JobGen Plus の学習

本書では以下を学習します:

- 簡単な JobGen Plus アプリケーションの作成方法
- C 言語で複雑な JobGen Plus アプリケーションを作成する方法

					目	次			
第1章	簡単な	JOBGEN	PLUS	アプリ	ケーショ	ン	 	 	 4
資産管理	!						 	 	 4
第2章	複雑な	JOBGEN	PLUS	アプリ	ケーショ	ン	 	 	 23
在庫管理	!						 	 	 23

### 第1章 簡単な JobGen Plus アプリケーション

#### 資産管理

このアプリケーションは各地にある現在の資産情報の収集を行うために設計されました。 それぞれの資産は資産番号で表されます。ユーザはこのアプリケーションの入ったポー タブル・データコレクタを持って各地にある資産番号を収集します。ユーザがこの JobGen Plus アプリケーションを設計する前に以下を決定する必要があります。

- データのタイプ ポータブル・データコレクタの各データファイルは、複数のデータ レコードで構成されています。各データレコードは同じデータフィールド の番号を含んでいます。二つのデータフィールドが資産管理の場所と資産 番号に必要です。ユーザは各データフィールドのデータタイプを決めなけ ればなりません。
- プロセス ユーザは実行するために必要なプロセスの概略をまとめなければなりません。資産管理のアプリケーションは5つの別なプロセスが必要です。
  - 1. 場所の情報を収集する。
  - 2. 場所の情報を検証する。
  - 3. 資産番号を収集する。
  - 4. 資産番号を検証する。
  - 5. データを編集する。
- 作業のフロー 資産管理の作業は以下の方法で行われます。
  - 1. 場所の情報を収集します。あるいはポータブルターミナルに収集され たデータを編集する場合は "F2"キーを押します。
  - 2. 場所の情報を検証します。
  - 場所の情報が指定された要求をパスした場合、ステップ 4 へ進みます。
     そうでなければステップ1に進みます。
  - 4. 資産番号を収集します。
  - 5. "F1" キーを押した場合、ステップ1へ進みます。そうでなければステ ップ6へ進みます。このステップは新しい場所の情報を入力するため に"F1"キーを押すことができます。

- 6. 資産番号を検証します。
- 7. 資産番号が指定した要求をパスした場合、ステップ4へ進みます。このサンプルでは、それ以外もステップ4へ進みます。

### 資産管理の作成

上記のリストに従って、資産管理のアプリケーションは二つのデータフィールドについ て二つの収集(collect)ノード、そして収集したデータの表示と編集のために編集(Edit)ノ ードが必要です。また、いくつかのリンクがノード間の移動を実現するために必要で す。以下のステップは資産管理のアプリケーションをどのようにして作成するかを示し ています。

**ヒント:**日本語メニューを作成する場合は、上記の作業をする前に Edit メニューから Define Portable を選択し、Setting > Screen > Large Font をチェックして下さい。

Define Portable					×
Model Setting Keymap	Screen Scanner Type: Auto Buzzer Volume: C High	<ul> <li>C Pen</li> <li>C Medium</li> </ul>	C Low	© Off	]
	OK	Cancel		Help	

 マウスポインタを Collect ノード ボタン 描 に移動して下さい。マウスポインタが Collect ノード ボタン 描 上にある時に左マウスボタンをクリックして下さい。そし てマウスポインタを Collect ノード が置かれる場所に移動して下さい。

Sila Edit Nadaa Maru Puild Ta	ala Window Data		<u>×</u>
	al 🖴 Lee Lev L		
	<u>□   -                                  </u>		
	<mark>∦?</mark> Ē   ₩₩	🍳 🛍 🐹 📝 🔳	
😤 Job2			<u>_                                    </u>
<ul> <li>Portable</li> <li>Records</li> <li>Comment</li> <li>Excluded</li> <li>Nodes, Menu</li> <li>Nodes, Collect</li> <li>Nodes, Edit</li> <li>Nodes, Edit</li> <li>Nodes, Erase</li> <li>Nodes, Upload</li> <li>Nodes, Run-Job</li> <li>Links</li> <li>Links, Key Conditions</li> <li>Links, Other Conditions</li> </ul>		collect1	
	•	· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •
For Help, press F1			

左マウスボタンをダブルクリックして下さい。 あるいは、ノードを置きたい場所で右クリックすると、コンテキストメニューがポ ップアップします。作成するために Collect ノード を選択して下さい。



# マウスボタンが Collect ノード にある間に左マウスボタンをダブルクリックして下 さい。 "Define Node [Collect]" ウインドウがスクリーンに現れます。

Define Node [Collect] - Location		×
Basic A	This is a collect node.	
⊡-Screen …Position ⊡-Input	Name: Location	
Initial Value Device Barcode Lookup File Formula	☑ Start Node: Job will start running from this node	
Lookup File Qualifier Function Keys Links, in	Color: Node 💌	
	┌─ * Exclude This Node When Make Job	
	OK Cancel Help	

以下のオプションを変更します。

- (a) Basic プロパティで、 "collect1" から "Location"に名前を変更します。
- (b) Start Node をチェックします。このノードは資産管理アプリケーションの始ま りをマークします。
- (c) Screen > position プロパティで、"Clear Screen From Specified Location To The End Of Screen"(指定した場所からスクリーンの最後までをクリアする) オプションを選択し、 "0" と "0" を row (行)と column(列)のボックスにタイプします。このオプションはターゲットデバイスのスクリーンを行 0 / 列 0 から最後の行/列まですべてクリアします。

Define Node [Collect] - Location		X
Basic Comment Screen Input Initial Value Device Barcode Lookup File Verify Lookup File Qualifier Function Keys Links, in <- Asset_Number, #2 <- edit, #4 Lockup #1	<ul> <li>Display Prompt:</li> <li>Print Following The Last Print, Scroll Up If Necessary</li> <li>Print Directly To Screen</li> <li>Clear Screen From Specified Location To The End Of Screen Row (0.3):</li> <li>Column (0.11):</li> <li>Do Not Display Position Mark Position Mark:</li> <li>Display Input:</li> <li>At Current Position</li> <li>At Specified Location: Row (0.3):</li> <li>Column (0.11):</li> <li>As Password (Print * For Each Input)</li> <li>Do Not Display Prompt And Input For Auto Generated Data</li> </ul>	
	OK Cancel Help	

- (d) "At Specified Location" オプションを選択して、Row と Column のボックス に"1"と"0"をタイプします。 このオプションは行1と列0の位置から入力デー タ(ユーザによって入力される)を表示します。
- 2. Screen プロパティを選択します。スクリーンのスペースに "所在地:" をタイプしま す。

Define Node [Collect] - Location															×
Comment Screen		Scre	en: r	ows	: 12	2, co	lum	ns:	4				1		
⊡…Input Initial Value Device		Ē	[	在		١	ļ	•							
Barcode Barcode		•							0						
l ⊡ Formula ⊡ Verify Lookup File		。													
Qualifier Function Keys		。													
		I —													
i⊟-Links, out > asset, #2 -> moiomonu, #5															
	•														
					0	ĸ				Can	icel		Help		

3. Input プロパティを選択して下さい。 "Define Collect" ウインドウは以下のように変わります。

Define Node [Collect] - Location		×
Basic Comment Screen Position Input Nevice Barcode Cookup File Formula Overify Cookup File Cookup File Lookup File Lookup File Lookup File Lookup File Lookup File Lookup File Lookup Sile Lookup	Data Type:         ○ Numerical, Integer         ○ Numerical, Floating-Point         ○ Character String         ✓ Set Data As Same As Last Record, And Complete Input:         □ Fill Tail With White Spaces         □ Allow Empty Data	
	OK Cancel Help	

- 4. **"Set Data As Same As Last Record, And Complete Input"**をチェックします。これは "所在地"情報を一回だけ入力すればようようにし、そして"F1"キーを押すまで"資 産番号"の入力を続けます。
- 5. Verify プロパティを選択して下さい。 "Define Collect" ウインドウは以下のように変わります:

Define Node [Collect] - Location		×
Basic Comment Screen Position Input Device Barcode Lookup File Formula Verity Lookup File Qualifier Function Keys Links, in Links, out	Length:   From:   1   Range:   From:   To:   Picture: Do Not Force To Re-Input If Not Passed Verification Do Not Beep If Not Passed Verification Do Not Print Error Sign (Character * ) If Not Passed Verification	
	OK Cancel Help	

6. OK をクリックして下さい。

次のステップは二番目の Collect ノード を作ります。

7. マウスカーソルを Collect ノード を置く場所に移動します。 左マウスボタンをダブ ルクリックします。

Image: Second secon	🍣 JobGen Lite - Job2 File Edit Nodes View Build Too	ols Window Help		<u>- 0 ×</u>
Image: Constraint of the constraint o		i 🗇 🐟 メ 🗖	🗖 🖂 🞝 🔒 💂 🍍	
Image: Job2       Image: Job2         Image: Portable       Image: Portable         Image: Records       Image: Comment         Image: Comment       Image: Comment         Image: Excluded       Image: Comment         Image: Nodes, Menu       Image: Comment         Image: Nodes, Menu       Image: Comment         Image: Nodes, Collect       Image: Comment         Image: Nodes, Collect       Image: Comment         Image: Nodes, Rese       Image: Collect         Image: Node       Image: Collec		<mark>//?</mark> 🗉 ╫ 📎	001 💌 🖅 🔳 🖫 🍾	
Portable Records Comment Excluded Nodes, Collect Nodes, Erase Nodes, Program Nodes, Run-Job Links, Key Conditions Links, Other Conditions	😤 Job2			
	<ul> <li>Job</li> <li>Portable</li> <li>Records</li> <li>Comment</li> <li>Excluded</li> <li>Nodes, Menu</li> <li>Nodes, Collect</li> <li>Nodes, Edit</li> <li>Nodes, Edit</li> <li>Nodes, Upload</li> <li>Nodes, Run-Job</li> <li>Links</li> <li>Links, Key Conditions</li> <li>Links, Other Conditions</li> </ul>		Location Collect2	

- 8. マウスポインタが新しく作った Collect ノード にある間に左マウスボタンをダブル クリックして下さい。 "Define Collect" ウインドウがスクリーンに現れます。
- 9. 以下のオプションを変更します:
  - (a) "collect2"の名前を "Asset\_Number"に変更します。
  - (b) Screen > Position プロパティで、 "Clear Screen From Specified Location To The End Of Screen"を選択して、 "1" と "0" を入力します。 このオプションはター ゲットデバイスのスクリーンを行 1 / 列 0 から最後の行と欄まですべてクリア します。
  - (c) "At Specified Location"を選択して、"1"と"0"をタイプします。このオプションは、行1と列0の位置から入力データ(ユーザ入力データ)を表示します。
  - (d) Input > Initial Value のチェックは両方とも付けません。
- 10. Screen プロパティを選択します。 "Define collect" ウインドウは以下のように変わり ます。スクリーンのスペースに "資産#入力:" をタイプして下さい。

Define Node [Collect] - collect2																×
Basic Comment Screen Position	Scree	en: r Ç	ows 產	: 12	. co #	lumi 入	ns: 4	₁ לב	ſ	•	0	0				
	<u>ہ</u>										0					
Laookup File Formula	<u>ہ</u>															
		0	<u> </u>	<u>ہ</u>	<u> </u>		<u> </u>	<u>ہ</u>	<u>ہ</u>		0					
				0	(			1	Can	cel			Help	p		

- 11. OK をクリックして下さい。 次のステップは Edit ノード を作ります。
- 12. マウスポインタを Edit ノード ボタン<sup>回</sup>に移動します。マウスポインタが Edit ノー ド ボタン<sup>回</sup>にある間に左マウスボタンをクリックします。次に、マウスポインタ を Edit ノード を置く場所に移動します。

🍣 JobGen Lite - Job2	,	×
Eile Edit Nodes View Build Loo	als window Help	
		a
terson ⊕-Job ⊕-Portable		
Nodes, Menu ⊞-Nodes, Collect	edit3	
Nodes, Hath →-Nodes, Edit Nodes, Erase		
Nodes, Upload Nodes, Program Nodes, Run-Job		
Links Links, Key Conditions	Asset	
	• • • • • • • • • • • • • • • • • • •	
For Help, press F1		

左マウスボタンをダブルクリックします。

13. マウスポインタが Edit ノード にある間に左マウスボタンをダブルクリックします。 "Define Edit" ウインドウがスクリーンに現れます。

Define Node [Edit] - edit				×
Basic Comment Function Keys ⊕ Links, in ⊕ Links, out	Record Name: Data File Name: Field Separator:	ASSETMGT.TXT , (Comma)		
	ОК	Cancel	Help	

14. 以下のオプションを変更します。

- (a) 名前を "Edit3" から "Edit"に変更します。
- (b) Data File Name オプションとして ASSETMGT.TXT をタイプします。
- (c) Field Separator について", [Comma]"を選択します。

15. OK をクリックします。

次のステップは最初の Collect ノード から二番目の Collect ノード へのリンク を作 ります。

- 16. マウスポインタをリンク ボタン に移動します。 マウスポインタがリンク ボタン にある間に左マウスボタンをクリックします。 そしてマウスポインタを "Location" ノード に移動します。左マウスボタンをクリックしたままにします。マウスポイン タを "Asset\_Number" ノード に移動し、左マウスボタンを離します。
- 17. マウスポインタを新しい作成したリンクに移動し、そして、この リンク について の情報が現れます。リンク の標準設定は "On Always"です。 これは "Location" ノー ド がデータ収集を終了したら、 "Asset\_Number" ノード に進むことを示しています。

🍣 JobGen Lite - Job2		
<u>File Edit N</u> odes <u>V</u> iew <u>B</u> uild <u>T</u> oo	pols <u>W</u> indow <u>H</u> elp	
🗅 🖻 🔒 🚺 🕺 🖻 🖡	8 🗗 < 😹 🖪 🗖 🖉 🖧 💂 🎁	
	<mark>112</mark> E HH 🖲 💷 🗹 🔳 🚼 💫	
😤 Job2		
<ul> <li>Dob</li> <li>Portable</li> <li>Records</li> <li>Comment</li> <li>Excluded</li> <li>Nodes, Menu</li> <li>Nodes, Collect</li> <li>Nodes, Edit</li> <li>Nodes, Edit</li> <li>Nodes, Upload</li> <li>Nodes, Run-Job</li> <li>Links</li> <li>Links, Key Conditions</li> <li>Links, Other Conditions</li> </ul>	edit Location Asset	

- 18. 次のステップは "Asset\_Number" ノード から "Location" ノードへの別なリンクを作 ります。
- 19. マウスポインタを "Asset\_Number" ノード に移動します。左マウスボタンをクリッ クしたままにします。マウスポインタを "Location" ノード に移動し、左マウスボタ ンを離します。

W JobGen Lite - Job2	als Wündow Help	<u></u>
	B 🗇 🔦 メ 🗖	🖂 😼 🗛 💂 🍵
	<mark>//?</mark> 🗉 🛲 🖏 🐽 (	🕺 🗊 🔳 🖫 🐾
📆 Job2		
<ul> <li>Job</li> <li>Portable</li> <li>Records</li> <li>Comment</li> <li>Excluded</li> <li>Nodes, Menu</li> <li>Nodes, Collect</li> <li>Nodes, Edit</li> <li>Nodes, Edit</li> <li>Nodes, Erase</li> <li>Nodes, Program</li> <li>Nodes, Run-Job</li> <li>Links</li> <li>Links, Key Conditions</li> <li>Links, Other Conditions</li> </ul>	edit Loc	cat ion

- 20. 新しく作ったリンク上にマウスポインタがある間に左のマウスボタンをダブルクリ ックします。 "Define Node" ウインドウがスクリーンに現れます。
- 21. 以下のオプションを変更します。
  - (a) **Keypad Input** をチェックします。 そしてマウスを空白の行に移動し、左マウ スボタンをクリックします。 "PT-600 Keypad" ウインドウがスクリーンに現れ ます。

マウスポインタを F1 キーに移動し、左マウスボタンをクリックします。OK をクリックします。この選択はユーザがポータブル・データコレクタで"F1"キ ーを押した場合、"Location" ノード が直ちに処理されることを示しています。



- (b) "RRLI, Reset 'Repeat Last Input' For All Nodes"をチェックして下さい。この 選択は"Location" ノードの "Set Data As Same As Last Record, And Complete Input" オプションに関係します。RRLI がチェックされているこのリンクがア クティブになったら、 "Location" ノード は新しいデータをユーザが入力する必 要があります。
- 22. **OK** をクリックします。 次のステップは、最初の Collect ノード から Edit ノード へのリンク を作成します。
- 23. マウスポインタを "Location" ノード に移動します。 左マウスボタンをクリックし たままにします。マウスポインタを "Edit" ノード に移動し、左マウスボタンを離し ます。
- 24. マウスポインタを新しく作成したリンクに移動し、そしてこのリンクについての情報がスクリーンに現れます。

Job2	
Uob2	
Demisiple	
Portable     Records     Comment     Excluded     Nodes, Menu     Nodes, Collect     Nodes, Edit     Nodes, Edit     Nodes, Erase     Nodes, Program     Nodes, Run-Job     Links     Links, Key Conditions     Links, Other Conditions	edit On Asset

25. 以下のオプションを変更します:

- (a) **Keypad Input** をチェックします。 そしてマウスを空白の行に移動して左マウ スボタンをクリックします。"PT-600 Keypad" ウインドウがスクリーンに現れ ます (前のリンクと全く同じ)。
- (b) カーソルポインタを F2 キーへ移動して、左マウスボタンをクリックします。
   OK をクリックします。この選択はユーザがポータブル・データコレクタの"F2"キーを押した場合、 "Edit" ノード が直ちに処理されます。次のステップは Edit ノード から最初の Collect ノード へのリンク を作ります。
- 26. マウスポインタを "Edit" ノード へ移動します。 左マウスボタンをクリックしたま まにします。マウスポインタを"Location" ノードへ移動して、左のマウスボタンを 離します。

🍣 JobGen Lite - Job2	
	<mark>//?</mark> 🗉 🛲 🚱 💷 🖾 🔛 💽
🕎 Job2	
<ul> <li>Job</li> <li>Portable</li> <li>Records</li> <li>Comment</li> <li>Excluded</li> <li>Nodes, Menu</li> <li>Nodes, Collect</li> <li>Nodes, Edit</li> <li>Nodes, Edit</li> <li>Nodes, Program</li> <li>Nodes, Run-Job</li> <li>Links, Key Conditions</li> <li>Links, Other Conditions</li> </ul>	edit Location

27. マウスポインタを新しく作成した リンク に移動し、そしてこの リンク についての 情報がスクリーンに現れます。 リンク の標準設定は "On Always"です。 これは "Edit" ノード が終わった場合、 "Location" ノード が処理されることを示しています。

アプリケーションを終了する前に、 ジョブ はまず保存されなければなりません。マ ウスポインタをメニューバーの Build > Make Job に移動します。マウスポインタが Make Job にある間に左マウスボタンをクリックします。そしてマウスポインタを File > Save As...に移動して、左マウスボタンをクリックします。

28. ジョブ のファイル名として、**File Name** スペース中に"assetmgt.job"を入力します。 OK をクリックします。

🕎 Job2	名前を付けて保存	<mark>?</mark> ]	× L – D ×
. Job ⊕…Portable	保存する場所①:	🔁 PT600 🔽 🖛 🗈 📸 🎫	
Records Comment Excluded Nodes, Mer ⊕-Nodes, Col Nodes, Edi Nodes, Era Nodes, Upl Nodes, Upl	barcode bisech Chinese collect collect2 count	Proreign     Prunjob1     Prespanish       Protkey     Prunjob2       Protkey     Prunjob3       Protkupb     Protkey       Protkupb     Protkey <td></td>	
Nodes, Pro Nodes, Rur	ファイル名(N):	assetmet.job 保存( <u>S</u> )	
⊕ Links, Key Links, Ott	ファイルの種類(工):	JobGen Lite Files (*.jgp) 💌 キャンセル	

ジョブを保存した後、最後のステップはデータレコードの最後をマークする最後の リンクを別にすることです。

29. 新しいリンクの相手先として "Asset\_Number" ノード を選択するループしたリンク を作成します。ノードメニューでリンクを選択するか、ツールバーのリンクアイコ ン<sup>™</sup>をクリックします。マウスポインタを "Asset\_Number" ノード に移動して、左 マウスボタンをクリックしたまま、ノードの中を少し動かして左マウスボタンを離 します。

JobGen Plus	×
?	Dreate a self loop link?
(#CNY)	いいえ( <u>N</u> )

はいをクリックします。

30. Edit > New Record で新しいレコードを作成し、次に Edit > Define Record でレコード を定義します。

Define Record			×
ASSETMGI     Separator     EOR Links     Comment	Record Name: AS Data File Name: C3 This Is The Default Data Nodes:	SSETMGT  #JGPLUS¥SAMPLES¥ASSETMGT.TXT Record Or Data File  Record Fields:  Asset_Number Locat ion	-
	ОК	Cancel Help	_

Record Name: を ASSETMGT とします。

Data Node の Asset\_Number と Location を移すために、>> キーを使用します。

OK を押します。



31. 作成したリンク上をダブルクリックします。

Define Node [Collect] - Asset_Number		×
Basic Comment Position Position Input Initial Value Device Barcode Lookup File Formula Cookup File Qualifier Function Keys Links, in Asset Number, #8 -> Location, #2	<ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_ =</li> <li>INIT, Activate Initializatioin</li> <li>RRLI, Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul>	
	OK Cancel Help	

EOR をチェックします。この動作は "Asset\_Number" がそのデータ収集を終了した 後で、データレコードの収集も終了します。

- 32. Verify Node's Status をチェックし、**Pass Always** オプションを選択します。 これにより、F1 キーが押されるまでは、資産番号の収集を続けます。
- 33. OK をクリックし、ジョブを再度保存します。
- 34. ジョブを保存した後で、マウスのポインタをツールバーの Make JoB ボタンに置き ます。資産管理のジョブをコンパイルするために左マウスボタンをクリックします。 これにより実行可能な形式ファイル作成のプロセスをスタートします。



EXE ファイルを作った後で、JobGen Plus はポータブルターミナルにジョブの実行形式 をダウンロードし、実行の準備ができました。

35. ポータブルターミナルを起動して、本プログラムを実行して下さい。 所在地情報を入力した後、F1キーを押すまで資産番号の入力が続きます。

# 第2章 複雑な JobGen Plus アプリケーション

#### 在庫管理

このアプリケーションは、現在の在庫情報を収集します。

在庫情報は三つの種類、部品番号のみ、部品番号と数量、そして部品番号とシリアル番号に分けることができます。このジョブを設計する前に、ユーザはこれらの基本的な要求のノートを作る必要があります:

- データタイプ ポータブル・データコレクタの各データファイルは複数のデータレ コードで作られています。三つのデータフィールド、Part Number(部品番 号), Quantity(数量)、Serial Number(シリアル番号) がこのジョブに必要です。 ユーザは各データフィールドについてデータタイプ(文字列または整数)を 決めなければなりません。このジョブは三つの違った情報タイプを含んで いるので、データレコードの内容は異なっています。部品番号だけの在庫 情報については、Quantity フィールドと Serial Number フィールドは "0" に セットされます。部品番号と数量のある在庫情報については、Serial Number フィールドは "0" にセットされます。部品番号とシリアル番号を 持つ在庫情報については Quantity フィールドは "1" にセットされます。
- プロセス ユーザは実行する必要のあるプロセスをまとめなければなりません。この ジョブは15のプロセスを必要としています:
  - 1. 在庫情報のタイプを選択
  - 2. 部品番号を収集
  - 3. 部品番号を検証
  - 4. 数量のフィールドを0にセット(選択したタイプが部品番号のみの場合)
  - 5. 数量のフィールドを1にセット(選択したタイプが部品番号とシリア ル番号の場合)
  - シリアル番号フィールドを0にセット(選択したタイプが部品番号の みか、部品番号と数量の場合)
  - 7. 数量を収集 (選択したタイプが部品番号と数量の場合)
  - 8. 数量を検証(選択したタイプが部品番号と数量の場合)
  - 9. シリアル番号がバラバラか、連続かを選択(選択したタイプが部品番号とシリアル番号の場合)

- 10. シリアル番号を収集 (選択したタイプが部品番号とシリアル番号の場合)
- 11. シリアル番号を検証 (選択したタイプが部品番号とシリアル番号の場合)
- 12. 最初のシリアル番号を収集 (選択したタイプが部品番号とシリアル番号で、シリアル番号が連続の場合)
- 13. 最初のシリアル番号を検証(選択したタイプが部品番号とシリアル番号で、シリアル番号が連続の場合)
- 14. 在庫の全数量を収集 (選択したタイプが部品番号とシリアル番号、シ リアル番号は連続の場合)
- 15. 連続したシリアル番号を自動的にデータファイルに書き込む (選択したタイプが部品番号とシリアル番号、シリアル番号は連続の場合)

在庫管理の作業フローは以下の方法で行われます:

- 1. ユーザは情報のタイプを選択します:
  - (a) 部品番号のみ
  - (b) 部品番号と数量
  - (c) 部品番号とシリアル番号
- 2. (a) を選択した場合、次のステップに進みます。 (b) を選択した場合、 ステップ 7 に進みます。(c)を選択した場合、ステップ 13 に進みます。
- 3. 部品番号(Part Number) 情報の収集
- 4. ユーザが "Exit" キー (CMD -> ESC) を押したら、ステップ 1 に進みま す。
- 5. 部品番号情報を検証します。入力データが検証をパスしなかった場合、 ステップ3へ進み、それ以外は次のステップへ進みます。
- 6. 数量(Quantity) フィールドとシリアル番号(Serial Number)の両方のフィールドに "0"をセットします。次の入力のためにステップ 3 へ進みます。
- 7. 部品番号(Part Number) 情報を収集します。
- 8. "Exit" キー(CMD -> ESC) が押されたら、ステップ1へ進みます。
- 9. 部品番号情報を検証します。入力データが検証をパスしなかった場合、 ステップ7へ進みます。それ以外は、次のステップへ進みます。
- 10. 数量(Quantity)情報を収集します。

- 11. 数量(Quantity) 情報を検証します。入力データが検証をパスしなかった場合、ステップ11に進みます。それ以外は、次のステップへ進みます。
- 12. シリアル番号(Serial Number) フィールドに"0" を入れます。ステップ 7 に進みます。
- 13. 部品番号(Part Number) 情報を収集します。
- 14. "Exit" キー (CMD -> ESC) が押された場合、ステップ1に進みます。
- 15. 部品番号(Part Number) 情報を検証します。入力データが検証をパス しなかった場合、ステップ 13 に進みます。それ以外は、次のステッ プに進みます。
- 16.シリアル番号が連続か、連続でないかを選択します。
- 17. シリアル番号が連続の場合、次のステップに進みます。それ以外は、 ステップ 25 に進みます。
- 18. 最初のシリアル番号を収集します。
- 19. シリアル番号(Serial Number) 情報を検証します。入力データが検証を パスしなかった場合、ステップ 18 に進みます。それ以外は、次のス テップに進みます。
- 20. 連続する項目の番号を収集します。
- 21. 番号を検証します。入力データが検証をパスしなかった場合、ステップ 20 に進みます。それ以外は、次のステップに進みます。
- 22. カウンタをゼロにセットします。
- 23. 数量(Quantity) フィールドに"1" を入れ、データファイルに連続シリ アル番号(Serial Number) を追加します。
- 24. カウンタに 1 を加えます。カウンタは連続する項目番号に等しい場合、 ステップ 13 に進みます。それ以外はステップ 23 に進みます。
- 25. シリアル番号を収集します。
- 26. シリアル番号情報(Serial Number) 情報を検証します。入力データが検証をパスしなかった場合、ステップ 25 に進みます。それ以外は次のステップに進みます。
- 27. 数量(Quantity) フィールドに "1" を入れます。ステップ 13 に進みます。

## 在庫管理の作成

複雑なので、このジョブの作成は作業フローに沿って進みます。各ステップは作業フローのプロセスとプロセスを実現するのに必要な JobGen Plus で対応する手順を示します。

### <u>作業フロー</u>

- 1. ユーザは情報のタイプを選択します:
  - (a) 部品番号のみ
  - (b) 部品番号と数量
  - (c) 部品番号とシリアル番号

### JobGen Plus



- 1. マウスポインタを Menu ノード ボタンに移動します。
- 2. マウスポインタが Menu ノード ボタンにある時に左マウスボタンをクリックします。
- 3. マウスポインタを Menu ノード が置かれる場所に移動します。左マウスボタンをダ ブルクリックします。



 新しい Menu ノード が作成されます。マウスポインタが Menu ノード にある時に左 マウスボタンをダブルクリックします。"Define Menu" ウインドウがスクリーンに現 れます。

**ヒント:**日本語メニューを作成する場合は、上記の作業をする前に Edit メニューから Define Portable を選択し、Setting > Screen > Large Font をチェックして下さい。

Define Node [Menu] - menu1		×
Basic Comment Screen Function Keys Links, in Links, out	Screen: rows: 12, columns: 4 1.部品#のみ 2.部品#&数量 3.部品#&SN# 〈Exit〉終了。 Do Not Clear Screen Before Display Message Do Not Clear Screen When Exit This Menu Node	4
	OK Cancel Help	

- 5. 以下の情報を入力します:
  - (a) Basic プロパティで、名前を "mainMenu"に変更します。
  - (b) Screen プロパティで、ポータブルターミナルのスクリーンに表示する内容をタイ プします。
  - (c) Basic プロパティで、Start Node オプションをチェックします。
  - (d) Function Keys プロパティで、 "Exit" をチェックします。 "Exit"キー (CMD -> ESC) が押された場合、mainMenu ノード が処理されます。

Define Node [Menu] - menu1				×
Basic Comment Screen Function Keys Links, in Links, out	Press selected function node has no link with Function Keys: F1 F2 F5 F5 F6 Arrow Keys: Left Rig Other Keys: Exit (Esc)	n key defined be same function ke O F3 O F7 ht O Up	elow will run this node, if the runn ey as trigger condition. C F4 C F8 C Down	
	ОК	Cancel	Help	

6. OK をクリックします。

<u>作業フロー</u>

- 2. (a) を選択した場合、ステップ3に進みます。 (b) を選択した場合、ステップ7に進みます。(c) を選択した場合、ステップ13に進みます。
- 3. 部品番号 (Part Number) 情報を収集します。
- 4. "Exit" (CMD > ESC) キーが押されたら、ステップ1に進みます。
- 5. 部品番号 (Part Number) 情報を検証します。入力データが検証をパスしない場合、 ステップ3に進みます。それ以外は、ステップ6に進みます。
- 6. 数量(Quantity) フィールドとシリアル番号(Serial Number) フィールドの両方に"0" を 入れます。次の入力のためにステップ3に進みます。

#### JobGen Plus

- Collect(収集) ノードボタンにマウスポインタを移動することによって、部品番号の ための収集 ノードを作成します。マウスポインタが Collect(収集) ノードボタンにあ る時に左マウスボタンをクリックします。
- 2. Collect(収集)ノードが置かれる場所にマウスポインタを移動します。左マウスボタンをダブルクリックします。

Eile Edit Nodes View Build Tools Window Help     Image: Constraint of the state of the	🍣 JobGen Lite - Job1		
D D   D	<u>File E</u> dit <u>N</u> odes <u>V</u> iew <u>B</u> uild <u>T</u> i	ools <u>W</u> indow <u>H</u> elp	
Image: Second		🖻   🗖   ≪   🗻   🗖 🖾   🖧   🖧   🐺   📲	
Image: Solution       Image: Solution	🖬 🛃   🚦	) 🕂 E 🛲 🖏 📾 🛛 E 🔳 🚼 🍾	
<ul> <li>Portable</li> <li>Records</li> <li>Comment</li> <li>Excluded</li> <li>Nodes, Menu</li> <li>Nodes, Collect</li> <li>Nodes, Collect</li> <li>Nodes, Erase</li> <li>Nodes, Upload</li> <li>Nodes, Run-Job</li> <li>Links</li> <li>Links, Other Conditions</li> <li>Links, Other Conditions</li> </ul>	😤 Job1		- I X
	<ul> <li>Job</li> <li>Portable</li> <li>Records</li> <li>Comment</li> <li>Excluded</li> <li>Nodes, Menu</li> <li>Nodes, Collect</li> <li>Nodes, Edit</li> <li>Nodes, Edit</li> <li>Nodes, Upload</li> <li>Nodes, Run-Job</li> <li>Links, Key Conditions</li> <li>Links, Other Conditions</li> </ul>	collect2 mainMenu	

3. 新しい収集(collect) ノードが作成されました。マウスポインタが収集(Collect)ノード にある時に左マウスボタンをダブルクリックします。 "Define Collect" ウインドウが スクリーンに現れます。

Define Node [Collect] - collect2	×
Basic Comment Screen -Input Initial Value Device	This is a collect node. Name: collect2
Barcode Lookup File Formula Verify Lookup File	T Start Node: Job will start running from this node
Qualifier Function Keys Links, in Links, out	Color: Node 💌
	K Exclude This Node When Make Job

- 4. 以下のオプションを変更します。
  - (a) Basic プロパティで、名前を "pnOnly"に変更します。
  - (b) Screen プロパティをクリックします。

Define Node [Collect] - pnOnly Basic Comment	Screen: rows: 12, columns: 4	X
Position Input Device Barcode Lookup File	                   	
Formula ⊡.Verify …Lookup File …Qualifier …Function Keys ⊕.Links, in ⊕.Links, out		
	OK Cancel Help	

- (c) スクリーンにプロンプト(入力要求)をタイプします。.
- (d) Input > Device  $\mathcal{T}$ ロパティをクリックします。

Define Node [Collect] - pnOnly		×
Basic Comment Screen Position Position Input Initial Value Device Barcode Lookup File Lookup File Qualifier Function Keys CLinks, in <- copy_all, #8 <- mainMenu, #1 Links, out	Timer:         ● Date Only       ● Time Only       ● Date And Time         Date Type:       MM/DD/YYYY       ▼         ▼ Scanner:       ■       Turn Off Scanner After It Read A Barcode         ▼ Keypad:       ●       ■         ▼ Digits And +       ▼ Alphabet       ▼ Symbols         ▼ Complete With Enter       ■	
	OK Cancel Help	

- (e) Keypad と Scanner の両方のオプションが選択していることを確認して下さい。
- (f) Verify プロパティをクリックして下さい。

Define Node [Collect] - pnOnly	×
Basic — Comment — Comment — Position — Input — Initial Value — Device — Barcode — Lookup File — Formula — Verify — Lookup File — Qualifier — Runction Keys — Links, in — <- copy_all, #8 — <- mainMenu, #1 — Links, out — <- copy_all #7	Image:       Image:         From:       Image:         From:       Image:         Picture:       Image:         Image:       Image: <td< td=""></td<>
	OK Cancel Help

- (g) Length オプションをチェックして、最小値:3を From に、そして最大値 16 を To: にセットして下さい。
- (j) OK をクリックして下さい。

- 5. Program ノード ボタンにマウスポインタを移動して、ポインタが Program ノードボ タンにある間に左マウスボタンをクリックして下さい。
- Program ノード を置く場所にマウスポインタを移動して下さい。左マウスボタンを ダブルクリックして下さい。



7. 新しく作成した Program ノード にマウスポインタがある間に左マウスボタンをダブ ルクリックして下さい。 "Define Node[Program]" ウインドウがスクリーンに現れま す。

Define Node [Program] - program3	×
Basic Comment Program Function Keys Links, in Links, out	<pre>copy_all() {   strcpy(_pNumber,_pnOnly);     quantity = 0;     strcpy(_serial, "0"); } </pre>
	OK Cancel Help

 Cの Program ノードは主にデータフィールドに情報を埋めるためのものです。この ジョブには pNumber、 quantity(数量)と serial(シリアル)の三つのデータフィールド があります。Collect ノード pnOnly は pNumber についての情報を一時的に保存し ます。したがって、この Program ノード は pnOnly の情報を pNumber にコピーし、 そして他の二つのフィールドを "0"で埋めます。この Program ノードを定義するた めに、ユーザはノードの名前を "copy\_all"に変更する必要があります。そして、以 下の C ソースコードをタイプします:

```
_copy_all()
{
   strcpy(_pNumber,_pnOnly);
   _quantity = 0;
   strcpy(_serial, "0");
}
```

- これらのノード間のリンクをここで作成する必要があります。マウスポインタを リンクボタンに移動し、マウスボタンがリンクボタン上にある間に左マウスボタン をクリックします。
- 10. マウスポインタを mainMenu ノード に移動し、左マウスボタンをクリックしたまま 保持します。マウスポインタを pnOnly ノード に移動し、左マウスボタンを離しま す。

🍣 JobGen Plus - Job1		×
<u>F</u> ile <u>E</u> dit <u>N</u> odes <u>V</u> iew <u>B</u> uild	<u>T</u> ools <u>W</u> indow <u>H</u> elp	
🗅 🖻 🔒 🗐 🐰 🖻	🖻 🗇 💰 🧩 🛅 🗖 🖾 🖧 💂	Ö
<b>1</b>   <b>1</b>   <b>1</b>	<mark>#?</mark> 🗉 ### 💁 📾 🐼 🗗 📠 🚼 👟	
🕎 Job1		×
F8 Left Right Up Down Exit Records Comment Excluded Nodes, Menu MainMenu Nodes, Collect PnOnly: string Nodes, Math Nodes, Edit Nodes, Edit Nodes, Erase Nodes, Program Nodes, Run-Job Clinks MainMenu -> pnOnly, #3 Links Key Conditions		

11. マウスポインタが新しく作成した リンク にある間に左マウスボタンをダブルクリッ クします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Menu] - mainMenu	×
Basic Comment	
Screen Function Keve	
Links, in	O Pass Always O Success O Fail
⊡-Links, out <mark>-&gt; pnOnly, ‡1</mark>	C Expression:
	_RESULT_ =
	EQB Write Becord:
	□ INIT. Activate Initialization
	RRLI, Reset "Repeat Last Input" For All Nodes
	I Alarm
	j ★ Exclude When Make Job
	OK Cancel Help

- 12. **Keypad Input** オプションをチェックし、そして **Keypad Input** オプションの下のス ペースにマウスポインタがある間に左マウスボタンをクリックして下さい。 "PT-600 Keypad" ウインドウがスクリーンに現れます。
- 13.1 ボタンをクリックします。 これは"1"を押すと mainMenu ノード から pnOnly ノ ード にプロセスが移動することを示しています。 "PT-600 Keypad" ウインドウを閉 じるために OK ボタンをクリックします。



OK をクリックして下さい・

14. マウスポインタを pnOnly ノード に移動して、そして左マウスボタンをクリックし たままにします。 マウスポインタを copy\_all ノード に移動し、左マウスボタンを 離します。



15. 新しく作成した リンク にマウスポインタがある間に左マウスボタンをダブルクリックします。 "Define Node" ウインドウがスクリーンに現れます。
| Defir | ne Node [Collect] - pnOnly  |   | × |
|-------|---|---|---|
|       | Basic<br>Comment<br>Screen<br>Input<br>Initial Value<br>Device<br>Barcode<br>Lookup File<br>Formula<br>Verify<br>Qualifier<br>Function Keys<br>Links, in<br>Links, out<br>Scopy all, #4 | <ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_=</li> <li>EOR Write Record:</li> <li>INIT, Activate Initializatioin</li> <li>RRLI Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul> |   |
|       |   | OK Cancel Help  |   |

Pass Always オプションは ジョブ が pnOnly ノード のプロセスを終えたら、copyall ノード を処理することを示しています。

- 16.**OK をクリックして下さい。**
- 17. マウスポインタを copy\_all ノード に移動し、そして左マウスボタンをクリックした ままにします。マウスポインタを pnOnly ノード に移動し、マウスボタンを離しま す。



18. マウスポインタが新しく作成した リンク 上にある間に左マウスボタンをダブルクリ ックします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Program] - copy_all	×
Basic Comment Program Function Keys ⊡ Links, in ⊡ Links, out PnOnly, #5	<ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_=</li> <li>EOR Write Record:</li> <li>INIT, Activate Initializatioin</li> <li>RRLI Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul>
	OK Cancel Help

Pass Always オプションはジョブ が copy\_all ノード の処理を終えたら、その後 pnOnly の処理を行うことを示しています。

19. EOR オプションをチェックして下さい。

21. Ok をクリックしてください。

## <u>作業フロー</u>

7. Part Number 情報の収集.

- 8. "Exit" キーが押されたら、ステップ1へ進む。
- 9. Part Number 情報の検証。入力データが検証をパスしなかったら、ステップ 7 へ進む。 そうでなければ、ステップ 10 へ進む。
- 10. Quantity (数量)情報を収集。
- 11. Quantity 情報を検証。入力データが検証をパスしなかった場合、ステップ 10 に進み ます。そうでなければ、ステップ 12 に進みます。
- 12. Serial Number フィールドに"0"を入れます。ステップ7に進みます。

JobGen Plus

- マウスポインタを Collect Node ボタンに移動することによって Part Number につい ての Collect ノード を作ります。マウスポインタが Collect ノード ボタンにある間に 左マウスボタンをクリックします。
- Collect ノード を置く場所にマウスポインタを移動します。左マウスボタンをダブ ルクリックします。



3. 新しい Collect ノード が作られました。マウスポインタが Collect ノード にある間 に左マウスボタンをダブルクリックして下さい。 "Define Node [Collect]" ウインドウ がスクリーンに現れます。

Define Node [Collect] - collect4								
Basic Comment	This is a collect node.							
⊕ Screen ⊡ Input … Initial Value … Device	Name: collect4							
Barcode Lookup File Formula	Start Node: Job will start running from this node							
Lookup File Qualifier Function Keys Links, in Links, out	Color: Node 💌							
	┌─ * Exclude This Node When Make Job							
	OK Cancel Help							

- 4. 以下のオプションを変更します。
  - (a) 名前を "pnQty" に変更します。
  - (b) Screen プロパティをクリックします。

Define Node [Collect] - pnOnly		×
Basic Comment Position Position Input Initial Value Device Barcode Lookup File Formula Cookup File Qualifier Qualifier Links, in Links, out	Screen: rows: 12, columns: 4 く E x i t ) メ イ ン メ ニ ュ ー 部 品 番 号 : 。。。 。 。 。 。 。 。 。 。 。 。 。 。 。	
	OK Cancel Help	

- (c) スクリーンのプロンプト(入力要求)をタイプします。
- (d) Input > Device  $\Box \Box \Box \Box \neg z = z = z$

Define Node [Collect] - collect4		×
Basic Comment Screen Position Input Device Barcode Lookup File Formula Perify Lookup File Qualifier Function Keys Links, in Links, out	Timer:         O Date Only       Image: Time Only         Date Type:       MM/DD/YYYY         ✓       Scanner:         ✓       Turn Off Scanner After It Read A Barcode         ✓       Keypad:         ✓       Digits And +         ✓       Alphabet         ✓       Symbols         ✓       Complete With Enter	
	OK Cancel Help	

- (e) Keypad と Scanner オプションが選択されていることを確認して下さい。
- (f) Verify プロパティをクリックして下さい。

Define Node [Collect] - collect4		×
Basic Comment Screen InPosition Input Initial Value Device Barcode Lookup File Formula Formula Qualifier Qualifier Links, in Links, out	Image:   From:   3   To:   16   From: To: Picture: Picture: Do Not Force To Re-Input If Not Passed Verification Do Not Beep If Not Passed Verification Do Not Beep If Not Passed Verification Do Not Print Error Sign (Character * ) If Not Passed Verification	
	OK Cancel Help	

- (g) **Length** オプションをチェックし、最小長 3 を From に、最大長 16 を To に を入 れて下さい。
- (j) OK をクリックして下さい。

- 5. Collect ノード が置かれる場所にマウスポインタを移動して下さい。左マウスボタン をダブルクリックして下さい。
- 6. マウスポインタが Collect Node にある間に左マウスボタンをダブルクリックして下 さい。 "Define Node [Collect]" ウインドウがスクリーンに現れます。



- 7. 以下のオプションを変更します。
  - (a) 名前を "quantity"に変更します。
  - (b) Screen プロパティをクリックします。

Define Node [Collect] - collect5																×
Basic Comment Screen Position Input	Screen: rows: 12, columns: 4 数量:◇◇◇◇◇◇◇◇															
- Initial Value - Device - Barcode	<u>ہ</u>															
Lookup File Formula Forfy												<u>ہ</u>				
Lookup File Qualifier Function Keys Links, in Links, out	~	<u> </u>	<u> </u>		•	•	•	0		•	•	<u> </u>				
				O	<				Can	cel			Н	lelp		

- (c) プロンプト(入力要求)をタイプします。
- (d) **Input** プロパティをクリックします。Numerical Integer をチェックします。

Define Node [Collect] - quantity		×
Basic Comment Screen □ Input Initial Value Device Barcode Lookup File Formula □ Verify Lookup File Qualifier Function Keys ⊡ Links, in ⊡ Links, out	Data Type: • Numerical, Integer • Numerical, Floating-Point • Character String • Set Data As Same As Last Record, And Complete Input • Fill Tail With White Spaces • Allow Empty Data	
	OK Cancel Help	

- (e) Input > Device  $\Box \Box \Gamma \Gamma \tau = 0$
- (f) Keypad と Scanner オプションの両方が選択されていることを確認して下さい。
- (g) Verify プロパティをクリックして下さい。

Define Node [Collect] - quantity		×
Basic Comment Screen Position Input Initial Value Device Barcode Lookup File Verify Lookup File Qualifier Function Keys Links, in Links, out	✓ Length:         From:       1         To:         From:         To:         Picture:         On Not Force To Re-Input If Not Passed Verification         Do Not Beep If Not Passed Verification         Do Not Print Error Sign (Character * ) If Not Passed Verification	
	OK Cancel Help	

- (h) **Length** オプションをチェックして、最少長1を From に、そして最大長5を To: に入れます。
- (i) **Range** オプションをチェックし、最低限の値: 0 を From:に、そして最大の制限値 32000 を To: にタイプします。

(k) **OK** をクリックします。

- 8. マウスポインタを Program ノード ボタンに移動します。ポインタが Program ノード ボタンにある間に左マウスボタンをクリックします。
- 9. Program ノード がおかれる場所にマウスポインタを移動します。左マウスボタンを ダブルクリックします。

🍣 JobGen Plus - Job1		
<u>File Edit N</u> odes <u>V</u> iew <u>B</u> uild	d <u>T</u> ools <u>W</u> indow <u>H</u> elp	
D 🛎 日 🗐   X 🛛	è 🖻 🖆 🐟 🧩 🖪 🗔 🖂 🞝 💂	Ċ
<b>7,</b> 3,   - <b>2</b> ,	) <mark>//?</mark> 🗉 배 💁 💷 🖾 🛃 🐁	
😤 Job1		
F8	A     A	· · · •
Left		
IIRight	· · · · · · · · · · · · · · · · · · ·	
Demo	e e e e en	
Evit		· · ·
Becords	· · · · · · · · · · · · · · · · · · ·	· · ·
Comment	corv all	
Excluded	program6	guantit
📄 Nodes, Menu		
mainMenu	· · · · · · · · · · · · · · · · · · ·	
🛛 🖻 Nodes, Collect		
pnUnly: string		
pnuty: string		🛛
Nodes Meth		📘
Nodes, Math	· · · · · · · · · · · · · · · · · · ·	
Nodes, Erase	· · · · · · · · · · · · · · · · ·	· · ·
Nodes, Upload		
🛛 🗄 Nodes, Program		
Nodes, Run-Job		الخر م
]] 🗄 Linke		

10.マウスポインタが新しく作成した Program ノード にある間に左マウスボタンをダブ ルクリックします。 "Define Node [Program]" ウインドウがスクリーンに現れます。

Define Node [Program] - program6		×
Basic Comment Program Function Keys Links, in Links, out	<pre>copy_serial() {   strcpy(_pNumber, pnQty);   strcpy(_serial, "0"); }</pre>	
	OK Cancel Help	

11. この Program ノード は主にデータフィールドに情報を埋めるためのものです。ジョ ブに、pNumber、quantity そして serial の三つのデータフィールドがあります。 二番目の Collect ノード: quantity は quantity フィールドの情報を保存します。 最初 の Collect ノード: pnQty は pNumber について情報を一時的に保存します。 したが って、 Program ノード は pnQty から pNumber へ情報をコピーし、そして serial の フィールドを "0"で埋めます。この Program ノード を定義するために、ユーザはノ ードの名前を "copy\_serial"に変更し、そして以下の C ソースコードをタイプして下 さい。

```
_copy_serial()
{
   strcpy(_pNumber,_pnQty);
   strcpy(_serial, "0");
}
```

```
12. ここでこれら三つのノード間のリンク を作成する必要があります。マウスポインタ
を リンク ボタンに置いて、マウスポインタが リンク ボタンにある間に左マウスボ
タンをクリックします。
```

13. マウスポインタを mainMenu ノード に移動し、そして左マウスボタンをクリックし たままにします。マウスポインタを pnQty ノード に移動し、左マウスボタンを離し ます。

🍣 JobGen Plus - Job1	
<u>File Edit Nodes View Build Tools Wi</u>	ndow <u>H</u> elp
	ז 🚳 メ 🗖 🗖 🖾 😽 🗛 👫 🍵
	<mark>//?</mark> 🗉 🛲 🖏 💷 🖾 😰 📶 📉
😤 Job1	
	pnOn ly

14. マウスポインタが新しく作成した リンク.にある間に左マウスボタンをダブルクリッ クします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Menu] - mainMenu	Keypad Input     O Input Data:
- Comment - Screen - Function Keys	Verify Node's Status:
Links, in     ⊡ Links, out	Pass Always     Success     Fail
-> pnOnly, #3 -> pnQty, #6	
	EOR, Write Record:
	<ul> <li>INIT, Activate Initializatioin</li> <li>RRLI, Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> </ul>
	☐ * Exclude When Make Job
	OK Cancel Help

15. Key Input オプションをチェックし、マウスポインタを Key Input の次のスペース に置いているときに左マウスボタンをクリックします。 "PT-600 Keypad" ウインド ウがスクリーンに現れます。



16. 2のボタンをクリックします。これは、ユーザが"2"のキーを押した場合、プロセス は mainMenu ノード から pnQty ノード に 進むことを示しています。 "PT-600 Keypad" ウインドウを閉じるために OK ボタンをクリックして下さい。

17.OKをクリックして下さい。

18. マウスポインタを pnQty ノード に移動し、左マウスボタンをクリックしたままに します。 マウスポインタを quantity ノード に移動し、左マウスボタンを離します。

File Edit Nodes Yew Build Tools Window Help     Image: Strate Strate   Image: Stra	🍣 JobGen Plus - Job1		
Image: Constraint of the second s	<u>File Edit N</u> odes <u>V</u> iew <u>B</u> uild	d <u>T</u> ools <u>W</u> indow <u>H</u> elp	
Image: Constraint of the second s	D 🛎 🔒 🗐 🐰 🛛	à 🛱 🗇 🤸 🧩 🗖 🗖 🖾 😽 🗣 👘	
	7 <mark>1</mark> 7 <u>5</u>	<mark>#?</mark> E ## 📎 💷 🖾 🗗 🔂 🌄	
F8      Left        Right      Up        Down	📆 Job1		
	-F8 -Left -Right -Up -Down -Exit -Records -Comment Excluded Nodes, Menu -mainMenu Nodes, Collect -pnOlly: string -pnQty: string -quantity: string -Nodes, Edit -Nodes, Edit -Nodes, Erase Nodes, Program -Nodes, Run-Job -Links	<pre>mainMenu pnQty pnQty quar quar </pre>	nt ity

- 19. マウスポインタが新しく作成した リンク上にある間に左マウスボタンをダブルクリ ックします。
- 20. "Define Node" ウインドウがスクリーンに現れます。

Define Node [Collect] - pnQty Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Curatity 12	Keypad Input: O Input Data: O Verify Node's Status: O Pass Always Success Fail Expression: RESULT_= EOR Write Record: INIT, Activate Initializatioin RRLI, Reset "Repeat Last Input" For All Nodes Alarm
E-Links, out E-Links, out -> quantity, #7	RRLI Reset "Repeat Last Input" For All Nodes Alarm * Exclude When Make Job OK Cancel Help

Pass Always オプションは、ジョブ が pnQty ノードの処理を終えたら、quantity ノードの処理に進むことを示しています。

- 21. OK をクリックして下さい。
- 22. マウスポインタを quantity ノードに移動し、 左マウスボタンをクリックしたまま にします。マウスポインタを copy\_serial ノード に移動し、左マウスボタンを離し ます。



23. マウスポインタが新しく作成したリンク上にある間に左マウスボタンをダブルクリ ックします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Collect] - quantity	×
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Formula Cookup File Qualifier Lookup File Qualifier Lookup File Qualifier Lookup File Qualifier Lookup File Qualifier Lookup File Qualifier Function Keys Copy serial, #8	<ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_=</li> <li>EOR Write Record:</li> <li>INIT, Activate Initializatioin</li> <li>RRLI, Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul>
	OK Cancel Help

Pass Always オプションは、ジョブが quantity ノードの処理を終了したら、 copy\_serial ノード の処理を行うことを意味しています。

- 24. Ok をクリックして下さい。
- 25. マウスポインタを copy\_serial ノード に移動し、左マウスボタンをクリックしたま まにします。マウスポインタを pnQty ノード に移動し、左マウスポインタを放しま す。



26. 新しく作成した リンク 上にマウスポインタがある間に左マウスボタンをダブルクリ ックします。 "Define Node" ウインドウがスクリーンに現れます。

EOR Wri EOR Wri INIT, Act RRLI Re Alarm * Exclude	ite Record:

Pass Always オプションは、ジョブが copy\_serial ノード の処理を終えたら、 pnQty ノード の処理を行うことを示しています。

- 27. EOR オプションをチェックして下さい。
- 27. Ok をクリックして下さい。

## <u>作業フロー</u>

- 13. Part Number 情報を収集。
- 14. "Exit" キーが押されたら、ステップ1へ進みます。
- 15. Part Number 情報を検証。 入力データの検証がパスしなかったら、ステップ 13 に進みます。そうでなければ、次のステップに進みます。
- 16. Serial Number が連続か、連続でないかを選択します。
- 17. シリアル番号(Serial Number)が連続の場合、次のステップに進みます。そうでなけ ればステップ 25 に進みます。
- 18. 最初の Serial Number を収集します。
- Serial Number 情報を検証します。 入力データが検証をパスしなかった場合、ステップ18 に進みます。そうでなければ次のステップに進みます。.
- 20. 連続した項目の番号を収集します。
- 21. 番号を検証します。入力データが検証をパスしなかった場合、ステップ 20 に進みま す。そうでなければ、次のステップに進みます。
- 22. カウンタをゼロにセットします。
- 23. Quantity フィールドに"1"を入れ、データファイルに連続する Serial Number を加えま す。
- 24. カウンタに1を加えます。カウンタが連続する項目の番号に等しい場合、ステップ 13 に進みます。そうでなければステップ23 に進みます。
- 25. Serial Number を収集します。
- 26. Serial Number 情報を検証します。入力データが検証をパスしない場合、ステップ 25 に進みます。そうでなければ、次のステップに進みます。
- 27. Quantity フィールドに"1"を入れます。 ステップ 13 に進みます。

JobGen Plus

 マウスポインタを Collect ノードボタンに移動することによって Part Number につい ての収集ノードを作成します。マウスポインタが Collect ノード ボタンにある間に 左マウスボタンをクリックします。 2. Collect ノード が置かれる場所にマウスポインタを移動します。左マウスボタンをダ ブルクリックします。



3. 新しい Collect ノード が作成されました。マウスポインタが Collect ノード にある間 に、左マウスボタンをダブルクリックします。 "Define Node [Collect]" ウインドウ がスクリーンに現れます。

Define Node [Collect] - collect7		
Basic Comment	This is a collect node.	
⊕ Screen ⊡ Input — Initial Value — Device	Name: collect7	
Barcode Lookup File	Start Node: Job will start running from this node	
Lookup File Qualifier Function Keys Links, in Links, out	Color: Node 💌	
	☐ * Exclude This Node When Make Job	
OK Cancel Help		

- 4. 以下のオプションを変更します:
  - (a) **ノー**ド名を "pNumber" に変更します。
  - (b) Screen プロパティをクリックします。

Define Node [Collect] - pNumber		×
Basic Comment	Screen: rows: 12, columns: 4	
Screen	<exit>メイン</exit>	
Initial Value Device Barcode	Х <u> </u>	
Lookup File Formula	部品番号:◎◎◎	
Lookup File …Qualifier …Function Keys ⊡…Links, in ⊡…Links, out		
	OK Cancel Help	

- (c) プロンプト(入力要求)をタイプします。
- (d) Input > Device  $D \cap C = D \cap C$

Define Node [Collect] - collect7	×
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	Timer:         O Date Only       Image: Time Only       Date And Time         Date Type:       MM/DD/YYYY       ✓         Image: Scanner:       Image: Turn Off Scanner After It Read A Barcode         Image: Keypad:       Image: Symbols         Image: Digits And +       Image: Alphabet       Image: Symbols         Image: Complete With Enter       Image: Symbols
	OK Cancel Help

- (e) Keypad と Scanner オプションの両方が選択されていることを確認して下さい。
- (f) Input プロパティをクリックし、Set Data As Last Record, And Complete Input オプションをチェックします。
- (g) **Verify** プロパティをクリックします。

×
✓ Length:         From:       3         To:       16         Range:         From:       To:         Picture:
OK Cancel Help

(h) Length オプションをチェックし、最小値 3 を From:に、最大値 16 を To: に入れ ます。

## (k) **OK** をクリックします。

- 5. Menu ノード ボタンをクリックします。
- 6. マウスポインタを Menu ノード が置かれる場所に移動します。左マウスボタンをダ ブルクリックします。

🍣 JobGen Plus - Job1	l ×
<u>F</u> ile <u>E</u> dit <u>N</u> odes <u>V</u> iew <u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp	
D 🖉 日 🗐 🖇 🛍 🗂 🔦 🧩 🛅 🖾 🖧 💂 🛱	
🖬 🖏 📲   🚾 🗉   🗰 🖺 🛰	
🞇 Job1	×
-F8         -Left         -Right         -Up         -Down         Exit         -Comment         -Excluded         PNOdes, Menu         -mainMenu         -menu8         PNUmber: string         -pNumber: string         -pNumber: string         -pNumber: string         -Nodes, Math         -Nodes, Edit	

7. 新しい Menu ノード が作成されました。マウスポインタが Menu ノード にある間に 左マウスボタンをダブルクリックして下さい。 "Define Node[Menu]" ウインドウが スクリーンに現れます。

Define Node [Menu] - menu8	×
Basic Comment Screen Function Keys Links, in Links, out	This is a menu node. Name: menu8
	T Start Node: Job will start running from this node
	Color: Node 💌
	OK Cancel Help

- 8. 以下の情報を入力します:
  - (a) 名前を "choice"に変更します。
  - (b) ポータブル・ターミナルのスクリーンに表示する内容をタイプします。

Define Node [Menu] - menu16		×
Basic	Screen: rows: 12, columns: 4	
Screen Function Keys	S/N選択	
Links, in Links, out	1連続   • • • • • • •	
	2 非 連 続 💿 💿	
	☐ Do Not Clear Screen Before Display Message ☐ Do Not Clear Screen When Exit This Menu Node	
	OK Cancel Help	

9. OK をクリックします。

- 10. マウスポインタを Collect ノード ボタンに移動することによってシリアル番号につ いての収集ノード を作成します。 マウスポインタが Collect ノード ボタンにある 間に左マウスボタンをクリックします。
- 11. Collect ノード を置く場所にマウスポインタを移動します。左マウスボタンをダブル クリックします。



12. 新しい収集ノード(Collect ノード)が作成されました。マウスポインタが収集ノード (Collect ノード)にある間に左マウスボタンをダブルクリックします。 "Define Node [Collect] "ウインドウがスクリーンに現れます。

Define Node [Collect] - collect9		×
Basic Comment	This is a collect node.	
⊕ Screen ⊖ Input Initial Value Device	Name: collect9	
Barcode Lookup File Formula ⊡-Verify	TStart Node: Job will start running from this node	
Lookup File Qualifier Function Keys Links, in Links, out	Color: Node 💌	
	☐ * Exclude This Node When Make Job	
	OK Cancel Help	

以下のオプションをチェックします:

- (a) ノードの名前を "serial"に変更します。
- (b) Screen プロパティをクリックします。

Define Node [Collect] - collect9	Scree	en: ro	)ws: "	12, 0	colun	nns: -	4					×
Comment     Screen     Input     … Initial Value     … Device     … Barcode     … Barcode	く シ	F	4 ) IJ		新 P	部 ル	β ,	品番		# 号		
Formula Formula Verify Qualifier Function Keys Links, in Links, out	<ul> <li></li> <li></li> </ul>										<ul> <li></li> <li></li> </ul>	
			(	ЭК				Cano	cel		Help	_

- (c) プロンプト(入力要求)をタイプします。
- (e) **OK** をクリックします。

Program ノード ボタンにマウスカーソルを移動します。Program ノード ボタンにポ インタがある間に左マウスボタンをクリックします。

13. Program ノード を置く場所にマウスポインタを移動します。左マウスボタンをダブ ルクリックします。



14. 新しく作成したプログラムノード(Program ノード)にマウスポインタがある間に左 マウスボタンをダブルクリックします。 "Define Node [Program]" ウインドウがスク リーンに現れます。

Define Node [Program] - copy_qty	×
Basic Comment Function Keys ⊡ Links, in C- serial, #11 ⊕ Links, out	<pre>copy_qty() {   strcpy(tempstr, _pNumber);   strcat(tempstr, ",");   strcat(tempstr, ",");   strcat(tempstr, serial);   AppendRecord(NULL,tempstr); } </pre>
	OK Cancel Help

15. この Program ノード は主に quantity フィールドに情報を埋めます。 pNumber フィ ールドと serial フィールドにデータを入れたら、この Program ノード は quantity フ ィールドに"1"を入れます。 この Program ノード を定義するには、ユーザはノード 名を "copy\_qty"に変更し、以下の C ソースコードをタイプする必要があります。

```
_copy_qty()
```

{

```
strcpy(tempstr, _pNumber);
```

```
strcat(tempstr,",");
```

```
strcat(tempstr,"1");
```

```
strcat(tempstr,",");
```

strcat(tempstr,\_serial);

AppendRecord(NULL,tempstr);

```
}
```

16.**OK をクリックして下さい。** 

- 17. マウスポインタを Collect ノード ボタンに移動します。マウスポインタが Collect ノード ボタンにある間に左マウスボタンをクリックします。
- 18. Collect ノード を置く場所にマウスポインタを移動します。左マウスボタンをダブル クリックします。

🍣 JobGen Plus - Job1		
<u>File E</u> dit <u>N</u> odes <u>V</u> iew	<u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp	
D 📽 日 🗐   J	( 🖻 🖻 🔿 メ 🖪 🗖 🗷 😽 🐥 👘	
	<mark>#?</mark> 🗉 🗰 🗞 💷 🔀 🔂 🧏	
😤 Job1		
F8		
Left Diabet		
		·
Down	pn0nly mainMenu	:
Exit	· · · · · · · · · · · · · · · · · · ·	·
Comment		·
Excluded	quantit	y
Nodes, Menu		
mainMenu	colle	
📄 Nodes, Collect	choice serial	•
collect11: string		
pnQty: string	· · · · · · · · · · · · · · · · · · ·	•
	Copy_qty	:
quantity: string	· · · · · · · · · · · · · · · · · · ·	
Nodes, Math		
<u>]]</u>		
		11.

19. 新しい Collect ノード が作成されました。マウスポインタが Collect ノードにある間 に左マウスボタンをダブルクリックします。 "Define Node [Collect]" ウインドウが スクリーンに現れます。

Define Node [Collect] - collect11	×
Basic Comment	This is a collect node.
⊡-Screen ⊡-Input Initial Value Device	Name: collect11
Barcode Lookup File Formula	Start Node: Job will start running from this node
Lookup File Qualifier Function Keys Links, in Links, out	Color: Node 💌
	🔲 * Exclude This Node When Make Job
	OK Cancel Help

## 20. 以下のオプションを変更します。

- (a) ノード名を "startSN"に変更します。
- (b) Screen プロパティをクリックします。

Function Keys	-Screen -Input - Initial Value	ļ	F]	捝	ì	S	N	λ	•	ታ	[	•	0			
Formula Verify Lookup File Qualifier Function Keys 	Device Barcode	•											0			
		<ul> <li></li> </ul>														
		<u>ہ</u>														

(c) プロンプト(入力要求)をタイプします。

(d) Input > Device  $D \cap C$ 

Define Node [Collect] - collect11	
Basic Comment Comment Screen Input Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	Timer:       Image: Date Only       Time Only       Date And Time         Date Type:       MM/DD/YYYY       Image: Date Type:       MM/DD/YYYY         Image: Scanner:       Image: Disconner After It Read A Barcode         Image: Keypad:       Image: Disconner After It Read A Barcode         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols         Image: Disconner After It Read A Barcode       Image: Symbols
	OK Cancel Help

- (e) Keypad と Scanner オプションの両方が選択されていることを確認して下さい。
- (f) Verify プロパティをクリックして下さい。

Define Node [Collect] - collect11	X
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Formula Formula Ualifier Qualifier Links, in Links, out	Image:       Image:         From:       To:         Image:       Image:         Ima
	OK Cancel Help

- (g) Length オプションをチェックして、最小値 1 を From: に、そして最大値 32 を To: に入れます。
- (h) **OK**をクリックします。
- 21. マウスポインタを Collect ノード が置かれる場所に移動します。左マウスボタンを ダブルクリックします。

🍣 JobGen Plus - Job1		
<u>F</u> ile <u>E</u> dit <u>N</u> odes <u>V</u> iew	<u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp	
	h 🖬 🗂 🚳 🧩 🛅	🗆 🖂 😽 🔒 🛱
	<mark>#?</mark> 🗉 🗰 🗞 🔤	] 🖾 🗊 📠 🚼 💫
😤 Job1		
F8 -Left -Right -Up -Down Exit -Records -Comment -Excluded -Nodes, Menu -choice -mainMenu -choice -mainMenu -collect12: string -pnQty: string -pnQty: string -serial: string -srartSN: string -srartSN: string	▶       Pn0nly         Pn0nly	a inMenu pnQty Number copy quantity choice serial

22. マウスポインタが Collect ノード にある間に左マウスボタンをダブルクリックしま す。 "Define Node [Collect]" ウインドウがスクリーンに現れます。

Define Node [Collect] - collect12	×
Basic Comment Screen Input Initial Value Device	This is a collect node. Name: collect12
Barcode Lookup File Formula ⊟-Verify Lookup File	T Start Node: Job will start running from this node
Longualifier Function Keys Links, in Links, out	Color: Node 💌

23. 以下のオプションを変更します。

(a) ノード名を "counter"に変更します。

(b) Screen プロパティをクリックします。

Define Node [Collect] - collect12																×	1
Basic	Scre	en: r	ows	: 12	l, co	lum	ns: 4	4									
⊕-Screen ⊖-Input Initial Value	ታ		ゥ		ン		ş		λ	•	t	J					
Device Barcode Lookup File	:	0															
… Formula ⊡ Verify … Lookup File	<u> </u>																
Qualifier —Function Keys —Links, in —Links, out	<ul> <li></li></ul>																
				01	<				Can	cel			Help	Þ			

- (c) プロンプト(入力要求)をタイプします。
- (d) Input  $\mathcal{J}$ ロパティをクリックします。Numerical Integer をチェックします。

Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Uerify Lookup File Qualifier Function Keys Links, in Links, out	Data Type: Numerical Integeri Numerical, Long Integer Numerical, Floating-Point Character String Set Data As Same As Last Record, And Complete Input Fill Tail With White Spaces Allow Empty Data
	OK Cancel Help

(e) **Input > Device**  $\mathcal{T}$ ロパティをクリックします。

Define Node [Collect] - collect12		×
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	<ul> <li>Timer:</li> <li>Date Only</li> <li>Time Only</li> <li>Date Type: MM/DD/YYYY</li> <li>✓ Scanner:</li> <li>Turn Off Scanner After It Read A Barcode</li> <li>✓ Keypad:</li> <li>✓ Digits And +</li> <li>✓ Alphabet</li> <li>✓ Symbols</li> <li>✓ Complete With Enter</li> </ul>	
	OK Cancel Help	

- (f) Keypad オプションをチェックします。
- (g) Verify プロパティをクリックします。

Define Node [Collect] - collect12	×
Basic Comment G.Screen □.Input Initial Value Device Barcode Lookup File Formula Formula Cookup File Qualifier Qualifier Links, in Links, out	✓ Length:       From:       To:       2         Range:       From:       To:       1         Prom:       To:       1       1         Picture:       1       1       1         Do Not Force To Re-Input If Not Passed Verification       1       1         Do Not Force To Re-Input If Not Passed Verification       1       1         Do Not Force To Re-Input If Not Passed Verification       1       1         Do Not Print Error Sign (Character * ) If Not Passed Verification       1       1
	OK Cancel Help

**Length** オプションをチェックし、最小値 1 を From:に、最大値 2 を To: に入れます。

(h) **Range** オプションをチェックし、From を 0、To:を 30 とします。

(k) **OK**をクリックします。

- 26. マウスポインタを Program ノード ボタンに移動します。ポインタが Program ノード ボタンにある間に左マウスボタンをクリックします。
- 27. Program ノード が置かれる場所にマウスポインタを移動します。左マウスボタンを ダブルクリックします。



28. 新しく作成した Program ノード にマウスポインタがある間に左マウスボタンをダブ ルクリックします。 "Define Node [Program]" ウインドウがスクリーンに現れます。



29. この Program ノード は主にデータフィールドに情報を埋めます。ユーザははじめの Serial Number と保存する Serial Numbers 数を入力する必要があります。この Program ノード は quantity フィールドに"1"を入れ、そして生成した連続するシリ アル番号を serial フィールドに入れます。この Program ノード を定義するには、ユ ーザはノード名を "add sn"に変更し、以下のCソースコードをタイプします。

```
_add_sn()
```

```
{
```

```
long int temp;
```

int i;

```
clear_screen();
```

```
temp = atol(_startSN);
```

```
for (i=0;i<_counter;i++)</pre>
```

```
{
```

```
printf("%ld\n",temp);
```

```
_ltoa(temp,str,10);
```

```
strcpy(tempstr,_pNumber);
```

```
strcat(tempstr, ",");
```

```
strcat(tempstr, "1");
```

```
strcat(tempstr, ",");
```

```
strcat(tempstr, str);
```

AppendRecord(NULL,tempstr);

```
temp++;
}
```

- 30. OK をクリックします。
- 31. これらの三つの ノード 間にリンクを作成する必要があります。 マウスポインタを リンクボタンに移動し、マウスポインタがリンクボタンにある間に左マウスボタン をダブルクリックします。
- 32. マウスポインタを mainMenu ノード へ移動し、左マウスボタンをクリックしたまま にします。マウスポインタを pNumber ノード へ移動し、左マウスボタンを離しま す。



33. マウスポインタが新しく作成したリンク上にある間に左マウスボタンをダブルクリ ックします。 "Define Node" ウインドウがスクリーンに現れます。
| Define Node [Menu] - mainMenu  | ×   |
|--|---|
| Basic<br>Comment<br>Screen<br>Function Keys<br>Links, in<br>⊡-Links, out<br>-> pnOnly, #3<br>-> pRty, #6<br>> pNumber, #10 | <ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_=</li> <li>EOR Write Record:</li> <li>INIT, Activate Initializatioin</li> <li>RRLL Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul> |
|  | OK Cancel Help  |

34. **Key Pad Input** オプションをチェックし、**Key Pad Input** オプションの次のスペース 上にマウスポインタがある間に左マウスボタンをクリックします。 "PT-600 Keypad" ウインドウがスクリーンに現れます。



35. 3 のボタンをクリックします。これは、ユーザが"3" のキーを押した場合、プロセ スが mainMenu ノード から pNumber ノードに進むことを意味します。 "PT600 Keypad" ウインドウを閉じるために OK ボタンを押します。

36. **OK をクリックします**。

37. マウスポインタを pNumber ノード に移動し、 左マウスボタンをクリックしたまま にします。マウスポインタを choice ノード に移動し、左マウスボタンを離します。

Eile Edit Nodes View Build Tools Window Help	🍣 JobGen Plus - Job1			
D       B       B       B       B       C	<u>File E</u> dit <u>N</u> odes <u>V</u> iew	<u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp		
Image: Second	D 🛎 日 🗐   >	; 🖻 🛍 🎒 🚳 🦂	メ 🛅 🗖 🖉 🛛	<b>A</b>   🖶   🐴
F8 Left Right Up Down Exit Records Comment Excluded B Nodes, Menu Copy_all PNUmber PNUmber PNUmber Copy_all PNUmber Copy_all PNUmber Copy_all Copy		J <mark>#?</mark> 🗉	### 💁 💷 🖾 📝 📠	
F8 Left Right Up Down Exit Records Comment Excluded Nodes, Menu choice mainMenu Nodes, Collect counter: string pnOnly: string pnOnly: string pnOnly: string counter: string srartSN choice serial string srartSN: string srartSN: string srar	😤 Job1			
	F8 Left Right Up Down Exit Records Comment Excluded Nodes, Menu choice mainMenu choice mainMenu counter: string pnOnly: string pnQty: string pnQty: string nQty: string 	■ PnOn ly • • • • • • • • • • • • • • • • • • •	mainMenu PNumber choice	pnQty  pnQty  quantity  serial  copy_qty

38. 新しく作成したリンク上にマウスポインタがある間に左マウスボタンをダブルクリ ックします。"Define Node" ウインドウがスクリーンに現れます。

Define Node [Collect] - pNumber	×
Basic — Comment → Screen → Input — Initial Value — Device — Barcode — Lookup File — Formula — Verify — Lookup File — Qualifier — Function Keys — Links, in — Links, out — Choice, #11	<ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_=</li> <li>EOR Write Record:</li> <li>INIT, Activate Initializatioin</li> <li>RRLI Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul>
	OK Cancel Help

Pass Always オプションはジョブ が pNumber ノードの処理を終えたら、 choice ノードを処理することを示しています。

- 39. OK をクリックします。
- 40. マウスポインタを choice ノード に移動し、左マウスボタンを押したままにします。 マウスポインタを serial ノードに移動し、左マウスボタンを離します。



41. マウスポインタが新しく作成したリンク上にある間に左マウスボタン間をダブルク リックします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Menu] - choice	×
Basic Comment Screen ⊡ Links, in ⊡ Links, out □ -> serial, #12	<ul> <li>Keypad Input</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:         <ul> <li>RESULT_=</li> <li>INIT, Activate Initializatioin</li> <li>RRLL Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul> </li> </ul>
	OK Cancel Help

42. **Key Pad Input** オプションをチェックし、そしてマウスポインタが **Key Pad Input** オプションの次のスペース上にある時に左マウスボタンをクリックします。 "PT-600 Keypad" ウインドウがスクリーンに現れます。



43. 2のボタンをクリックします。これは"2"キーが押された場合、プロセスは choice ノード から serial ノード に移動することを示しています。 "PT-600 Keypad" ウイ ンドウを閉じるために OK をクリックします。

44. Ok をクリックします。

45. マウスポインタを serial ノード に移動し、左マウスボタンをクリックしたままにし ます。マウスポインタを copy\_qty ノード に移動し、左マウスボタンを離します。



46. 新しく作成したリンク上にマウスポインタがある間に左マウスボタンをダブルクリ ックします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Collect] - serial	×
Basic Comment ■ Screen □ Input Device Barcode Lookup File Formula □ Verify Lookup File Qualifier Function Keys ■ Links, in □ Links, out 2 copy_qty, #13	<ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_=</li> <li>EOR, Write Record:</li> <li>INIT, Activate Initializatioin</li> <li>RRLI, Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul>
	OK Cancel Help

Pass Always オプションは、ジョブ が serial ノード の処理を終えたら、 copy\_qty ノ ードの処理を行うことを示しています。

- 47. **OK** をクリックします。
- 48. マウスポインタを copy\_qty ノード に移動し、左マウスボタンをクリックしたまま にします。マウスポインタを serial ノードに移動し、左マウスボタンを離します。



49. 新しく作成したリンク上にマウスポインタがある間に左マウスボタンをダブルクリ ックします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Program] - copy_qty		×
Basic Comment Program Function Keys ∴ Links, in ∴ Links, out Serial, #14	<ul> <li>Keypad Input:</li> <li>Input Data:</li> <li>Verify Node's Status:</li> <li>Pass Always</li> <li>Success</li> <li>Fail</li> <li>Expression:</li> <li>RESULT_=</li> <li>INIT, Activate Initializatioin</li> <li>RRLI, Reset "Repeat Last Input" For All Nodes</li> <li>Alarm</li> <li>* Exclude When Make Job</li> </ul>	
	OK Cancel Help	

Pass Always オプションは、ジョブ が copy\_qty ノード の処理を終えたら、 serial ノード の処理を行うことを示しています。

- 50. EOR オプションをチェックします。
- 51. Ok をクリックします。
- 52. マウスポインタを serial ノード に移動し、左マウスボタンをクリックしたままにし ます。マウスポインタを pNumber ノード に移動して左マウスボタンを離します。



53. マウスポインタが新しく作成したリンク上にある間に、左マウスボタンをダブルク リックします。 "Define Node" ウインドウがスクリーンに現れます。

Basic Comment ⊕ Screen ⊡ Input Initial Value Device Barcode	Keypad Input:     Input Data:     O Input Data:     O Verify Node's Status:     O Pass Always     O Success     O Fail     O Expression:
Lookup File     Formula     Formula     Cookup File     Qualifier     Qualifier     Function Keys     Links, in     Links, out     -> copy_qty, #13     PNumber, #15	RESULT_ =         EOR Write Record:         INIT, Activate Initializatioin         RRLI, Reset "Repeat Last Input" For All Nodes         Alarm         * Exclude When Make Job
	OK Cancel Help

54. **RRLI, Reset "Repeat Last Input" For All Nodes** オプションをチェックします。 この オプションは、pNumber ノード に新しいデータを入力する場合に必要です。 **Keypad Input** オプションをチェックし、そしてマウスポインタが **Keypad Input** オ プションの次のスペースにある間に左マウスボタンをクリックします。 "PT-600 Keypad" ウインドウがスクリーンに現れます。



55. F4 ボタンをクリックします。これはユーザが"F4"キーを押した場合に、プロセスは serial ノード から pNumber ノード に移動することを示します。 "PT-600 Keypad" ウ インドウを閉じるために OK ボタンをクリックします。

56. OK をクリックします。

57. マウスポインタを choice ノード へ移動し、左マウスボタンをクリックしたままにし ます。マウスポインタを startSN ノード へ移動し、左マウスボタンを離します。



- 58. 新しく作成したリンクにマウスポインタがある間に左マウスボタンをダブルクリッ クします。 "Define Node" ウインドウがスクリーンに現れます。
- 59. Keypad Input オプションをチェックし、Keypad Input オプションの次のスペース にマウスポインタがある間に左マウスボタンをクリックします。 "PT-600 Keypad" ウインドウがスクリーンに現れます。.



- 60.1のボタンをクリックします。 これはユーザが"1"のキーを押したら、プロセスはユ ーザが"1"のキーを押した場合、プロセスは choice ノード から startSN ノード へ移 動することを示しています。 "PT-600 Keypad" ウインドウを閉じるために OK ボタ ンをクリックして下さい。
- 61. OK をクリックして下さい。
- 62. マウスポインタを startSN ノード に移動し、左マウスボタンをクリックします。 マウスポインタを counter ノード に移動し、左マウスボタンを離します。



63. マウスポインタか<sup>\*</sup>新しく作成した リンク 上にある間に左マウスボタンをダブルク リックします。 "Define Node" ウインドウがスクリーンに現れます。

Define Node [Collect] - srartSN 	Keypad Input:   Image: Constraint of the second   Image: Constraint of the second <
⊡-Links, out	■ * Exclude When Make Job OK Cancel Help

Pass Always オプションはジョブ が startSN ノード の処理を完了したら、 counter ノード の処理に進むことを示しています。

64. Ok をクリックして下さい。

65. マウスポインタを counter ノード に移動し、左マウスボタンをクリックしたままに します。マウスポインタを add\_sn ノード に移動し、左マウスボタンを離します。



66. マウスポインタが新しく作成した リンク 上にある間に左マウスボタンをダブルクリ ックします。"Define Node" ウインドウがスクリーンに現れます。

Define Node [Collect] - counter		×
Basic Comment Screen Position Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Gualifier Sunction Keys Links, in C-srartSN, #15 Links, out Sadd_sn, #16	Keypad Input:       Input Data:         Verify Node's Status:       Success         Pass Always       Success         Expression:       RESULT_=         EOR, Write Record:       INIT, Activate Initializatioin         RRLI, Reset "Repeat Last Input" For All Nodes         Alarm         * Exclude When Make Job	
		の連加漢

Pass Always オプションはジョブが counter ノードの処理を終えたら add\_sn ノードの処理に進むことを示しています。

- 67. Ok をクリックします。
- 68. マウスカーソルを add\_sn ノード に移動し、マウスボタンをクリックしたままにし ます。マウスポインタを pNumber ノード に移動し、左マウスボタンを離します。



69. 新しく作成したリンク上にマウスポインタがある間に左マウスボタンをダブルクリ ックします。"Define Node" ウインドウがスクリーンに現れます。

Define Node [Program] - add_sn — Basic — Comment — Program — Function Keys — Links, in —	Keypad Input:   Image: Constraint of the second   Image: Constraint of the second <
	OK Cancel Help

Pass Always オプションはジョブ が the add\_sn ノード の処理を終えたら pNumber ノードの処理に進むことを示しています。

- 70. RRLI オプションをチェックします。
- 71. マウスポインタを Program ノード ボタンに移動します。ポインタが Program ノード ボタンにある間に左マウスボタンをクリックします。適当な場所にマウスポインタ を移動します。左マウスボタンをダブルクリックします。Program ノードが作られ ます。

Program ノードを左マウスボタンでダブルクリックします。Basic をクリックし、名 前を Global に変更します。Global Access をチェックします。

- (a) **Program** をクリックします。
- (b) "Define Node [Program] Global" がスクリーンに現れます。

Define Node [Program] - Global Basic Comment Program Function Keys Links, in Links, out	char tempstr[60]; char str[10]; 	
	✓       OK     Cancel     Help	

(c) 右のスペースの中にグローバル変数を指定します。

char tempstr[60];

char str[10];

- 72. Ok をクリックし、 invent.jo としてジョブを保存します。.
- 73. ジョブを保存した後で、マウスポインタをツールバーの Simulate Job または Make Job ボタンに移動します。この在庫管理のジョブをコンパイルするために左マウス ボタンをクリックします。実行可能なファイルを作るプロセスが始まります。



EXE ファィルができた後で、JobGen Plus は Make Job あるいは Simulate Job オプ ションのいずれを選択したかによって、ファイルをポータブル・ターミナルにダウ ンロードするか、あるいは PC のスクリーンに結果を表示するためにシミュレーシ ョンを行います。