - success
' zero - value is empty or an error encountered
' Description: This function save the string to the specified section
' in the initialization file
Public Function SaveProfileItem(ByVal strSectionName As String, _ByVal strKeyName As String, _ByVal strFileName As String, _
ByVal strValue As String) As Integer
Dim rtnValue As Long
rtnValue = WritePrivateProfileString(strSectionName, strKeyName, strValue, _strFileName)
If rtnValue Then SaveProfileItem = rtnValue
Else
SaveProfileItem = 0
End If
end function

OLogger - 1
Option Explicit
' Description :
' Port from Log4j to Visual Basic.
' History
' 2002-01-18 Piek Created
Public Enum LogLevel
llAll = 0
llDebug = 1
llInfo = 2
llWarn = 4
llError = 8
llFatal = 16
llOff = 32
End Enum
Private mvarLevel As LogLevel
Private fs As FileSystemObject
Private ts As TextStream
' 2002-02-22
Private bVBIDE As Boolean
Private Const MODULENAME = "OLogger"
Private Sub Class_Initialize()
mvarLevel = llAll
Set fs = New FileSystemObject
' comment out for make lazy initialize
'OpenTextStream
bVBIDE = isVBIDERunning()
end sub

' 2002-02-10 Created
Private Sub Class_Terminate()
CloseTextStream
Set fs = Nothing
end sub

' 2002-02-10 Created the routine for make in lock-up loop exit easy,
' loop must use log and in case of lock-up occurred,
' just rename log file, stop will execute.

```

```

' By open log just only when logging occurred, other that
' log file is closed.
'
' 2002-02-22 Create a new log file when not in debug
'
Private Sub OpenTextStream()
On Error GoTo ErrHandler
' make sure that exists file handle (if has) close properly
CloseTextStream
' open the existing log file
Set ts = fs.OpenTextFile(App.Path + "\log.txt", ForAppending, False, TristateFalse)
Exit Sub
ErrHandler:
' 2002-02-22
If Err.Number = 53 Then
If bVBIDE Then
'Stop
Else
' open in append mode
Set ts = fs.OpenTextFile(App.Path + "\log.txt", ForAppending, True, TristateFalse)
Resume Next
End If
Else
Debug.Print Err.Number & " " & Err.Description
End If
end sub

OLogger - 2
' 2002-02-10 Created
Private Sub CloseTextStream()
If Not (ts Is Nothing) Then
ts.Close
Set ts = Nothing
End If
end sub

Public Property Let level(ByVal vData As String)
mvarLevel = vData
end property

Public Property Get level() As String
level = mvarLevel
end property

Public Sub assertLog(ByVal assertion As Boolean, ByVal strModule As String, ByVal strMethod As String, ByVal strMessage As String)
If assertion Then errorLog strModule, strMethod, strMessage
End If
end sub

Public Sub fetalLog(ByVal strModule As String, ByVal strMethod As String, ByVal strMessage As String)
log llFetal, strModule, strMethod, strMessage
end sub

Public Sub errorLog(ByVal strModule As String, ByVal strMethod As String, ByVal strMessage As String)
log llError, strModule, strMethod, strMessage

```

```

end sub

Public Sub debugLog(ByVal strModule As String, ByVal strMethod As String, ByVal strMessage As String)
log l!Debug, strModule, strMethod, strMessage
end sub

Public Sub infoLog(ByVal strModule As String, ByVal strMethod As String, ByVal strMessage As String)
log l!Info, strModule, strMethod, strMessage
end sub

Public Sub warnLog(ByVal strModule As String, ByVal strMethod As String, ByVal strMessage As String)
log l!Warn, strModule, strMethod, strMessage
end sub

Private Sub log(ByVal level As LogLevel, ByVal strModule As String, ByVal strMethod As String, ByVal strMessage As String)
Dim str As String
OpenTextStream
If mvarLevel <= level Then str = _DateToYYYYMMDDHHMMSS(Now()), False, True) & _
' [' & LevelToString(level) & '] * & _strModule & "." & strMethod & " > " & strMessage 'ts.WriteLine str
End If
CloseTextStream
end sub

Private Function LevelToString(ByVal level As LogLevel) As String
If level = l!Debug Then LevelToString = "DEBUG"
ElseIf level = l!Error Then LevelToString = "ERROR"
ElseIf level = l!Fetal Then LevelToString = "FETAL"
ElseIf level = l!Info Then LevelToString = "INFO"
O!Logger - 3
ElseIf level = l!Warn Then
LevelToString = "WARN"
End If
end function

Public Function StringToLevel(ByVal level As String, Optional ByVal def As LogLevel = l!Debug) As LogLevel
Dim result As LogLevel
If level = "DEBUG" Then result = l!Debug
ElseIf level = "ERROR" Then result = l!Error
ElseIf level = "FETAL" Then result = l!Fetal
ElseIf level = "INFO" Then result = l!Info
ElseIf level = "WARN" Then result = l!Warn
Else
result = def
End If
StringToLevel = result
end function

Public Function isEnabledFor(ByVal level As LogLevel) As Boolean isEnabledFor = (mvarLevel <= level)
end function

Public Function isDebugEnabled() As Boolean isDebugEnabled = (mvarLevel <= l!Debug)
end function

Public Function isInfoEnabled() As Boolean isInfoEnabled = (mvarLevel <= l!Info)
end function

```

```

Public Function readConfig(ByVal str As String) As Integer Const METHODNAME = 'readConfig'
Dim strLevel As String
With mIniUtil
.GetProfileItem "Log", "Threshold", strInitFullPath, strLevel
End With
' if can't read file, or the specific section was not found
' default level is debug
If Len(strLevel) <> 0 Then mvarLevel = StringToLevel(strLevel, llDebug)
Else mvarLevel = llDebug
End If
end function

SysTabContext - 1
Option Explicit
Dim rs As ADODB.Recordset
Public Function getKeyValue(ByVal strkey As String) As String
Dim strBasicSQL As String
If Not (rs Is Nothing) Then rs.Close
Set rs = Nothing
End If
Set rs = New ADODB.Recordset
strBasicSQL = "select keyvalue, value from systab where " & _
"keyvalue = " & strkey & "" rs.Open strBasicSQL, db
If Not rs.EOF Then
Dim keyvalue As String
keyvalue = rs.Fields!keyvalue
Dim value As String
value = rs.Fields!value
End If
getKeyValue = value
end function

SystabObj - 1
Option Explicit
'local variable(s) to hold property value(s)
Private mvarKey As String 'local copy
Private mvarKeyvalue As String 'local copy
Public Property Let keyvalue(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Keyvalue = 5
mvarKeyvalue = vData
end property
Public Property Get keyvalue() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Keyvalue
keyvalue = mvarKeyvalue
end property
Public Property Let Key(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Key = 5
mvarKey = vData
end property
Public Property Get Key() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Key
Key = mvarKey
end property

```


ประวัติผู้เขียนวิทยานิพนธ์

นางสาวชนกนันท์ ปัตตะโชติ เกิดเมื่อวันที่ 21 พฤศจิกายน 2518 ที่จังหวัดสงขลา เข้ารับการศึกษาระดับประถมศึกษาและระดับมัธยมศึกษาที่โรงเรียนธิดานุเคราะห์ จังหวัดสงขลา สำเร็จการศึกษาในระดับปริญญาตรี สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ เมื่อปี พ.ศ. 2541 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์ มหาบัณฑิต ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี การศึกษา 2542

