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
備品管理							 	 4
第2章	複雑な	JOBGEN	PLUS	アプリ	ケーショ	ン	 	 22
在庫管理							 	 22

第1章 簡単な JobGen Plus アプリケーション

備品管理

このアプリケーションは各部署の現在の備品情報の収集を行うために設計されました。 それぞれの備品は備品番号で表されます。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部署にある備品番号を収集します。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備品管理の部署と備品 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。備品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. 部署の情報を収集する。
 - 2. 部署の情報を検証する。
 - 3. 備品番号を収集する。
 - 4. 備品番号を検証する。
 - 5. データを編集する。
- 作業のフロー備品管理の作業は以下の方法で行われます。
 - 1. 部署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. 部署の情報を検証します。
 - 3. 部署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. 備品番号を収集します。
 - 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部署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. 備品番号を検証します。
- 7. 備品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。

備品管理の作成

上記のリストに従って、備品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。以下のステップは備品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編集メニューからポ ータブル定義を選択し、Setting > スクリーン > 大型フォントをチェックして下さ い。

ポータブル定義		×
Model Setting Keymap	スクリーン: ○ 小型フォント ○ 大型フォント スキャナタイプ: ○ 自動 ○ ペン ブザーボリューム: ○ 高 ○ 中 ○ 低 ○ オフ	
	OK キャンセル ヘルプ	

 マウスポインタを収集ノードボタン 描に移動して下さい。マウスポインタが 収集 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 スポインタを収集ノードが置かれる場所に移動して下さい。

🌾 JobGen Plus - [Job1]		
📆 ファイル(E) 編集(E) ノード(N)	表示(V) ビルド(B) ツール(T) ウインドウ(W) ヘルプ(H)	_ 7 ×
D 🗳 🔒 💭 X 🖻	🖻 🗂 🐟 🗼 🛅 🗔 🖂 🖧 💂	i
7, 3, 🛛 🔓 🔣 🗉) IIIII 🗞 💷 🖾 🖅 📠 🚼 🐾	
Comment Excluded Nodes, Menu		· · · · · · · · · · ·
■ Nodes, Collect Nodes, Math Nodes, Edit	collect1	· · · · · · · · · · ·
Nodes, Erase Nodes, Upload		
Nodes, Program Nodes, Run-Job Links		
Links, Key Conditions Links, Other Conditions		
		· · · · · · · · · · · · · · · · · · ·
」 ヘルプは、F1 を押して下さい。		

左マウスボタンをダブルクリックして下さい。 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。作成するために収集ノードを選択して下さい。

籿	ノード作成:
// ?	コメント (ノード)
E	メニューノード
₩₩	収集ノード
٩	演算ノード
Qb	編集ノード
X	消去ノード
E	アップロードノード
fo	プログラムノード
1	ジョブ実行ノード

マウスボタンが収集ノード 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] - co	llect1	×
Basic Comment	This is a collect node.	
Position Input Initial Value	名前: collect1	
Device Barcode Lookup File Formula ⊡- Verify Lookup File	▶ 開始ノード: ジョブはこのノードから実行を開始します	
Qualifier Function Keys Links, in Links, out	色: ノード -	
	▶ * ジョブ作成時にこのノードを外す	
	OK キャンセル ヘルブ	

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 "Section"に名前を変更します。
- (b) **開始ノード**をチェックします。このノードは備品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、"指定した場所から画面の最後までスクリーンをクリア"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] - Loc	ation	×
Basic Comment Screen Position P	 ブロンブト表示: ● 最後の印字に続いて印字、必要ならスクロールアップ ● 画面に直接印字 ▼ 指定した場所から画面の最後までスクリーンをクリア ■ Row ● Column ● 位置マークを表示しない ● 位置マーク: ● 初右の位置 ● 明在の位置 ● 指定した位置: Row (07): ■ Column ■ 「スワードとして(各入力文字を*で印字) ■ 自動的に生成されたデータについてはプロンプトと入力を表示しない 	
	OK キャンセル ヘルプ	

- (d) "**指定した位置**" 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 "部署コード:"をタ イプします。また、最下行に"F2: データ編集"を入力します。

Define Node [Collect] - Section					
Define Node [Collect] - Ser - Screen - Position - Initial Value - Device - Barcode - Lookup File - Formula - Verify - Lookup File - Coulificat	ction Screen: rows: 16, columns: 4 部署コード: 0000 00000000000000000000000000000000	×			
	F2:データ編集。 OK キャンセル ヘルブ				

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] - Lo	sation	
Basic Comment Position Input Barcode Cookup File Cookup File Cookup File Cualifier Function Keys Links, in Cunks, out	 データタイブ: ○ 数字, 整数 ○ 数字, 倍精度整数 ○ 数字, 浮動小数点 ○ 文字列 ▽ データを最後のレコードとしてセット、そして入力を終了 □ ホワイトスペースで後ろを埋める □ 空のデータを許す 	
	OK キャンセル ヘルブ	

- 4. "**データを最後のレコードとしてセット、そして入力終了**"をチェックします。 これは"部署コード"情報を一回だけ入力すればよいようにし、そして"F1"キーを押 すまで"備品番号"の入力を続けます。
- 5. Verify(ベリファイ、検証) 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] - Lo	cation	×
Basic Comment Screen Position Input Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	 ▼ 長さ: From: 0 To: 64 ● 範囲: From: To: □ ビクチャ: □ ビクチャ: □ ビクチャ: □ 検証パスしなかった場合再入力しない □ 検証パスしなかった場合モラーご者を出さない □ 検証パスしなかった場合エラー記号(文字 *)を印字しない 	
	OK キャンセル ヘルプ	

6. **OK** をクリックして下さい。

次のステップは二番目の収集ノードを作ります。

7. マウスカーソルを収集ノード 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。



- 8. マウスポインタが新しく作った収集(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、 "**指定した場所から画面の最後までスクリ** ーンをクリア" を選択して、 "0" 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) "**指定した位置**"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 "備品番号入力:"をタイプして下さい。また、 最下行に"F1: 部署入力"をタイプします。

Define Node [Collect] - Ass	et_number	×
Screen Position Position Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in C- Section, #1 Links, out -> Asset_number, #5 -> Section, #2	Screen: rows: 16, columns: 4 備品番号入力: ○○ ○○○○○○○○○○○○○○○○○ F1:部署入力○○○○	
	OK キャンセル ヘルプ	

- 11. OK をクリックして下さい。 次のステップはレコードの定義を行います。
- 12.編集 > 新規レコードで新しいレコードを作成し、次に編集 > レコード定義でレコ ードを定義します。

レコード定義		×
ASSETMGT Separator Comment New Record Separator EOR Links Comment	レコード名: ASSETMGT データファイル名: ASSETMGT.TXT ▼ これは標準のレコードまたはデータファイル データノード: レコードフィールド: Asset_number Section ≪	
	OK キャンセル ヘルプ	

レコード名: を ASSETMGT とします。

データファイル名を ASSETMGT.TXT とします。

データノードの Asset_Number と Section を移すために、>> キーを使用します。 OK を押します。 13. マウスポインタを編集ノード ボタン^回に移動します。マウスポインタが編集ノード ボタン^回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 を編集ノード 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。

14. マウスポインタが編集ノード 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				X
Basic Comment Function Keys I Links, in I ← Location, #3 Links, out I ← Location, #4	レコード名: データファイル名: フィールド区切り: 〒 最後のレコードカ	ASSETMGT ASSETMGT.TXT , (Comma) いらブラウズ開始	▼ ▼	
	ОК	キャンセル ヘルプ		

- 15. 以下のオプションを変更します。
 - (a) 名前を "Edit1" から "Edit"に変更します。
 - (b) データファイル名オプションとして ASSETMGT.TXT をタイプします。
 - (c) フィールド区切りについて", (Comma)"を選択します。
- 16. **OK** をクリックします。

次のステップは最初の収集ノード(Section)から二番目の収集ノード(Asset_Number) へのリンクを作ります。

- 17. 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 "Section" 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。
- 18. 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 "Section" 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。



- 19. 次のステップは "Asset_Number" ノード から "Section" ノードへの別なリンクを作 ります。
- 20. 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 "Section" ノード に移動し、左マウスボタ ンを離します。



- 21. 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。
- 22. 以下のオプションを変更します。
 - (a) **キーパッド入力**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-630 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、 "Section" ノードが直ちに処理されることを示しています。

Basic	┌ ᅊ キーパッド入力	PT-630 キーパッド	×
Comment Screen Position Input Device Barcode Lookup File Formula Verify Lookup File Gualifier Function Keys Links, in Links, out Location, #2	 ・ ノードステータ ・ デニバス ・ デニバス ・ デニー ・ RESULT_= ・ EOR, レコード ・ 「 EOR, レコード ・ 「 INIT, 初期化動 ・ RRLI, 全ノード ・ アラーム 「 * ジョブ作成時 	Esc Ent ABC DEF GHI 1 # 2 \$ 3 \$ JKL MNO POR 4 : 5 = 6 / 4 : 5 = 6 / \$ \$ 7 - 8 + 9 * \$<	0K(<u>0</u>) <u>キャンセル(0</u>) キー名: F1 キーコード: 134
	OK	F5 F6 F7 F8 F1 F2 F3 F4	任意キー

- (b) "RRLI,全ノードについてリセット"最後の入力繰り返し""をチェックして下さい。この選択は"Section"ノードの"データをレコードの最後にセット、そして入力終了"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、"Section"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。
- 23. **OK** をクリックします。 次のステップは、最初の収集ノードから編集ノードへのリンクを作成します。
- 24. マウスポインタを "Section" 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。
- 25. 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。



26. 以下のオプションを変更します:

- (a) **キーパッド入力**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-630 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。
- 27. 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"Section" ノードへ移動して、左のマウスボタンを離し ます。



28. 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"常にパス"です。 これは"Edit" ノード が終わった場合、 "Section" 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。マウスポインタをファイル > 名前を付けて保存...に移動して、左マウスボタンをクリックします。

29. ジョブのファイル名として、**ファイル名**のスペース中に"assetmgt.jgp" を入力しま す。**OK** をクリックします。

名前を付けて保存		? 🛛
保存する場所型:	C JOBGEN PLusマニュアル	• 🖬 📩 🖬 •
assetmet invent Job1 Imr_asset Sample tutorial		
ファイル名(<u>N</u>):	assetmet	保存(<u>S</u>)
ファイルの種類(工):	JobGen Plus 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。

30. 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[™]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
Create a self loop link?
<u>((()))</u> (())

はい、をクリックします。

🍣 JobGen Plus – C:¥Documents	s and Settings¥YAN	AShigeru¥My Docum	ents¥Unitech∃	
ファイル(E) 編集(E) ノード(N) 表示()	⊻) ビルド(<u>B</u>) ツール(<u>T</u>)	ウインドウ(W) ヘルプ(H)		
	ē 🛱 🐔 メ		8 県	i
	HH 🗞 💷 💌	1 50 🔁 💫		
👺 C:¥Documents and Settings	¥YAMAShigeru¥My	Documents¥Unitech	日本語化¥JGPlu	
 Job Portable Model: PT-630 Function Keys Records Comment Excluded Nodes, Menu Nodes, Collect Nodes, Edit Nodes, Edit Nodes, Erase Nodes, Upload Nodes, Run-Job Links Links, Other Conditions Links, Other Conditions 	Edit	Section		
ヘルブは、F1 を押して下さい。			NUN	1

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	×
Comment Screen Position Input Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in C- Location, #1 Links, out Screen Links, out Links,	 ▲ 「・キーパッド入力: ○ ハードステータス ○ 常にパス ○ 成功 ○ 失敗 ● ざ式: RESULT_= ■ EOR, レコード書き込み: MSSETMGT ■ INIT, 初期化動作 ■ RRLI, 全ノードについてリセット "最後の入力繰り返し" ■ アラーム ■ * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

EOR、レコード書き込みをチェックします。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を終了しファイルを書き込みます。

32. ノードステータスをチェックし、常にパス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、備品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ジョブ作成ボタンに置き ます。備品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。

部署コード情報を入力した後、F1 キーを押すまで備品番号の入力が続きます。

ESC キーを押すとプログラムを終了します。

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、品番、品番と数量、そして品番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、品番、数量、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。品番だけの在庫情報については、数量フィールドとシリアル 番号フィールドは "0" にセットされます。品番と数量のある在庫情報につ いては、シリアル番号フィールドは "0" にセットされます。品番とシリア ル番号を持つ在庫情報については数量フィールドは "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. 品番を収集
 - 3. 品番を検証
 - 4. 数量のフィールドを0にセット (選択したタイプが品番のみの場合)
 - 5. 数量のフィールドを1にセット (選択したタイプが品番とシリアル番 号の場合)
 - シリアル番号フィールドを0にセット(選択したタイプが品番のみか、 品番と数量の場合)
 - 7. 数量を収集 (選択したタイプが品番と数量の場合)
 - 8. 数量を検証(選択したタイプが品番と数量の場合)
 - 9. シリアル番号がバラバラか、連続かを選択(選択したタイプが品番と シリアル番号の場合)
 - 10. シリアル番号を収集(選択したタイプが品番とシリアル番号の場合)

- 11.シリアル番号を検証(選択したタイプが品番とシリアル番号の場合)
- 12. 最初のシリアル番号を収集 (選択したタイプが品番とシリアル番号で、 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 (選択したタイプが品番とシリアル番号で、 シリアル番号が連続の場合)
- 14. 在庫の全数量を収集 (選択したタイプが品番とシリアル番号、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 (選択したタイプが品番とシリアル番号、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) 品番のみ
 - (b) 品番と数量
 - (c) 品番とシリアル番号
- 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. 品番情報の収集
- 4. ユーザが "終了" キー (ESC) を押したら、ステップ1に進みます。
- 5. 品番情報を検証します。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. 数量フィールドとシリアル番号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. 品番情報を収集します。
- 8. "終了" キー(ESC) が押されたら、ステップ1へ進みます。
- 9. 品番情報を検証します。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. 数量情報を収集します。
- 11. 数量情報を検証します。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. シリアル番号フィールドに"0"を入れます。ステップ7に進みます。
- 13. 品番情報を収集します。

- 14. "終了" キー (ESC) が押された場合、ステップ1に進みます。
- 15. 品番情報を検証します。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) 品番のみ
 - (b) 品番と数量
 - (c) 品番とシリアル番号

|--|

\left JobGen Plus - [Job2]		
🕎 ファイル(E) 編集(E) ノード(N)	表示(V) ビルド(B) ツール(T) ウインドウ(W) ヘルプ(H)	_ 8 ×
D 🖻 🔒 🗐 🕺 🖻	🖻 🖆 ≼ 🛣 🗔 🖾 😽 🗣	Ť
5 , 5, 8, 8, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10	- ++++ 💁 💷 🖾 🖅 📠 🚼 🧏	
🛨 Job		
Becords		
Comment		
- Excluded		
Nodes, Menu		
Nodes, Collect		=
Nodes, Math		
Nodes, Edit		
Nodes, Erase		
Nodes Program		
Nodes, Run-Job		
Links		
- Links, Key Conditions		
Links, Other Conditions		
, ヘルプは、F1 を押して下さい。		

- 1. マウスポインタをメニューノード ボタンに移動します。
- マウスポインタがメニュー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. マウスポインタをメニューノード 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。



 新しいメニューノード が作成されます。マウスポインタがメニューノード にある時 に左マウスボタンをダブルクリックします。"Define Node [Menu]" (メニュー定義)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編集メニューからポータブル定義を選択し、Setting > スクリーン > 大型フォントをチェックして下さい。

Define Node [Menu] - menu	n 🛛 🔀
Basic Comment Screen Function Keys Links, in Links, out	Screen: rows: 16, columns: 4 1. 品番のみ。。。。。 2. 品番と数量。。。 3. 品番とS/N。。。。 $\langle ESC \rangle$ 終了。。。。。 $\overset{\vee}{}$ $\overset{\vee}{}$
	OK キャンセル ヘルプ

- 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、**開始ノード**オプションをチェックします。
 - (d) Function Keys プロパティで、 "終了(ESC)" をチェックします。"終了"キー (ESC) が押された場合、mainMenu ノード が処理されます。

Define Node [Menu] - main	Menu	
Basic Comment Screen Links, in Links, out	もし実行しているノードがトリガ状態の時に同じファンクショ ンキーとリンクしていない場合、、以下に定義される選択され たファンクションキーを押すとこのノードを実行します。 - ファンクションキー: ○ F1 ○ F2 ○ F3 ○ F4 ○ F5 ○ F6 ○ F7 ○ F8 - 矢印キー: ○ 左 ○ 右 ○ 上 ○ 下 その他キー: ○ 終了 (Esc) ○ なし	
	OK キャンセル ヘルプ	

6. **OK** をクリックします。

<u>作業フロー</u>

- (a) 品番のみを選択した場合、ステップ3に進みます。(b) 品番と数量を選択した場合、ステップ7に進みます。(c) 品番と S/N を選択した場合、ステップ13 に進みます。
- 3. 品番情報を収集します。
- 4. "終了"(ESC) キーが押されたら、ステップ1に進みます。
- 5. 品番情報を検証します。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. 数量フィールドとシリアル番号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

- 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、品番のための収集ノードを作成します。マウスポインタが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. 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。



3. 新しい収集(collect1) ノードが作成されました。マウスポインタが収集ノードにある 時に左マウスボタンをダブルクリックします。 "Define Node [Collect]" ウインドウが スクリーンに現れます。



- 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] - pn(Dnly	×
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	Screen: rows: 16, columns: 4 〈ESC〉 メインメニュー 品番: ○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	
	OK キャンセル ヘルプ	_

- (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 $\mathcal{T} \square \mathcal{T} \mathcal{T} = \mathcal{T} \mathcal{T} \mathcal{T}$

Define Node [Collect] - pn	Only	×
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Valifier Links, in Links, out	 ● タイマ ● 日付のみ ● 時刻のみ ● 日付と時刻 ● 日付クイブ: WM/DD/YYYY ▼ ■ スカ無いときに自動トリガ ■ 入力無いときに自動トリガ ■ 入力データ検証のために二回スキャン ■ バーコード読み取り後にスキャナをオフにする 	
	OK キャンセル ヘルプ	

- (e) **キーパッドとスキャナ**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] - pn	Only	X
Basic Comment Screen In Position Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	 ✓ 長さ: From: ③ To: □6 範囲: From: To: □ ビクチャ: ビクチャ: ビクチャ: ビクチャ: ビクチャ: 「 検証パスしなかった場合再入力しない □ 検証パスしなかった場合再入力しない □ 検証パスしなかった場合エラー記号(文字 *)を印字しない 	
	OK キャンセル ヘルブ	

- (g) **長さ**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. プログラムノードボタンにマウスポインタを移動して、ポインタがプログラムノー 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。
- 6. プログラムノード 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。

🍣 JobGen Plus – [C:¥Docume	nts and Settings¥YAMAShigeru¥My Documents¥Unitech	h 🔳 🗖 🔀
📆 ファイル(E) 編集(E) ノード(N) 🕴	表示(V) ビルド(B) ツール(T) ウインドウ(W) ヘルプ(H)	_ 8 ×
	🖻 🖆 💰 🧩 🛅 🗖 🖉 🖧 💂	i
🖬 🕷 🚜 🗉] ### 🗞 @] 🖾 🗊 📠 🖫 🗞	
I ⊞- Job		🔼
er e		
Comment		
Excluded		
🕀 Nodes, Menu	····	
⊡ Nodes, Collect	and a second sec	=
- Nodes, Math Nodes, Edit	· · · · · · · · · · · · · · · · · · ·	
- Nodes, Erase		
- Nodes, Upload		
🛨 Nodes, Program	program]	
Nodes, Run-Job		
Links Linka Kau Canditiana		
Links, Ney Conditions		
		⊻
		>
		NUM

7. 新しく作成したプログラムノード 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] - co	ppy_all	
Basic Comment Frogram Function Keys Links, in 	<pre>copy_all() { strcpy(_pNumber,_pnOnly); _quantity = 0; strcpy(_serial,"0"); } </pre>	
	OK キャンセル ヘルプ	

Cのプログラム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(品番)、 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。収集ノード 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、このプログラムノードは 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。このプログラム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");
}
```

- C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。



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 [Menu]" ウインドウがスクリーンに現れます。

Define Node [Menu] - main	Menu	
Basic Comment Screen Function Keys Links, in ⊡- Links, out Links, out	 ○ キーパッド入力: ○ 入力データ: ○ ○ ノードステータス ○ 成功 ○ 失敗 ○ ボ: ○ ■ EOR, レコード書き込み: ○ ■ INIT, 初期化動作 ■ RRLI, 全ノードについてリセット "最後の入力繰り返し" ■ アラーム ■ * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

- 12. **キーバッド入力**オプションをチェックし、そして**キーバッド入力**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-630 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-630 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。

$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	\mathbf{X}
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	((<u>0)</u> ノセル(<u>C</u>)
7 - 8 + 9 * 49 8?& ; . Cir ¥ 0 ' . ! Esc	.: ı — ⊦":
	1
Alp < > Spc PgDn PgUp PgUp Pwr Up Dn Cmd F5 F6 F7 F8 F1 F2 F3 F4 任意:	+

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 [Collect]" ウインドウがスクリーンに現れます。

Define Node [Collect] - pnO	nly	×
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Formula Cookup File Cookup	 キーパッド入力: ヘカデータ: ・ ノードステータス ・ 常にパス ・ 成功 ・ 失敗 ・ 式: 「RESULT_= ・ EOR, レコード書き込み: ・ EOR, レコード書き込み: ・ 「」 ・ ジョブ作成時に除く 	

常にパスオプションは ジョブ が pnOnly ノード のプロセスを終えたら、copy-all 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 [Program]" ウインドウがスクリーンに現れます。
| Define Node [Program] - co | ppy_all | × |
|---|--|---|
| Basic
Comment
Program
Function Keys
Links, in
Links, out
Links, out
Links, aut | ○ キーパッド入力: ○ 入力データ: ○ ノードステータス ○ 常にパス ○ 成功 ○ 失敗 ○ 式: □ RESULT_ = □ EOR, レコード書き込み: □ NIT, 初期化動作 □ RRLI, 全ノードについてリセット "最後の入力繰り返し" □ アラーム □ * ジョブ作成時に除く | |
| | OK キャンセル ヘルプ | |

常にパス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(End of Record,最終レコード)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(品番と数量)</u>

- 7. 品番情報の収集.
- 8. "終了" キーが押されたら、ステップ1へ進む。
- 9. 品番情報の検証。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. 数量情報を収集。
- 11. 数量を検証。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. シリアル番号フィールドに"0"を入れます。ステップ7に進みます。

JobGen Plus

 マウスポインタを収集ノードボタンに移動することによって品番についての収集ノ ードを作ります。マウスポインタが収集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. 収集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. 新しい収集ノード が作られました。マウスポインタが収集ノード 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] - collect2	
Basic Comment	This is a collect node.
⊡- Input Initial Value Device	名前: collect2
Barcode Lookup File Formula ⊡ Verify Lookup File Qualifier	▶ 開始ノード: ジョブはこのノードから実行を開始します
Function Keys Links, in Links, out	色: <u>ノード</u> ▼
	🥅 * ジョブ作成時にこのノードを外す
	OK キャンセル ヘルプ

- 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] - pn(aty	
Basic - Comment - Screen - Input - Initial Value - Device - Barcode - Lookup File - Formula - Verify - Lookup File - Qualifier - Function Keys - Links, in - Links, out	Screen: rows: 18, columns: 4 〈ESC〉 メインメニュー 品番: 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇	
	OK キャンセル ヘルプ	

- (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 \mathcal{T} ロパティをクリックします。

Define Node [Collect] - pn	Qty	X
Basic Comment Screen Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	 ● タイマー ● 日付のみ ● 時刻のみ ● 日付と時刻 ● 日付タイブ: WM/DD/YYYY ● ■ 入力無いときに自動トリガ ■ 入力データ検証のために二回スキャン ■ バーコード読み取り後にスキャナをオフにする ▼ キーパッド: ■ 数字と + - ▼ 英字 ▼ 記号 ▼ Enterで終了 	
	OK キャンセル ヘルプ	

(e) キーパッドとスキャナ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] - pn	aty	X
Basic Comment Screen ☐ Input ☐ Device Barcode Lookup File Formula ☐ Verify Lookup File Qualifier Function Keys Links, in Links, out	 ✓ 長さ: From: 3 To: 16 範囲: From: To: To: ビクチャ: ビクチャ: ビクチャ: ゲ語いれてしなかった場合再入力しない 検証いれてしなかった場合モーブ音を出さない 検証いれてしなかった場合エラー記号(文字 *)を印字しない 	
	OK キャンセル ヘルプ	

- (g) **長さ**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. 収集ノード 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. マウスポインタが収集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] - co	lect3	×
Basic	Screen: rows: 16, columns: 4	
Screen Position	数量:	
i⊟-Input Initial Value Device		
Barcode Lookup File Formula		
ie-Verify ↓ Lookup File ↓ Qualifier		
Function Keys Links, in Links, out	,	
	OK キャンセル ヘルブ	

(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。数字、整数をチェックします。

Define Node [Collect] - co	llect3	
Basic Comment Screen Position Incut Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, out	 データタイプ: ● 数字,整数 ● 数字,倍精度整数 ● 数字,浮動小数点 ● 文字列 ■ データをレコードの最後にセット、そして入力を終了 ■ ホワイトスペースで後ろを埋める ■ 空のデータを許す 	
	OK キャンセル ヘルブ	

- (e) Input > Device \mathcal{J} ロパティをクリックします。
- (f) **キーパッドとスキャナ**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 (検証)プロパティをクリックして下さい。

Define Node [Collect] - collect3		
■ Basic Comment Screen Position Input □ Initial Value Device Barcode Lookup File Formula Cokup File Cokup File Devity Lookup File Lookup Links, out	 ▼ 長さ: From: 1 To: 5 ● 範囲: From: To: ■ ビクチャ: ■ ビクチャ: ■ ビクチャ: ■ ビクチャ: ■ 検証バスしなかった場合再入力しない ■ 検証バスしなかった場合モラー記号(文字 *)を印字しない 	
	OK キャンセル ヘルプ	

(h) **長さ**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) **範囲**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。
- マウスポインタをプログラムノードボタンに移動します。ポインタがプログラム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。
- プログラム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 – [C:¥Docume	ents and Settings¥YAMAShigeru¥My Documents¥Unitech∃.	💶 🗖 🔀
🕎 ファイル(E) 編集(E) ノード(N)	表示(V) ビルド(B) ツール(I) ウインドウ(W) ヘルプ(H)	_ @ ×
	6 🗗 ≼ 🛣 🗖 🖾 😽 🗣	Ö
5, 5, 8, 8, 12	= +++ 🗞 💷 🖾 🖅 📠 🚼 🧏	
 Job Portable Records Comment Excluded Nodes, Menu Nodes, Collect Nodes, Edit Nodes, Erase Nodes, Program Nodes, Run-Job Links, Key Conditions Links, Other Conditions 	pnOnly pnQty pnQty program6	quantity
		>
		NUM

10.マウスポインタが新しく作成したプログラム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] - co	py_serial	
Basic Comment Function Keys Links, in Links, out Links, out Links, out D-Links, out Links, aut	<pre>copy_serial() { strcpy(_pNumber, _pnQty); strcpy(_serial, "0"); }</pre>	
	OK キャンセル ヘルプ	

11. このプログラム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。 二番目の収集ノード: quantity は quantity (数量)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、プログラムノードは 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。このプログラム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。



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 [Menu]" ウインドウがスクリーンに現れます。

Define Node [Menu] - main	Menu 🛛
Basic Comment Screen Function Keys Links, in ⊡- Links, out > pnOnly, #1 > pnQty, #4	 ● 非一パッド入力: 2 ○ ノードステータス ○ 常にパス ○ 成功 ○ 失敗 ○ 式: ■ RESULT_= ■ EOR, レコード書き込み: ■ INIT, 初期化動作 ■ RRLI, 全ノードについてリセット *最後の入力繰り返し* ■ アラーム ■ * ジョブ作成時に除く
	OK キャンセル ヘルプ

15. **キーバッド入力**オプションをチェックし、マウスポインタを**キーバッド入力**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-630 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-630 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。



- 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 [Collect]" ウインドウがスクリーンに現れます。

Define Node [Collect] - pn	Qty	
Basic Comment Screen Position Input Device Barcode Lookup File Formula Ureify Lookup File Gualifier Function Keys Links, in Links, out Quantity, #5	 ○ キーパッド入力: ○ 入力データ: ○ パンテータス ○ 前にパス ○ 成功 ○ 失敗 ○ 式: ■ EOR, レコード書き込み: ■ EOR, レコード書き込み:	

常にパスオプションは、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 [Collect]" ウインドウがスクリーンに現れます。

Define Node [Collect] - qu	antity	×
Basic Comment Screen Position Input Device Barcode Lookup File Formula Coup File Lookup File File File File File File File File File File Fi	 キーパッド入力: ・ ノードステータス ・ 第にパス ・ 成功 ・ 失敗 ・ 第にパス ・ 成功 ・ 失敗 ・ 第にパス ・ 成功 ・ 失敗 ・ 第ESULT_= ・ EOR, レコード書き込み: ・ EOR, レコード書き込み: ・ INIT, 初期化動作 ・ RRLI, 全ノードについてリセット **最後の入力繰り返し* ・ デラーム ・ * ジョブ作成時に除く 	
	OKキャンセル ヘルブ	

常にパスオプションは、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 [Program]" ウインドウがスクリーンに現れます。

Define Node [Program] - co	ppy_serial	×
Basic Comment Program Function Keys Links, in Links, out Links, out Difference in the second	 ○ キーパッド入力: ○ 入力データ: ○ ノードステータス ○ 常にパス ○ 成功 ○ 失敗 ○ 式: □ RESULT_= □ EOR, レコード書き込み: □ INIT, 初期化動作 □ RRLI, 全ノードについてリセット "最後の入力繰り返し" □ アラーム □ * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

常にパスオプションは、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(End of Record、最終レコード)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。

作業フロー(品番とシリアル番号)

13. 品番情報を収集。

- 14. "終了" キーが押されたら、ステップ1へ進みます。
- 15. 品番情報を検証。 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. 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。そうでなければステップ 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. 数量フィールドに"1"を入れ、データファイルに連続するシリアル番号を加えます。
- 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. シリアル番号を収集します。
- 26. シリアル番号情報を検証します。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. 数量フィールドに"1"を入れます。 ステップ 13 に進みます。

JobGen Plus

- マウスポインタを収集ノードボタンに移動することによって品番についての収集ノ ードを作成します。マウスポインタが収集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. 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。

52



3. 新しい収集ノード が作成されました。マウスポインタが収集ノード 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] - co	llect7	×
Basic Comment	This is a collect node.	
 Screen Position Input Initial Value Device Barcode Lookup File Formula 	名前: collect7	
	□ 開始ノード: ジョブはこのノードから実行を開始します	
Lookup File Qualifier Function Keys Links, in Links, out	色: ノード ▼	
	▶ * ジョブ作成時にこのノードを外す	

- 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] - co	lect7	
Basic Comment Position Input Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	Screen: rows: 16, columns: 4 くESC〉	
	OK キャンセル ヘルプ	

- (c) プロンプト(入力要求)をタイプします。
- (d) Input > Device $\mathcal{T} \Box \mathcal{N} \mathcal{T} + \mathcal{T} \mathcal{T} + \mathcal{T} \mathcal{T} \mathcal{T}$

Define Node [Collect] - co	llect7	
Basic Comment Screen Position Input Device Barcode Lookup File Formula	● 日付のみ ● 時刻のみ ● 日付と時刻 日付タイブ: MM/DD/YYYY ▼	
	 □ 入力無いときに自動トリガ □ 入力データ検証のために二回スキャン □ バーコード読み取り後にスキャナをオフにする 	
	 ✓ キーハッド: ▼ 数字と + - ▼ 英字 ▼ 記号 ▼ Enterで終了 	
	OK キャンセル ヘルプ	

(e) **キーパッドとスキャナ**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、**データをレコードの最後にセット、そして入 力を終了**オプションをチェックします。
- (g) Verify \mathcal{J} ロパティをクリックします。

Define Node [Collect] - pN	umber	
■ Basic Comment Screen Position Input Initial Value Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	 ✓ 長さ: From: 3 To: 16 ● 範囲: From: To: □ ビクチャ: □ ビクチャ: □ ビクチャ: □ ビクチャ: □ 検証パスしなかった場合再入力しない □ 検証パスしなかった場合モラー記号(文字 *)を印字しない 	
	OK キャンセル ヘルブ	

(h) **長さ**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. メニューノード ボタンをクリックします。
- マウスポインタをメニューノード 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。



 新しいメニューノードが作成されました。マウスポインタがメニューノードにある 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] - men	81
Basic Comment	This is a menu node.
Function Keys Links, in Links, out	名前: menu8
	├ 開始ノード: ジョブはこのノードから実行を開始します
	□ * ジョブ作成時にこのノードを外す
	OK キャンセル ヘルプ

- 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] - men	18 👂
Basic Comment Screen Function Keys Links, in Links, out	Screen: rows: 16, columns: 4 シリアル選択 1. 連続 2. 非連続 2. 非連続
	OK キャンセル ヘルプ

- 9. OK をクリックします。
- 10. マウスポインタを収集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。マウスポインタが収集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. 収集ノード 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 が作成されました。マウスポインタが収集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] - co	llect9	×
Basic Comment ⊕ Screen ⊡ Input Initial Value	This is a collect node.	
	名前: collect9	
Barcode Lookup File Formula	□ 開始ノード: ジョブはこのノードから実行を開始します	
 → Verify → Lookup File → Qualifier → Function Keys → Links, in → Links, out 	色: ノード -	
	▶ ※ ジョブ作成時にこのノードを外す	
	OK キャンセル ヘルプ	

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] - Se	rial	×
Basic Comment Diput Initial Value Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	Screen: rows: 16, columns: 4 〈F4〉新品番 シリアル番号 ◇ ◇	
	OK キャンセル ヘルプ	

- (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。

プログラムノードボタンにマウスカーソルを移動します。プログラムノード 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. プログラム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] - co	py_qty	
Basic Comment Frogram Function Keys Links, in Links, out Links, out Links, out Links, art Links, art	<pre>copy_qty() { strcpy(tempstr, _pNumber); strcat(tempstr, "i"); strcat(tempstr, "i"); strcat(tempstr, "j; strcat(tempstr, _serial); AppendRecord(NULL,tempstr); } </pre>	
	OK キャンセル ヘルプ	

15. このプログラムノード 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. マウスポインタを収集ノード ボタンに移動します。マウスポインタが収集ノード ボ タンにある間に左マウスボタンをクリックします。
- 18. 収集ノード 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。



19. 新しい収集ノード が作成されました。マウスポインタが収集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] - co	llect11	×
Basic Comment Screen Input Device Barcode Lookup File Formula Verify Uevify Lookup File Gualifier Function Keys Links, in Links, out	This is a collect node. 名前: collect11 □ 開始ノード: ジョブはこのノードから実行を開始します Ê: ノード ▼ 「 * ジョブ作成時にこのノードを外す	

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。

Define Node [Collect] - sta	rtSN	\mathbf{X}
Basic — Comment — Screen — Input	Screen: rows: 16, columns: 4 S/N入力開始:◇◇◇◇	
Device Barcode Lookup File		
Formula ⊡-Verify Lookup File Oualifier		
	OK キャンセル ヘルプ	

- (c) プロンプト(入力要求)をタイプします。
- (d) Input > Device $\mathcal{T} \square \mathcal{T} \mathcal{T} = \mathcal{T} \mathcal{T} \mathcal{T}$

Define Node [Collect] - sta	rtSN	
Basic Comment Screen Input Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	 ○ 日付のみ ○ 時刻のみ ○ 日付と時刻 日付タイブ: MM/DD/YYYY ▼ ○ スキャナ: □ 入力無いときに自動トリガ □ 入力データ検証のために二回スキャン □ バーコード読み取り後にスキャナをオフにする ○ キーパッド: ○ 数字と + - ▼ 英字 ▼ 記号 ▼ Enterで終了 	
	OK キャンセル ヘルブ	

- (e) **キーパッドとスキャナ**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] - st	artSN	X
Basic Comment Screen Input Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	 ▼ 長さ: From: 1 To: 32 範囲: From: To: ビクチャ: ビクチャ: ビクチャ: ゲ証いえしなかった場合再入力しない 検証いえしなかった場合ビープ音を出さない 「検証いえしなかった場合エラー記号(文字 *)を印字しない 	
	OK キャンセル ヘルプ	

- (g) **長さ**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. マウスポインタを収集ノード 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。



22. マウスポインタが収集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] - co	llect12	
Basic — Comment — Screen — Input — Initial Value — Device — Barcode — Lookup File — Formula — Verify	This is a collect node.	
	名前: collect12	
	□ 開始ノード: ジョブはこのノードから実行を開始します	
Lookup File Qualifier Function Keys Links, in Links, out		
	□ * ジョブ作成時にこのノードを外す	
	OK キャンセル ヘルプ	

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] - co	llect12	
Basic Comment Screen Input Device Barcode Lookup File Formula Verify Cualifier Function Keys Links, in Links, out	Screen: rows: 16, columns: 4	
	OK キャンセル ヘルブ	

- (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。数字、整数をチェックします。

Define Node [Collect] - co	llect12	
Basic Comment Screen Input Device Barcode Lookup File Formula Verify Lookup File Qualifier Function Keys Links, in Links, out	 データタイブ: ・ 数字, 揺動 ・ 数字, 倍積度整数 ・ 数字, 浮動小数点 ・ 文字列 ・ データをレコードの最後にセット、そして入力を終了 「 ホワイトスペースで後ろを埋める 「 空のデータを許す 	
	OK キャンセル ヘルブ	

(e) Input > Device プロパティをクリックします。

Define Node [Collect] - col	lect12
Basic Comment Screen Input Initial Value Device Barcode Lookup File Formula Verify Uevify Lookup File Device Lookup File Lookup File File File File File File File File File File File File File File File File	 ■ タイマ ● 日付のみ ● 時刻のみ ● 日付と時刻 ● 日付タイブ: MM/DD/YYYY ▼ ■ スキャナ: ■ 入力無いときに自動トリガ ■ 入力データ検証のために二回スキャン ■ バーコード読み取り後にスキャナをオフにする ● E-バッド: ■ 数字と + - ● 英字 ● 記号 ■ Enterで終了
	OK キャンセル ヘルプ

(f) **キーバッド**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] - co	llect12	×
Basic — Comment → Screen — Initial Value — Device — Barcode — Lookup File — Formula — Verify — Lookup File — Qualifier — Function Keys — Links, in — Links, out	 ✓ 長さ: From: 1 To: 2 範囲: From: To: ビクチャ: ビクチャ: ビクチャ: 検証パスしなかった場合再入力しない 検証パスしなかった場合モラー記号(文字 *)を印字しない 	
	OK キャンセル ヘルプ	-

長さ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) **範囲**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. マウスポインタをプログラムノード ボタンに移動します。ポインタがプログラムノ 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. プログラムノード 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. 新しく作成したプログラムノード 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] - pr	ogram13	×
Basic Comment Forgram Function Keys Links, in Links, out	<pre>add_sn0 { long int temp; int i; clear_screen0; temp = atol(startSN); for (=0;K_counter;i++) printf("%Id#n",temp); _ltoa(temp.str,10); strcpy(tempstr, pNumber); strcat(tempstr, "'); strcat(tempstr, "'); strcat(tempstr, str); strcat(tempstr, str); temp++; } </pre>	
	OK キャンセル ヘルブ	

29. このプログラムノードは主にデータフィールドに情報を埋めます。ユーザははじめのシリアル番号と保存するシリアル番号数を入力する必要があります。このプログラムノードは 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。このプログラムを定義するには、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] - main	Menu	
Basic Comment Screen Function Keys Links, in ⊡- Links, out > pnOnly, #1 > pnQty, #4 > pNumber, #3	 ○ キーパッド入力: ○ 八ードステータス ○ 常にパス ○ 成功 ○ 失敗 ○ 式: □ RESULT_= □ EOR, レコード書き込み: □ INIT, 初期化動作 □ RRLI, 全ノードについてリセット "最後の入力繰り返し" □ アラーム □ * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

34. **キーバッド入力**オプションをチェックし、**キーバッド入力**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-630 キー パッド" ウインドウがスクリーンに現れます。

PT-630 キーパッド	
Esc Ent ABC DEF GHI 1 # 2 \$ 3 * JKL MNO POR 4 : 5 = 6 / STU VWX YZ_ 7 - 8 + 9 * @?&;.	OK(Q) キャンセル(C) キー名: キーコード:
Clr ¥ 0 ' . ! Esc Alp < > Spc PgDn PgUp Pwr Up Dn Cmd F5 F6 F7 F8 F1 F2 F3 F4	任意キー

- 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。 "PT630 キ ーパッド" 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。



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 [Collect]" ウインドウがスクリーンに現れます。

Define Node [Collect] - pN	umber	×
Basic Comment Screen Input Device Barcode Lookup File Formula Cerify Lookup File Qualifier Function Keys Links, in Choice, #9	 キーパッド入力: ヘカデータ: ・ ノードステータス ・ 常にパス ・ 成功 ・ 失敗 ・ 式: RESULT_= EOR, レコード書き込み: INIT, 初期化動作 RRLI, 全ノードについてリセット "最後の入力繰り返し" アラーム * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

常にパス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 [Menu]" ウインドウがスクリーンに現れます。
| Define Node [Menu] - Choi | ce | × |
|--|-------------------------------|---|
| Basic
- Comment
- Screen
- Function Keys
⊕ Links, in
⊡ Links, out
-> Serial, #10 | ・ | |
| | OK キャンセル ヘルプ | |

42. **キーバッド入力**オプションをチェックし、そしてマウスポインタが **キーバッド入力** 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-630 キーパッド" ウインドウがスクリーンに現れます。

PT-630 キーパッド	
Esc Ent ABC DEF GHI 1 # 2 \$ 3 % JKL MNO PQR 4 : 5 = 6 / STU VWX YZ_ 7 - 8 + 9 * Q: X Q? 2 2 1 1	OK(0) キャンセル(C) キー名: キーコード:
CIr * 0 . : Esc Alp <	任意キー

- 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-630 パッド" 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] - Se	rial	×
Basic Comment Screen Dut Unitial Value Device Barcode Lookup File Formula Verify Lookup File Formula Formula Cookup File Formula Cookup File Formula Cookup File Cookup	 キーパッド入力: 〇入力データ: ・ ノードステータス ・ 常にパス の成功 の失敗 ・ 式: RESULT_= EOR, レコード書き込み: INIT, 初期化動作 RRLI, 全ノードについてリセット "最後の入力繰り返し" アラーム * ジョブ作成時に除く 	
OK キャンセル ヘルプ		

常にパスオプションは、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 [Program]" ウインドウがスクリーンに現れます。

Define Node [Program] - co	ppy_qty	X
Basic Comment Function Keys ⊡ Links, in ⊡ Links, out Serial, #12	 キーパッド入力: ・ ノードステータス ・ 常にパス ・ 成功 ・ 失敗 ・ 式: 「RESULT_= ・ EOR, レコード書き込み: ・ INIT, 初期化動作 ・ RRLI, 全ノードについてリセット "最後の入力繰り返し" ・ アラーム ・ ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

常にパスオプションは、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, レコード書き込み 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。

Define Node [Collect] - Se	rial 🛛 🗙
Basic Comment Comment Screen Injut Initial Value Device Barcode Lookup File Formula Formula Verify Lookup File Gualifier Function Keys Links, in Links, out -> copy_qty, #11 -> pNumber, #13	 ○ キーパッド入力: ○ 入力データ: ○ ノードステータス ○ 常にパス ○ 成功 ○ 失敗 ○ 式: □ RESULT_= □ EOR, レコード書き込み: □ NIT, 初期化動作 ○ 所LI, 至ノードについてリセット "最後の入力繰り返し" □ アラーム □ * ジョブ作成時に除く
	OK キャンセル ヘルブ

54. RRLI, **全ノードについてリセット "最後の入力繰り返し**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。**キーパッド入力**オプションをチェックし、そしてマウスポインタが **キーパッ** **ド入力**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-630 キーパッド" 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-630 キーパッ ド" 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. **キーバッド入力**オプションをチェックし、**キーバッド入力**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-630 キーパ ッド" 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-630 キーパッド" 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] - sta	artSN 🛛	<
Basic Comment Screen Input Device Barcode Lookup File Formula Cerify Lookup File Formula Formula Coulifier Links, in Counter, #15	 キーパッド入力: ・ ノードステータス ・ 常にパス ・ 成功 ・ 失敗 ・ 第にパス ・ 成功 ・ 失敗 ・ 式: 「RESULT_= EOR, レコード書き込み: INIT, 初期化動作 RRLI, 全ノードについてリセット **最後の入力繰り返し* アラーム * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

常にパス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] - Co	unter	×
Basic Comment Screen Input Device Barcode Lookup File Formula Cokup File Lookup File Qualifier Function Keys Links, in Links, out Links, out Links, out	 ○ キーパッド入力: ○ 入力データ: ○ アンテータス ○ 常にパス ○ 成功 ○ 失敗 ○ 式: □ RESULT_= □ EOR, レコード書き込み: □ INIT, 初期化動作 □ RRLI, 全ノードについてリセット "最後の入力繰り返し" □ アラーム □ * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

常にパス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] - ad	ld_sn 🔀
Basic Comment Program Function Keys ⊕- Links, in Links, out <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Program</u> <u>Pro</u>	 キーパッド入力: ・ ノードステータス ・ 常にパス ・ 成功 ・ 失敗 ・ 式: 「RESULT_= ・ EOR, レコード書き込み: ・ EOR, レコード書き込み: ・ INIT, 初期化動作 ・ RRLI, 全ノードについてリセット **最後の入力繰り返し* ・ デラーム ・ ジョブ作成時に除く
	OK キャンセル ヘルプ

常にパスオプションはジョブ が 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. EOR(End of Record,最終レコード)オプションをチェックして下さい。
- 72. マウスポインタをプログラムノードボタンに移動します。ポインタがプログラム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。プログラムノードが作られます。
- 73. レコードの定義

編集メニューから新規レコードを選択します。



「はい」をクリックして新しいレコードを作成します。



OK をクリックします。



編集メニューからレコード定義を選択します。



レコード定義			
 New Record Separator Comment tutorial Separator EOR Links Comment 	レコード名: [tutorial データファイル名: [tutorial.t マ これは標準のレコードまた]。 データノード: Counter pnOnly pnQty startSN	xt まデータファイル レコードフィールド: PNumber quantity serial	
OK キャンセル ヘルプ			

レコード名とデータファイル名を入力します。

データフィールドからデータレコードを含むノード名、pNumber, quantity と serial を>> をクリックしてレコードフィールドの枠中に移動します。

Copy_all から pnOnly のリンクをダブルクリックします。EOR フィールドの後ろのレコ ード名を選択します。

Define Node [Program] - co	ppy_all	
Basic Comment Program ⊡-Links, in ⊡-Links, out ∴-> pnOnly, #8	 ○ キーパッド入力: ○ 入力データ: ○ ノードステータス ○ 前にパス ○ 成功 ○ 失敗 ○ 式: □ RESULT_= ○ EOR, レコード書き込み: [utorial] □ INIT, 初期化動作 □ RRLI, 全ノードについてリセット "最後の入力繰り返し" □ アラーム ■ * ジョブ作成時に除く 	
	OK キャンセル ヘルブ	

copy_qty から serial へのリンクをダブルクリックします。EOR フィールドの後ろのレコ ード名を選択します。

Define Node [Program] - co	py_qty	×
Basic Comment Program Function Keys ⊥inks, in Links, out Serial, ‡12	 ○ キーパッド入力: ○ 入力データ: ○ パンテータス ○ 常にパス ○ 成功 ○ 失敗 ○ 式: □ RESULT_= □ EOR, レコード書き込み: tutorial □ INIT, 初期化動作 □ RRLI, 全ノードについてリセット "最後の入力繰り返し" □ アラーム □ * ジョブ作成時に除く 	
	OK キャンセル ヘルプ	

同様に、copy_serial から pNqty、add_sn から pNumber へのリンクについても EOR をセットし、レコード名を選択します。

74. プログラム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。グローバルアクセスをチェックします。

(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] - pr	ogram14	\mathbf{X}
Basic Comment Function Keys Links, in Links, out	char tempstr[60]; char str[10];	
	OK キャンセル ヘルプ	

(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];

- 75. **Ok**をクリックし、invent.jgpとしてジョブを保存します。
- 76. ジョブを保存した後で、マウスポインタをツールバーのジョブシュミレーションまたはジョブ作成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。