

Multi LABELIST V5

練習マニュアル

—機能別解説編—

第11版

株式会社サトー

2020年9月23日

ご注意

- 本マニュアルの一部または全部を弊社の許可なく複写・複製することは、その形態を問わず禁じます。
 - 本マニュアルの内容は、訂正・改善のため予告なく変更することがあります。
 - 本マニュアルを運用した結果の影響については責任を負いかねますのでご了承下さい。
 - 本マニュアルの内容については万全を期しておりますが、万一ご不審な点やお気づきの点がございましたら、弊社までご連絡ください。
-
- SATO、Multi LABELIST は、サトーホールディングス株式会社の登録商標または商標です。
 - Microsoft、Windows は、米国マイクロソフト社の登録商標です。
 - その他記載されている会社名、製品名は各社の登録商標または商標です。

目次

ご注意	2
1：税計算機能を使ってみよう	5
1. 税計算機能とは	5
2. 税計算の種類と税金に対する処理	5
■税編集の種類	5
■税金に対する端数処理	5
■税率の指定方法	5
■消費税編集例	6
3. 税率設定方法	6
4. 税編集の設定	8
5. 編集・表示機能	9
■入力変数の桁数	11
2：GS1-128 を作成してみよう	12
1. GS1-128 作成機能とは	12
2. 変数の作成	12
■カスタム構成	14
■定型パターン構成	16
3. GS1-128 のレイアウト作成	17
3：履歴ファイル	18
1. 履歴ファイルとは	18
2. 履歴ファイルの保存先設定	18
3. 履歴ファイルの作成	19
4：ML コンバート	23
1. ML コンバートとは	23
2. レイアウトの指定	23
3. オプション設定（プリンタ割付、デザイン設定）	24
■プリンタ割付	25
■デザイン設定	26
4. コンバートを行う	27
5. コンバートを行う（コンテキストメニュー）	29
5：連番を判別値毎に保存してみよう	31
1. 連番を判別値毎に保存するとは	31
2. 連番変数の設定	31
3. 連番保存値修正手順	33
6：オブジェクトの印字条件設定	34
1. オブジェクトの印字条件設定とは	34
2. オブジェクトの印字条件設定に必要な事前設定	35
3. オブジェクト印字有無の条件設定	36
4. 印字確認	38

7：発行枚数計算機能.....	39
1. 発行枚数計算機能とは.....	39
2. 変数の作成.....	39
3. 発行枚数計算の条件設定.....	41
4. 発行枚数計算結果確認.....	48
8：アドバンスモードを使ってみよう.....	49
1. アドバンスモードとは.....	49
2. アドバンスモードの設定方法.....	49
3. アドバンスモードの機能.....	51
■デザインオブジェクト.....	51
■変数.....	52
9：GS1 DataBar 及び GS1 合成シンボル印字機能.....	53
1. GS1 DataBar 及び GS1 合成シンボル印字機能とは.....	53
2. レイアウトの設計方法.....	53
■GS1 DataBar 指定.....	53
■GS1 合成シンボル指定.....	54
3. 補足説明.....	57
■GS1 合成シンボルでの FNC1 指定.....	57
■GS1-128 Composite(CC-A/CC-B)及び(CC-C)のバー高さ指定.....	57
■データに関する制限について.....	58
■読み取り確認のお願い.....	58
10：バイナリ情報設定機能（二次元コード）.....	59
1. バイナリ情報設定機能とは.....	59
2. レイアウトの設計方法.....	60
■結合変数で作成する.....	60
■固定変数で作成する.....	63
11：RFID 書込み機能を使ってみよう.....	66
1. RFID 書込み機能とは.....	66
2. ICタグの種類と容量.....	66
3. RFIDに書込んでみよう.....	67
4. その他のICタグへの書込み.....	68
12：条件変数を使ってみよう.....	70
1. 条件変数とは.....	70
2. 条件変数の作成.....	70
3. オブジェクトの作成.....	75
4. 印字確認.....	79
13：GS1 DataMatrix を使ってみよう.....	80
1. GS1 DataMatrix とは.....	80
2. 結合変数の作成.....	80
■可変桁の末尾に FNC1 を設定する場合.....	83
3. オブジェクトの作成.....	84

1：税計算機能を使ってみよう

1. 税計算機能とは

入力した金額に対して税計算を行い、「税込み」「税引き」「外税」「内税」のいずれかの値を出力できる機能です。

「内税」と「外税」は、価格に税が含まれているか否かで呼び方が変わります。

例えば、¥100という価格に税が含まれている場合¥7が税（内税）で¥93が本体価格ということになります。逆に¥100に税が含まれていない場合、この本体価格に税率をかけた¥8が税（外税）で¥108が税込価格となります。

※税率8%、1円未満を切り捨てと仮定した場合の例です。

2. 税計算の種類と税金に対する処理

■税編集の種類

税込み	：指定税率にて外税を求め、入力値に加算します。
税引き	：指定税率にて内税を求め、入力値から減算します。
外税金	：指定税率にて外税を求めます。
内税金	：指定税率にて内税を求めます。

■税金に対する端数処理

- 単位
 - 1円未満：対象となる位を1円未満とします。
 - 1円の位：対象となる位を1円の位とします。
 - 10円の位：対象となる位を10円の位とします。
 - 100円の位：対象となる位を100円の位とします。
 - 1000円の位：対象となる位を1000円の位とします。
- 端数計算
 - 切り捨て：対象となる位に対して切り捨て処理を行います。
 - 切り上げ：対象となる位に対して切り上げ処理を行います。
 - 四捨五入：対象となる位に対して四捨五入処理を行います。
 - 五捨六入：対象となる位に対して五捨六入処理を行います。

※税編集で求めた結果に対する端数処理ではなく、税額に対する端数処理です。

■税率の指定方法

ユーザ税率	：ML メンテナンスで設定した税率を使用します。
固定値	：変数の編集画面で直接入力した値を税率とします。
変数値	：入力変数に指定した値を税率とします。

■消費税編集例

税編集の設定	入力	税金	編集後税金	最終的な出力
種類 : 外税金 端数 : 1 円未満 切り捨て 税率 : 8%	19899	1591.92	1591	1591
種類 : 税込み 端数 : 小数第 2 位の位 切り上げ 税率 : 8%	19899	1591.92	1591.92	21490.92
種類 : 内税金 端数 : 1 円未満 四捨五入 税率 : 10%	2955	268.63...	269	269
種類 : 税引き 端数 : 10 円の位 五捨六入 税率 : 10%	2955	268.63...	270	2655

3. 税率設定方法

まず初めにユーザ税率の設定を行います。

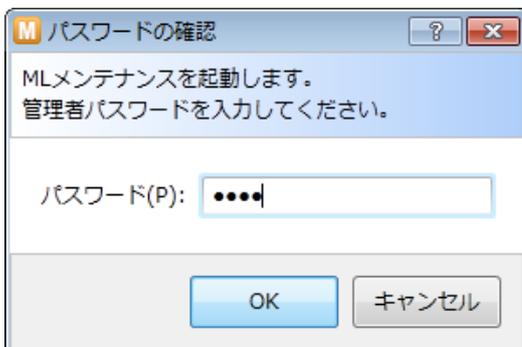
Windows スタートメニュー>すべてのプログラム>Multi LABELIST V5 の

 MLメンテナンス を選択します。

パスワードを聞いてきますので、「user」と入力してください。

画面上では「●」で表示されます。

「OK」ボタンを選択すると ML メンテナンスの画面が表示されます。



「税率設定」を押して消費税率設定画面を表示します。

No.	表示名	税率(%)	発行時指定
1		0	<input type="checkbox"/>
2		0	<input type="checkbox"/>
3		0	
4		0	
5		0	
6		0	
7		0	
8		0	
9		0	
10		0	
11		0	
12		0	
13		0	
14		0	
15		0	
16		0	
17		0	
18		0	

税率は20パターンまで登録できます。

表示名、税率を入力の上「OK」ボタンをクリックしてください。

No.	表示名	税率(%)	発行時指定
1	消費税 8%	8	<input type="checkbox"/>
2	消費税 10%	10	<input type="checkbox"/>
3		0	
4		0	
5		0	
6		0	
7		0	
8		0	
9		0	
10		0	
11		0	
12		0	
13		0	
14		0	
15		0	
16		0	
17		0	
18		0	

これで税率の設定は完了です。

4. 税編集の設定

ML デザインよりレイアウト新規作成をし、ローカル変数追加を行います。
最初にローカル変数画面を表示させます。

変数設定画面の「入力」を指定し、「編集パラメータ」タブにある税編集項目を以下のように設定します。

変数の設定を行います。

入力変数種類: 文字

順序	編集内容	設定内容	設定詳細
1	テーブル変換	なし	...
2	改行削除	なし	▼
3	指定文字削除	なし	▼
4	税編集	あり	種類:税込み, 計算:[切り捨て,1円未満], 税率:[標...
5	カンマ編集		
6	通貨編集		
7	1文字補填		
8	桁寄せ編集		
9	置換		
10	前ゼロ補填		

税編集を行う(T)

種類: 税込み

計算
計算方法: 切り捨て

単位: 1円未満

税率
消費税 8% 8.0

対象文字:

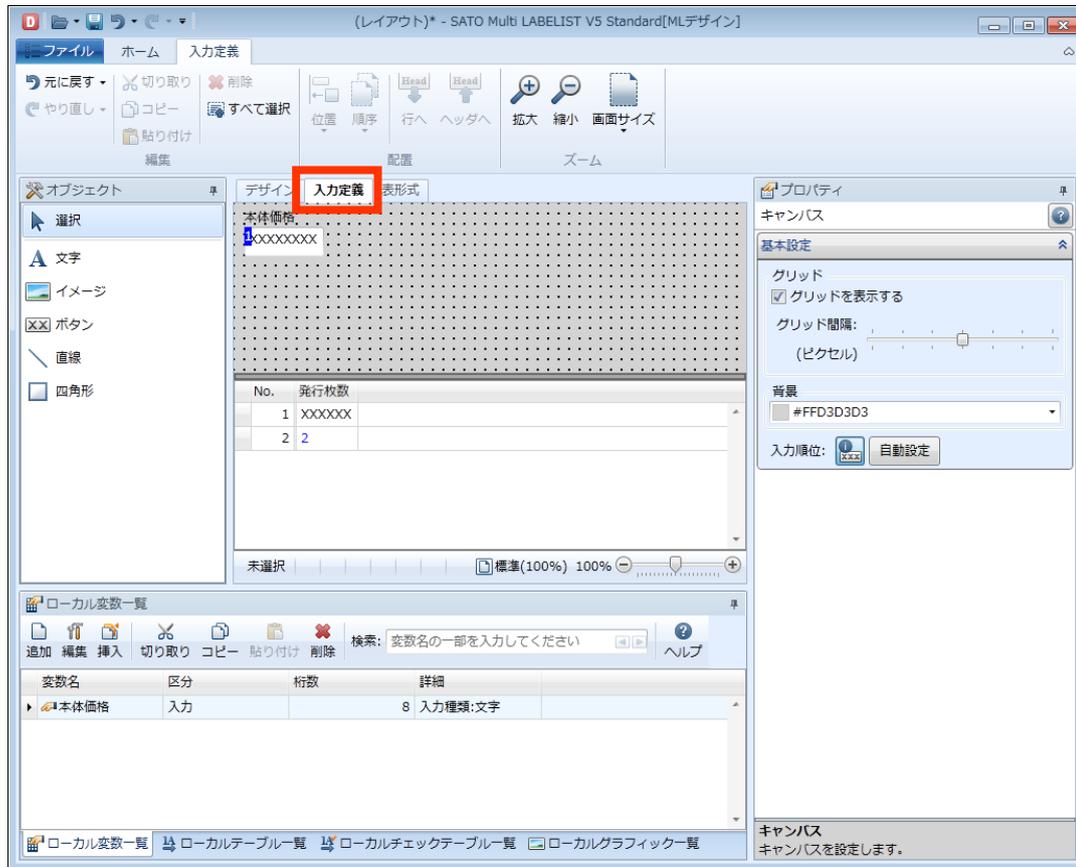
変数名: 本体価格

桁数: 8

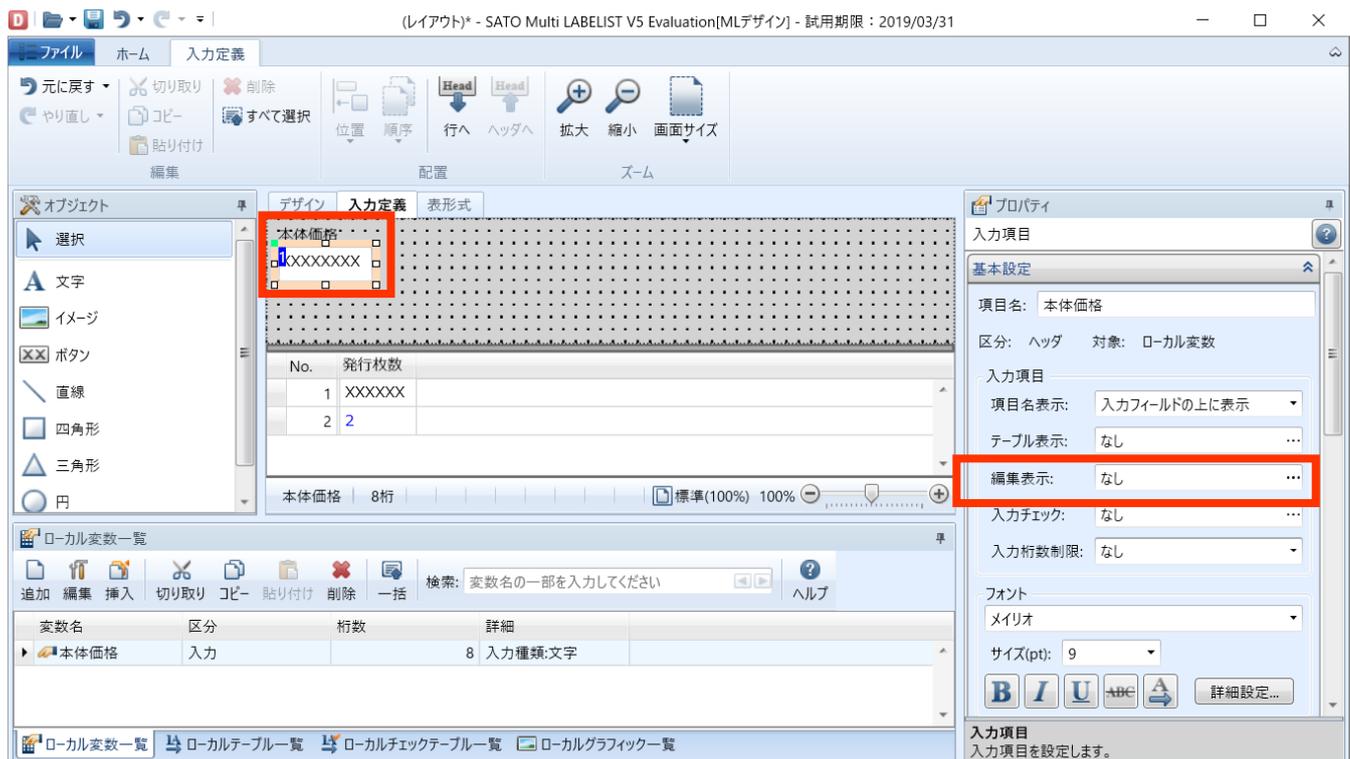
変数名、桁数を入力の上変数を登録し、レイアウト上に配置します。
変数で設定した税編集は、印刷結果に反映されます。

5. 編集・表示機能

ここでは、発行画面での表示の設定を行います。
「入力定義」タブで画面を切り替えます。



「本体価格」項目をクリックし、右側の「基本設定」ウィンドウの「編集表示」設定を行います。



「編集表示」画面が開くので、「編集表示を行う」を有効にします。

「編集設定」項目の「税編集」、「カンマ編集」「通貨編集」「桁寄せ編集」の設定を行います。

編集表示

最初に編集表示の有無を選択します。編集表示を行う場合は、表示設定と編集設定を行います。

編集表示を行う(E)

表示設定

編集表示位置
 入力フィールド内に表示する(F)
 上
 左 入力フィールド 右
 下

フォント
 メイリオ
 サイズ(pt): 9
 B I U ABC A 詳細設定

編集設定

編集種別: 通貨カンマ桁寄せ編集

詳細設定

順序	編集内容	設定内容	設定詳細
1	税編集	あり	種類:税込み, 計算:[切り捨て,1円未満], 税率:[標...
2	カンマ編集	半角	
3	通貨編集	あり	文字:¥, ゼロ空欄:無効
4	桁寄せ編集	右	
5	前スペース編集	なし	
6	前ゼロ編集	なし	

税編集を行う(T)
 種類: 税込み
 計算
 計算方法: 切り捨て
 単位: 1円未満
 税率
 消費税 8% 8.0

OK キャンセル

編集設定

編集種別: 通貨カンマ桁寄せ編集

詳細設定

順序	編集内容	設定内容	設定詳細
1	税編集	あり	種類:税込み, 計算:[切り捨て,1円未満], 税率:[標...
2	カンマ編集	半角	
3	通貨編集	あり	文字:¥, ゼロ空欄:無効
4	桁寄せ編集	右	
5	前スペース編集	なし	
6	前ゼロ編集	なし	

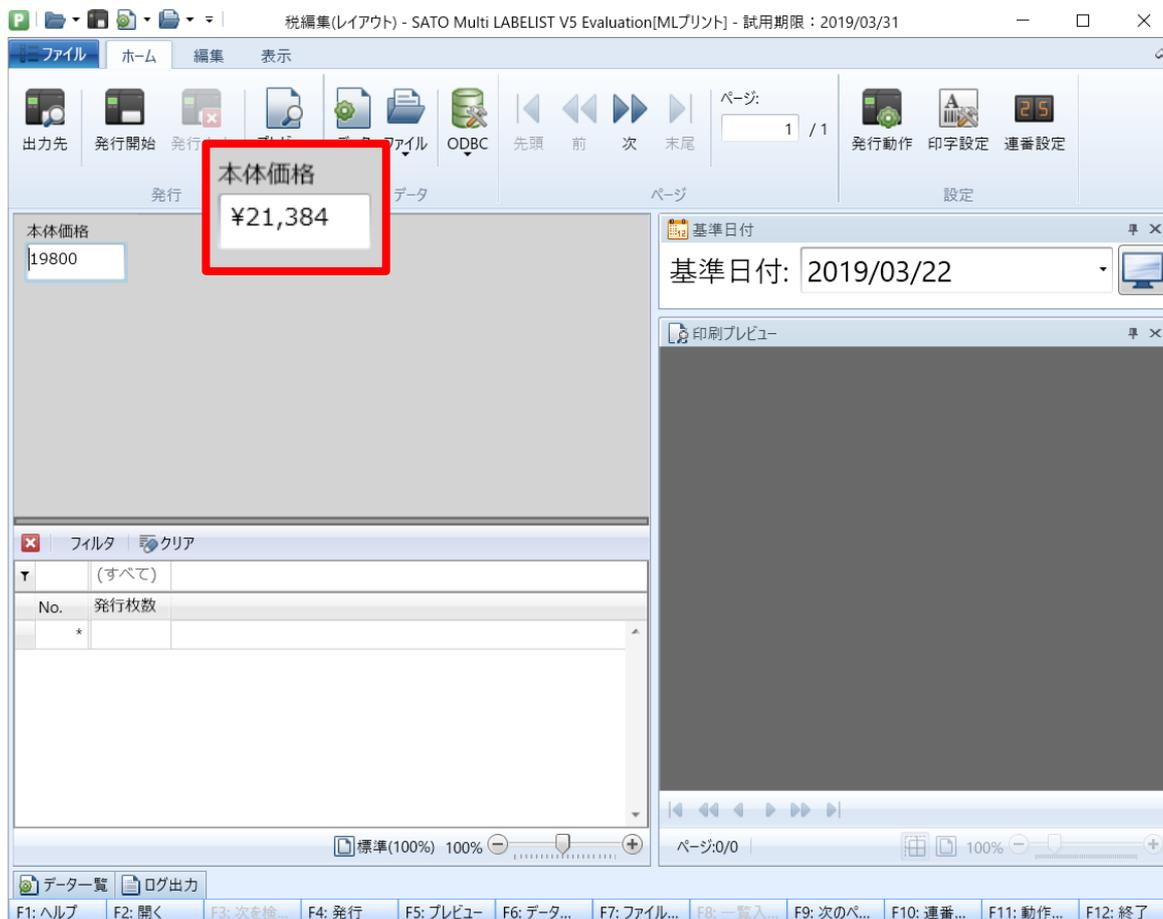
設定が終わったら、レイアウトを保存してML デザインを終了します。

ML プリントで、先ほど作成したレイアウトを開き、「本体価格」欄にデータ入力を行うと編集設定をした値に表示が切り替わります。

■入力変数の桁数

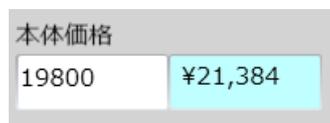
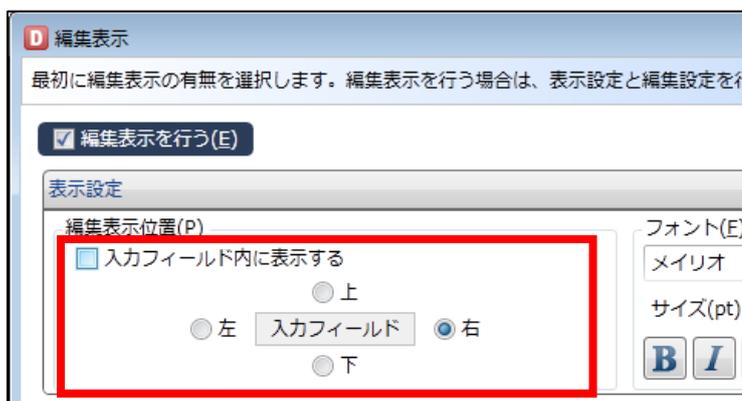
編集表示を利用する場合、「¥」「,」「」も1桁として扱われるため、あらかじめ編集後の桁数を考慮して入力項目を作成する必要があります。

例として、本体価格「19800」を入力して、税込み価格「¥21,384」を表示するには、変数の桁数を7桁以上に設定する必要があります。



■編集結果の表示位置

編集結果を入力フィールド内ではなく、別項目に並べて表示したい場合は、ML デザインの「入力定義」タブで編集表示画面を開き、「編集表示位置」の「入力フィールド内に表示する」を無効にしてください。



以上で「[1：税計算機能を使ってみよう](#)」は終了です。

2：GS1-128 を作成してみよう

1. GS1-128 作成機能とは

「GS1-128」は商品管理、物流管理、業務管理等、企業間商取引システムの中で、様々な管理データにアプリケーション識別子(AI)を付加した構成のCODE128（バーコード）です。

「GS1-128 作成機能」とは、アプリケーション識別子（AI）を一覧から選び、印字する変数との紐付けを行うことで容易にGS1-128が作成できる機能です。「解説文字のアプリケーション識別子（AI）へのカッコ自動表記」「バーコードが最も短くなるようコードタイプの自動切換え」「入力の桁数・及び入力値チェック機能」など便利な機能を多数搭載しております。

2. 変数の作成

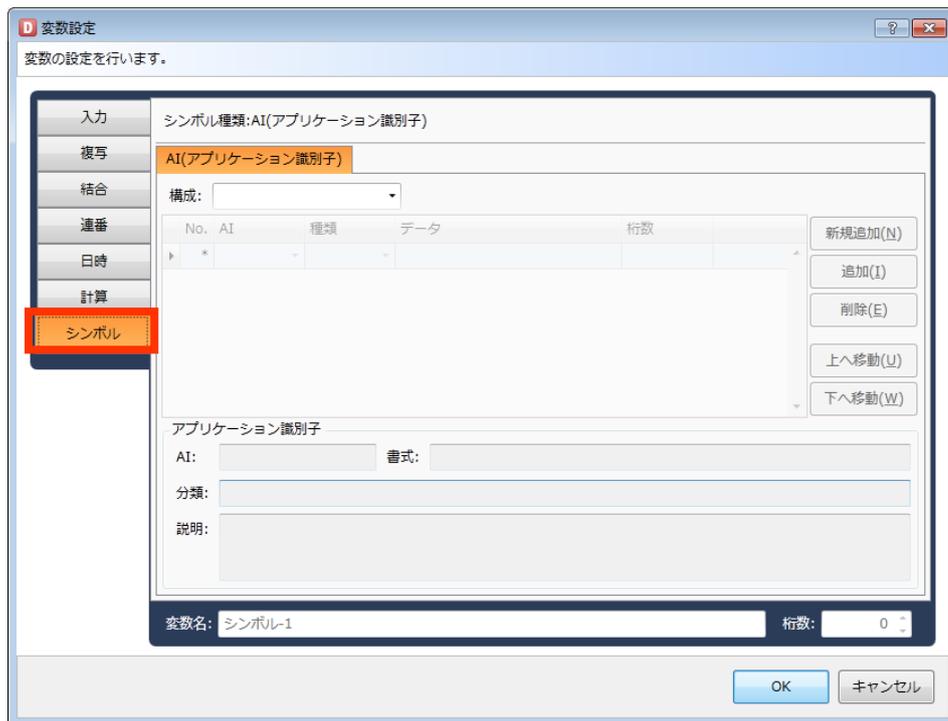
初めにバーコードに組み込む為のデータを作成しましょう。

今回は、「商品コード」「品質保持期限」「数量」「ロット No.」を含むGS1-128バーコードを作成する場合を例に説明します。

まず、MLデザインであらかじめGS1-128に組み込む変数を作成してください。

ローカル変数一覧					
変数名	区分	桁数	詳細		
商品コード	入力	14	入力種類:文字		
品質保持期限	入力	6	入力種類:文字		
数量	入力	8	入力種類:文字		
ロットNo.	入力	3	入力種類:文字		

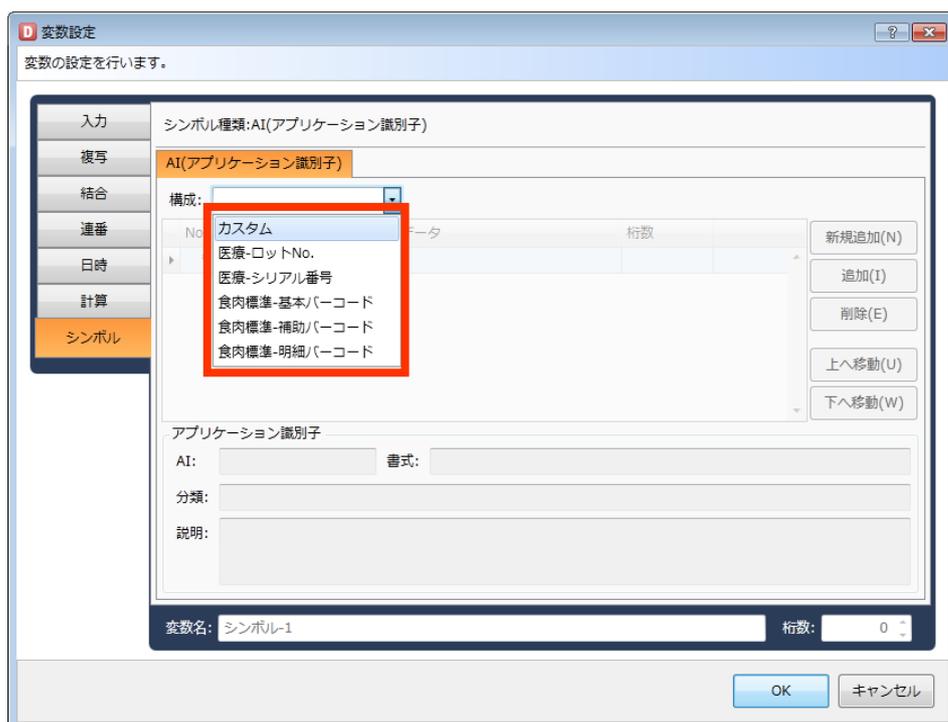
次に、これらの変数を組み込んだ GS1-128 変数を作ります。
変数の種類を「シンボル」を選択します。



GS1-128 の構成を選択します。

「構成」の種類には、自由に構成を編集することができる「カスタム」の他に、医療や食品分野で使用されている定型パターンも指定することができます。

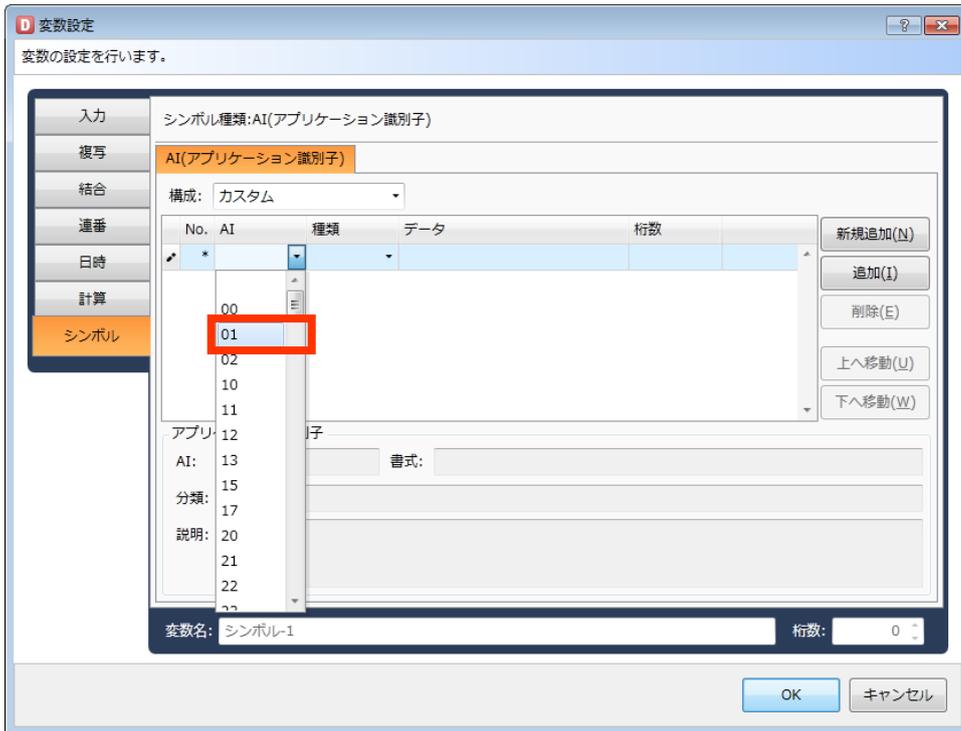
カスタムを選択した場合は次ページの「カスタム構成」へ、それ以外を選択した場合は「定型パターン構成」へお進み下さい。



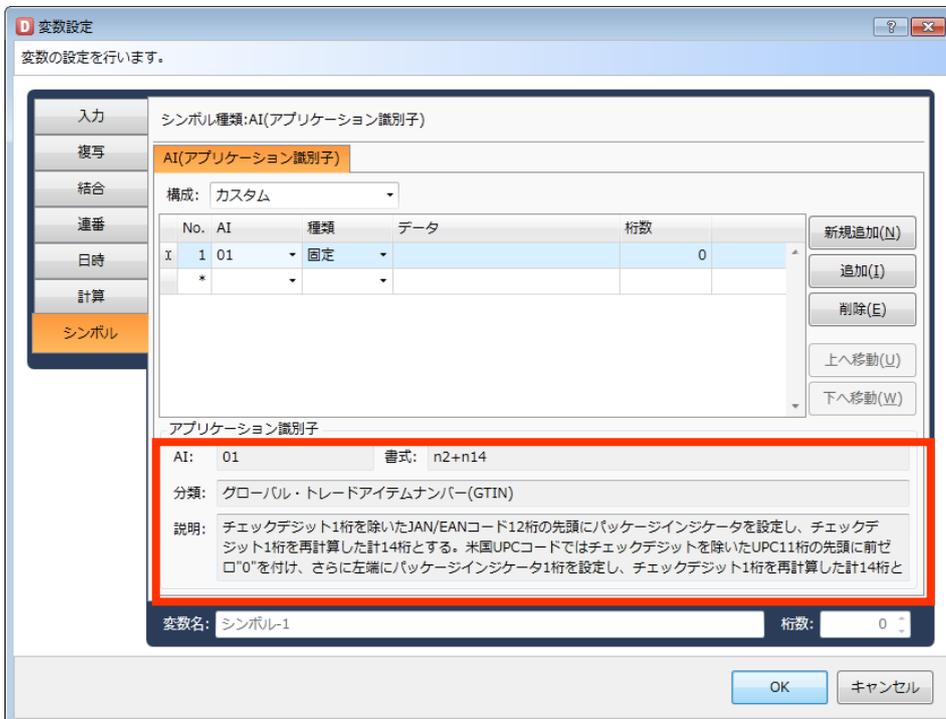
■カスタム構成

まず、商品コードのA Iを組み込みましょう。

AIの項目をクリックすると一覧が表示されます。



選択したAIの書式、分類、説明が表示されます。



AIの「書式」については、次の決まりがあります。

書式: n3+an...30

入力文字として「n」は数値、「a」は英字、「an」は英数字が利用可能であることを示します。記号の後の数値は入力可能な桁数を示し、「...」は桁数が可変になることを示します。

上記の例「n3+an...30」は、数値が固定3桁と英数字が可変30桁で入力可能なことを示します。

AI選択後、「種類：変数」にし、予め登録しておいた変数の中から「商品コード」を指定します。



同様の操作で必要なAIとデータをすべて設定します。

設定が終わったら変数名に「GS1-128」と指定し、「OK」ボタンをクリックして変数を登録します。各変数とAIの組み合わせは次の通りです。

AI	変数
17	品質保持期限
30	数量
10	ロットNo.

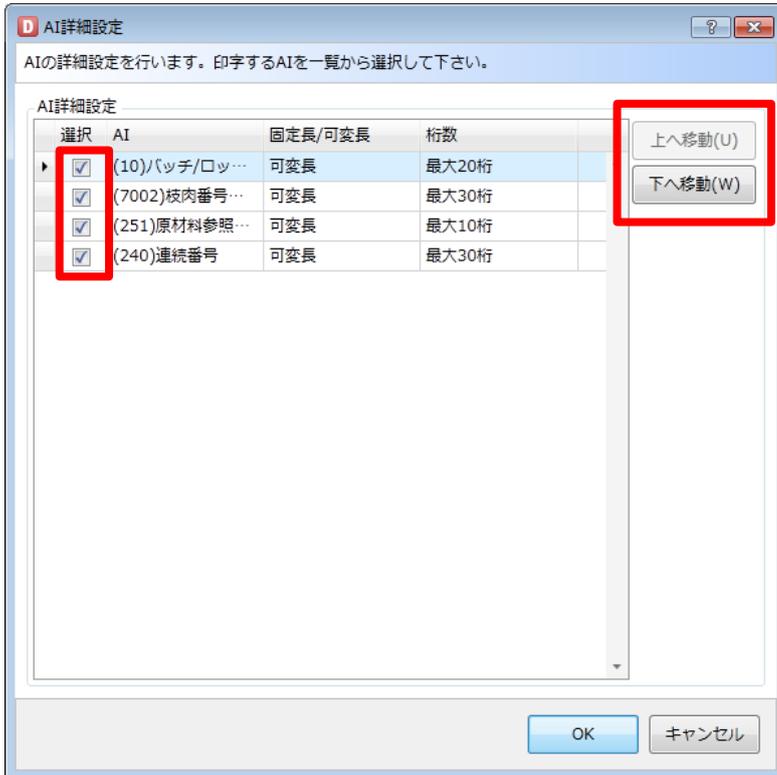


以上で「[カスタム構成](#)」は終了です。

■定型パターン構成

構成で「食肉標準-補助バーコード」を選択した場合のみ下のような画面が表示されます。

印字したい項目にチェックを入れてください。項目順序の入替えも可能です。



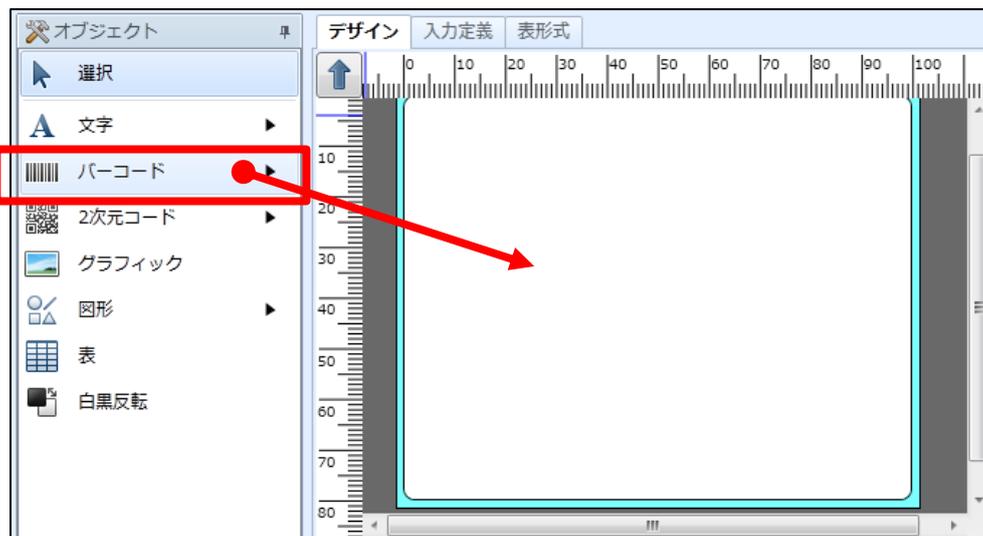
選択した定型パターンに応じて、AIと変数が自動作成されます。自動生成された変数から既存の変数に変更することもできます。



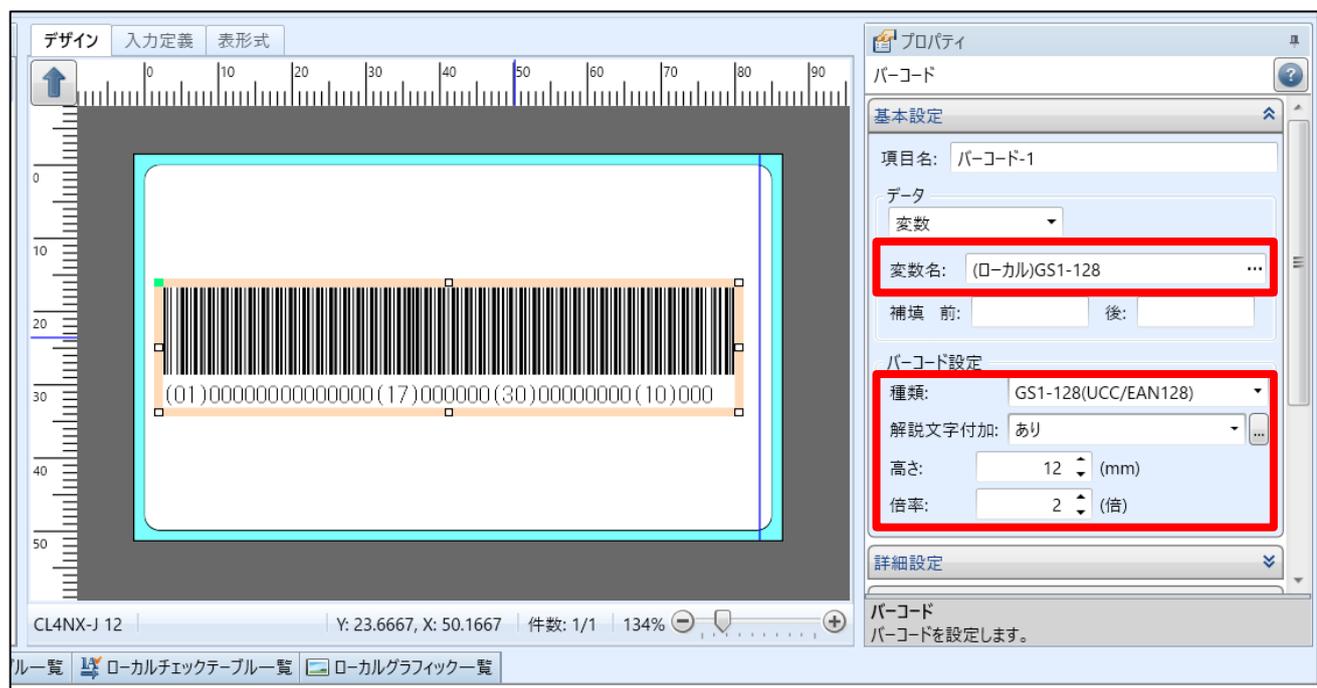
以上で「[定型パターン構成](#)」は終了です。

3. GS1-128 のレイアウト作成

さあ、それでは登録した GS1-128 変数を、バーコードでレイアウト上に作成しましょう。まず、バーコード作成のアイコンをクリックして、デザイン画面に配置してください。



データウィンドウのデータ種類を「変数」にして、変数名をクリックし、先ほど作成した GS1-128 用の変数を選択します。合わせて、バーコードの種類を「GS1-128(UCC/EAN128)」、解説文字「あり」、倍率「2(倍)」以上に設定します。



これで、GS1-128 をレイアウト上に作成する事が出来ました。

印字したバーコードがスキャナで読めるか確認してみましょう。

※バーコードが読めない場合、バーコードの倍率を大きくすると解消できることがあります。

以上で「[2 : GS1-128 を作成してみよう](#)」は終了です。

3：履歴ファイル

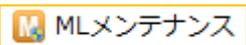
1. 履歴ファイルとは

レイアウトおよび振分発行の発行履歴と、カウンタ項目の値をカウント履歴として蓄積したファイルです。蓄積させたデータはテキスト形式ですので、Excel 等で取込んで編集することができます。

2. 履歴ファイルの保存先設定

まず、発行履歴の保存先を設定しましょう。

Windows スタートメニュー>すべてのプログラム>Multi LABELIST V5 の

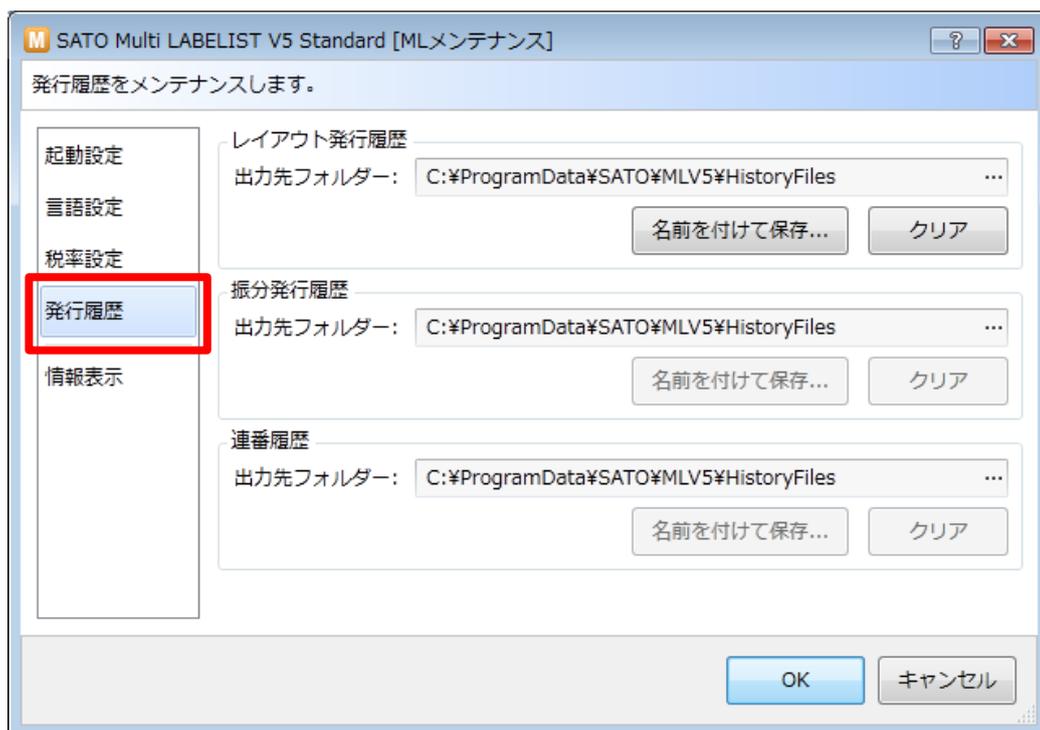
 MLメンテナンス を選択します。

パスワードを聞いてきますので、「user」と入力してください。画面上では「●」で表示されます。「OK」ボタンを選択すると ML メンテナンスの画面が表示されます。

発行履歴を選択し、各発行履歴の出力先フォルダを必要に応じて変更します。

「名前を付けて保存」で、既存の履歴ファイルを別名で保存することができます。

「クリア」ボタンでは、保存している発行履歴をクリアします。

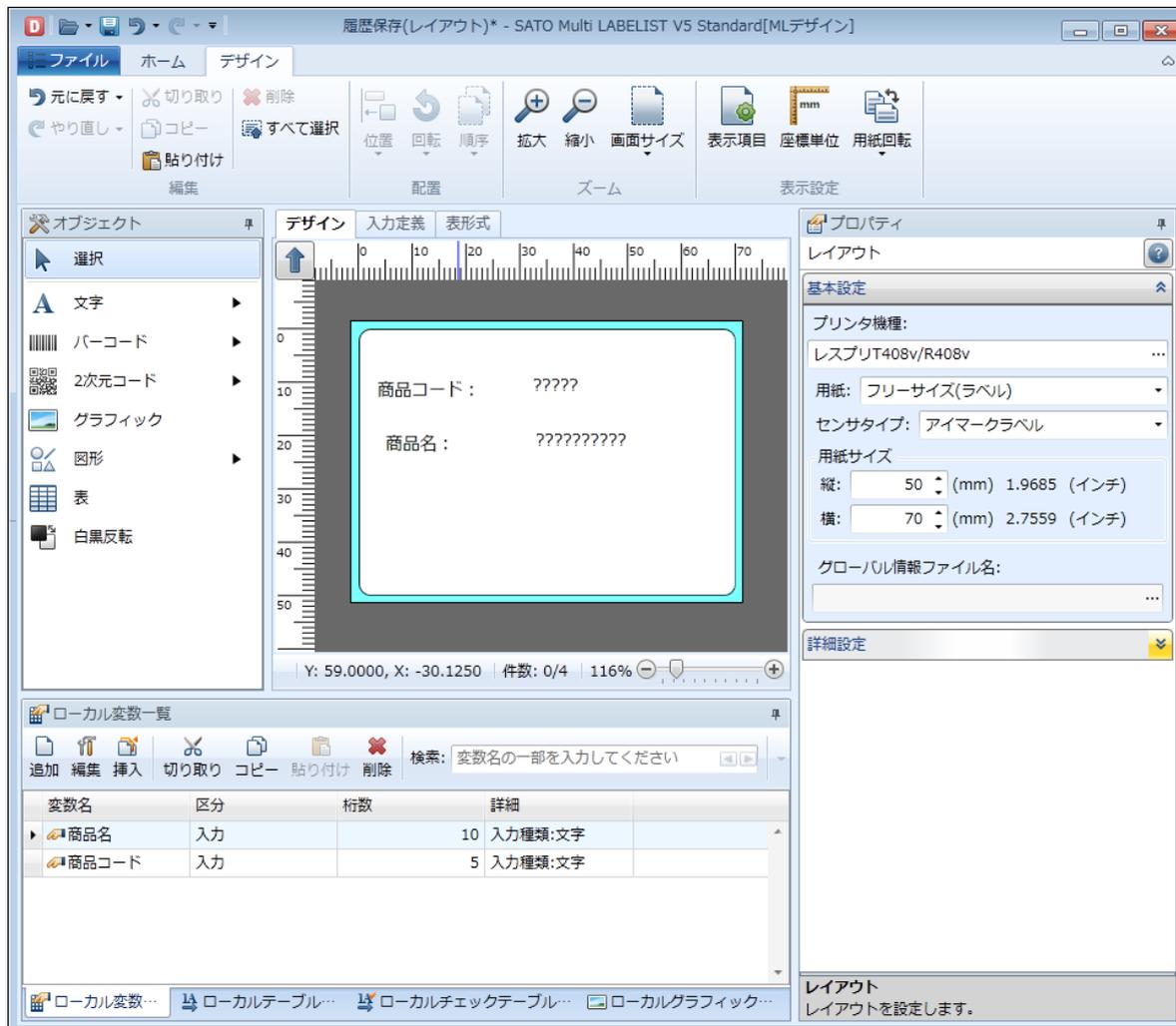


3. 履歴ファイルの作成

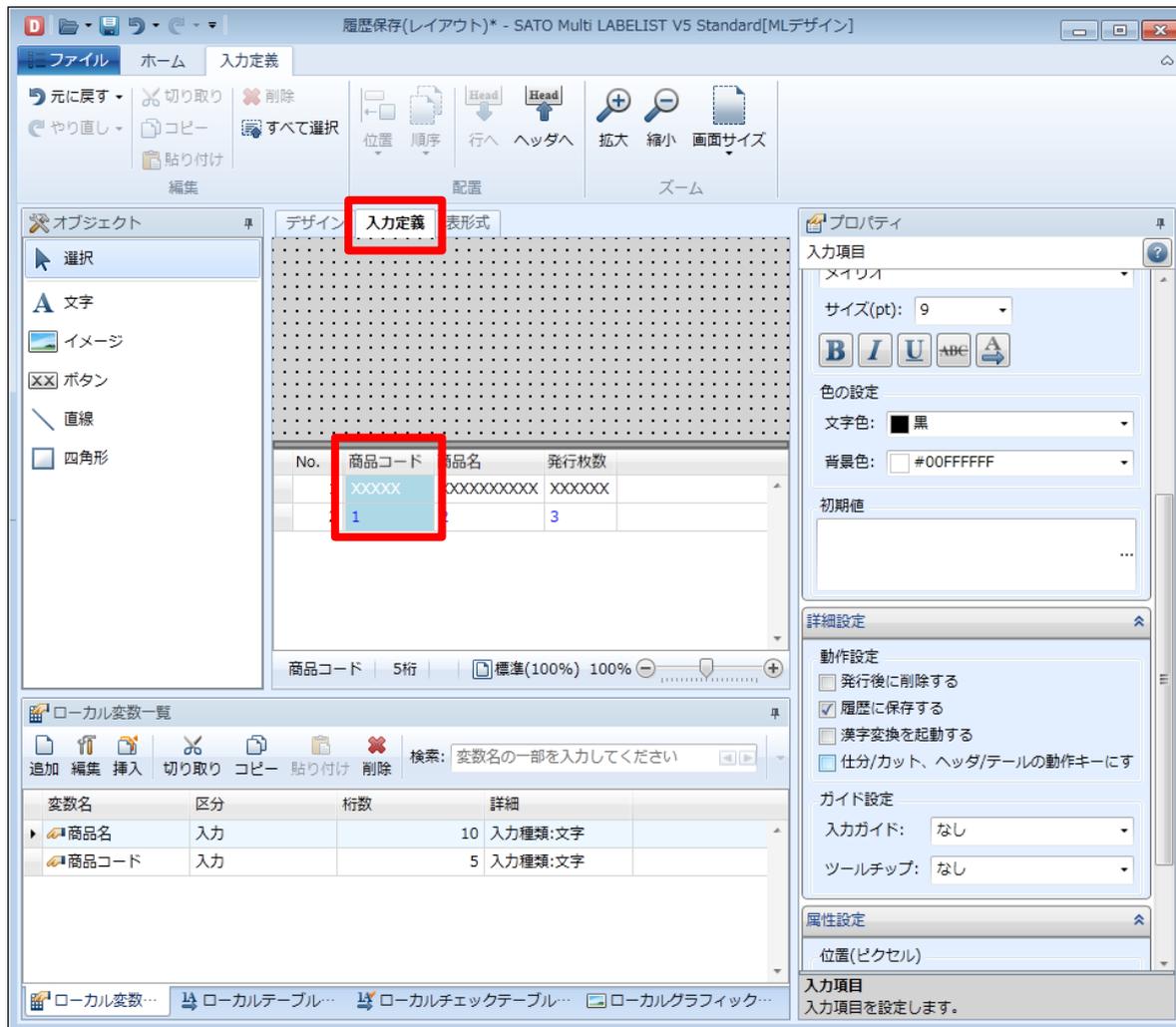
実際に下のレイアウトを使い、レイアウト発行による、商品コードを履歴保存対象にした履歴ファイルの作成を行います。

履歴の取得を行うレイアウトを「ML デザイン」で開きます。

(このレイアウトは入力項目として商品コード・商品名を使っています)

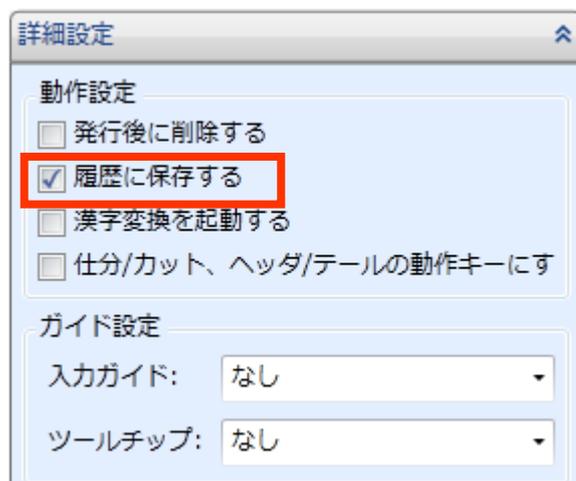


次に、入力定義画面に切替え、入力項目をクリックして、プロパティ画面を表示します。

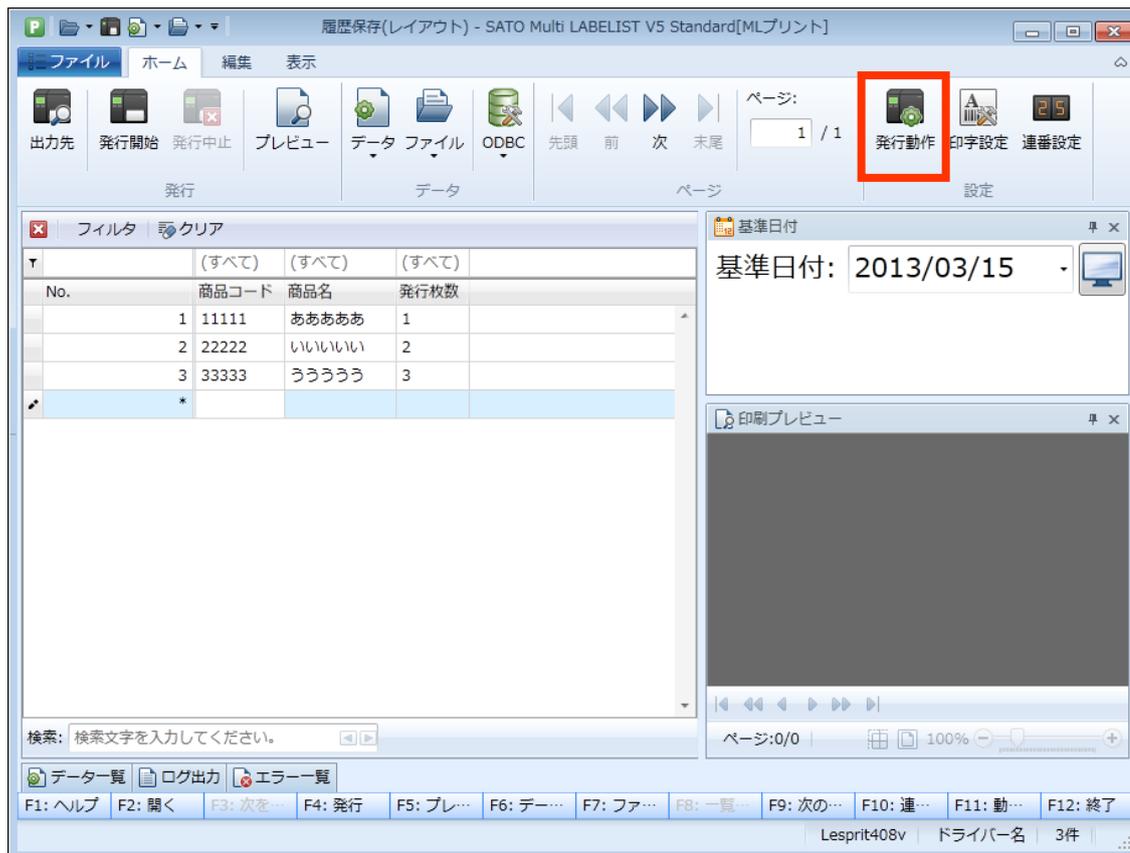


「詳細設定」項目内の「履歴に保存する」にチェックを入れ、レイアウトを保存します。
(デフォルトはチェックが入っています。)

履歴に保存する、しないの設定は入力項目毎に行う必要があります。

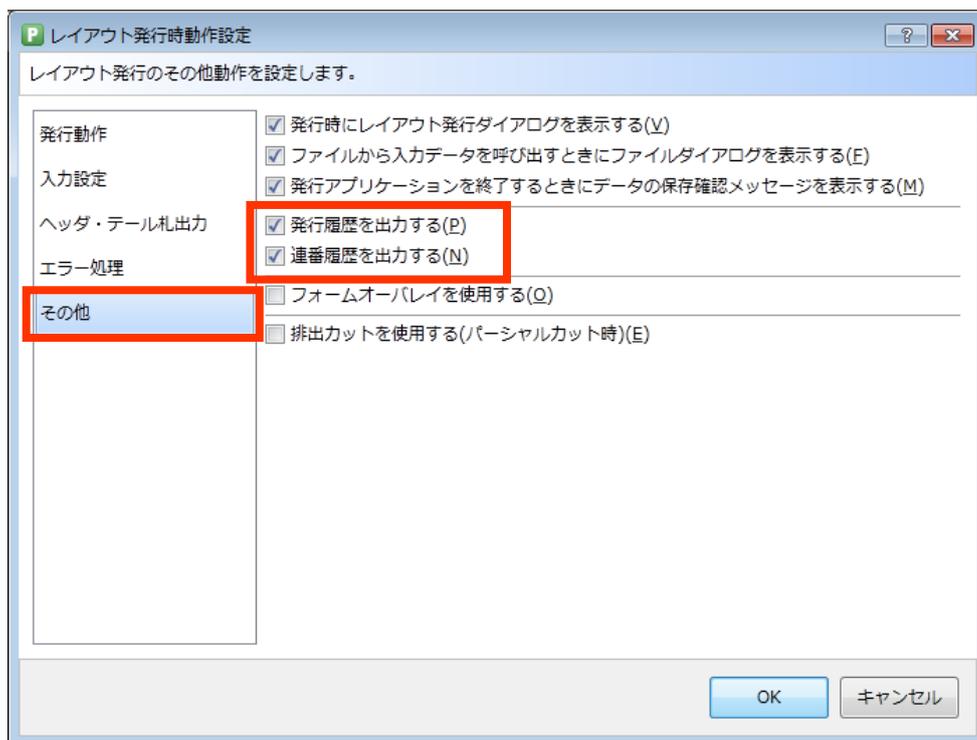


次に先ほどのレイアウトを「ML プリント」で開き、発行動作をクリックします。



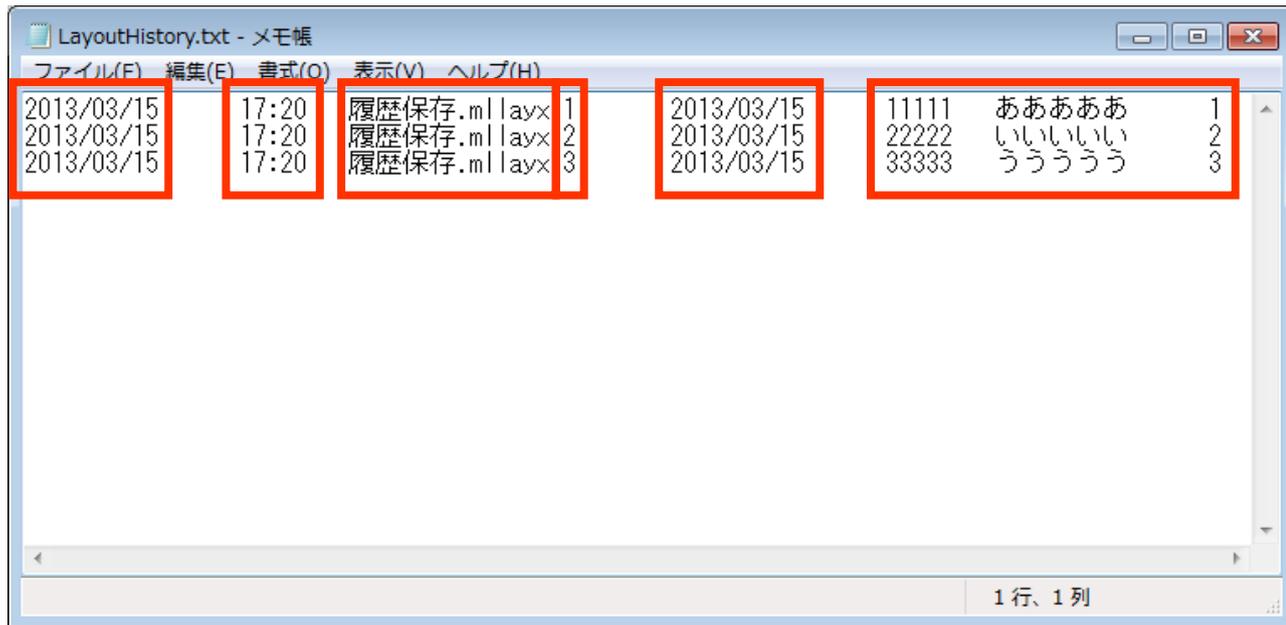
発行動作設定の画面でその他のタブを選択します。

「発行履歴を出力する」または「連番履歴を出力する」にチェックを入れ、「OK」ボタンをクリックします。（デフォルトはチェックが入っています。）



実際発行してみましょ。う。

発行後、ML メンテナンスで指定したパスに履歴ファイルが作成されているか確認してみましょ。う。



「発行した日付」「時間」「レイアウト (振分) 名称」「発行枚数」「標準日付」、そして最後に履歴保存対象として入力設計画面で指定した項目の値が書き込まれます。

連番履歴は、1 回の発行でヘッダ部とデータ部の 2 行のデータで構成されます。

分類	内容 (タブ区切り)				
ヘッダ部	ヘッダ区分 (H)	日時	レイアウト名称	変数区分 ローカル/グローバル	カウンタ変数名
データ部	データ区分 (D)	カウンタ値 1/カウンタ値 2/…/カウンタ値 n			

以上で「[3 : 履歴ファイル](#)」は終了です。

4：ML コンバート

1. ML コンバートとは

Multi LABELIST V4 で使用していたレイアウトおよび振分情報を Multi LABELIST V5 で使用できるように変換する機能です。オプションを有効にすることで、レイアウトに紐づく内部データも変換可能です。

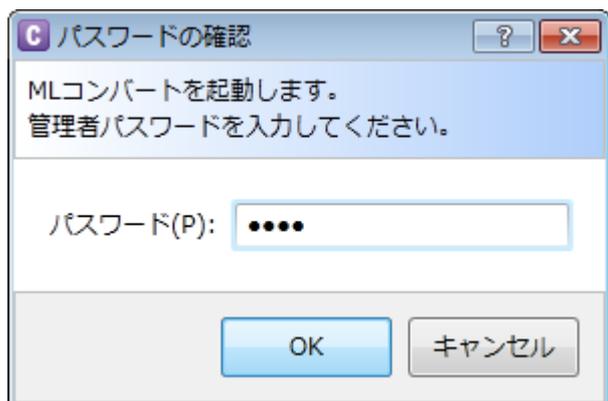
2. レイアウトの指定

それではデータコンバートの操作方法を説明致します。

Windows スタートメニュー>すべてのプログラム>Multi LABELIST V5 の

C MLコンバート を選択します。

パスワードを聞いてきますので、「user」と入力してください。画面上では「●」で表示されます。「OK」ボタンを選択すると ML コンバートの画面が表示されます。

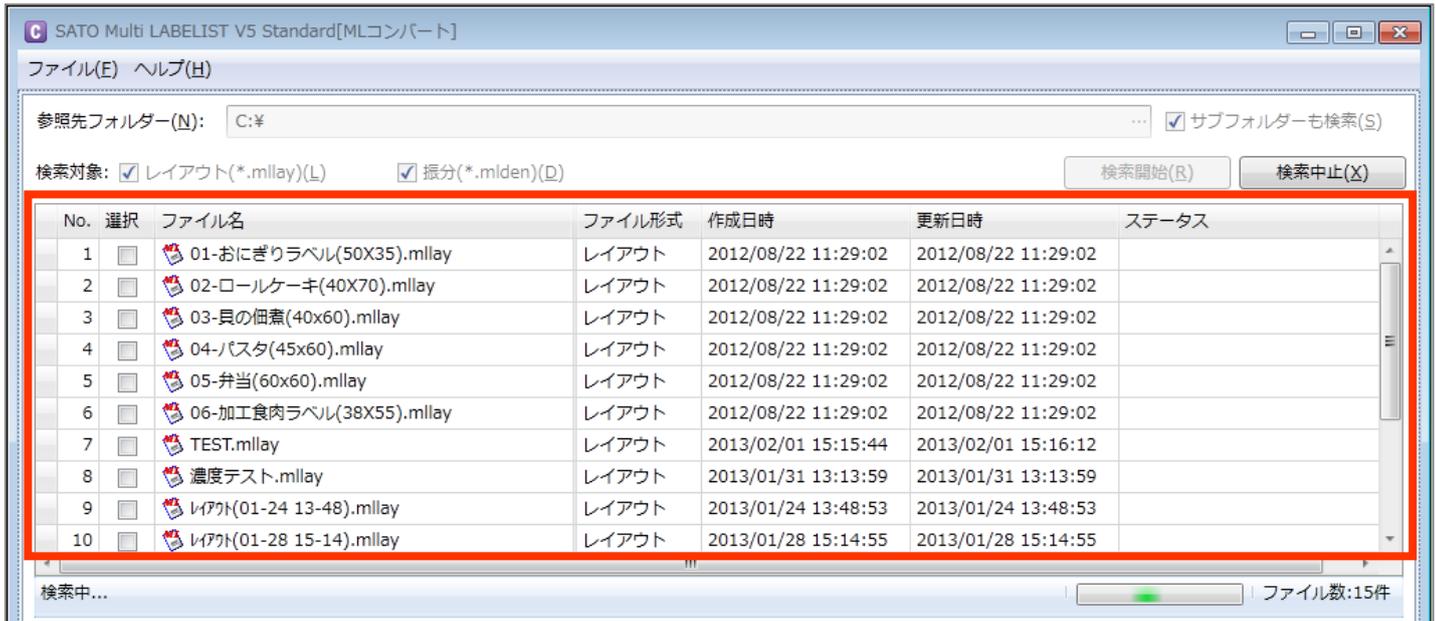


まず、Multi LABELIST V4 のレイアウトファイル、振分ファイルを検索します。

検索対象のフォルダ及び検索する対象ファイルの種類を指定し、「検索開始」ボタンをクリックします。



検索結果が表示されます。



3. オプション設定

オプション設定では、旧プリンタ機種に設定されているレイアウトを後継機種への割付けや、図形オブジェクト・Windows フォントの変更方法を変更することが可能です。

• 表示方法

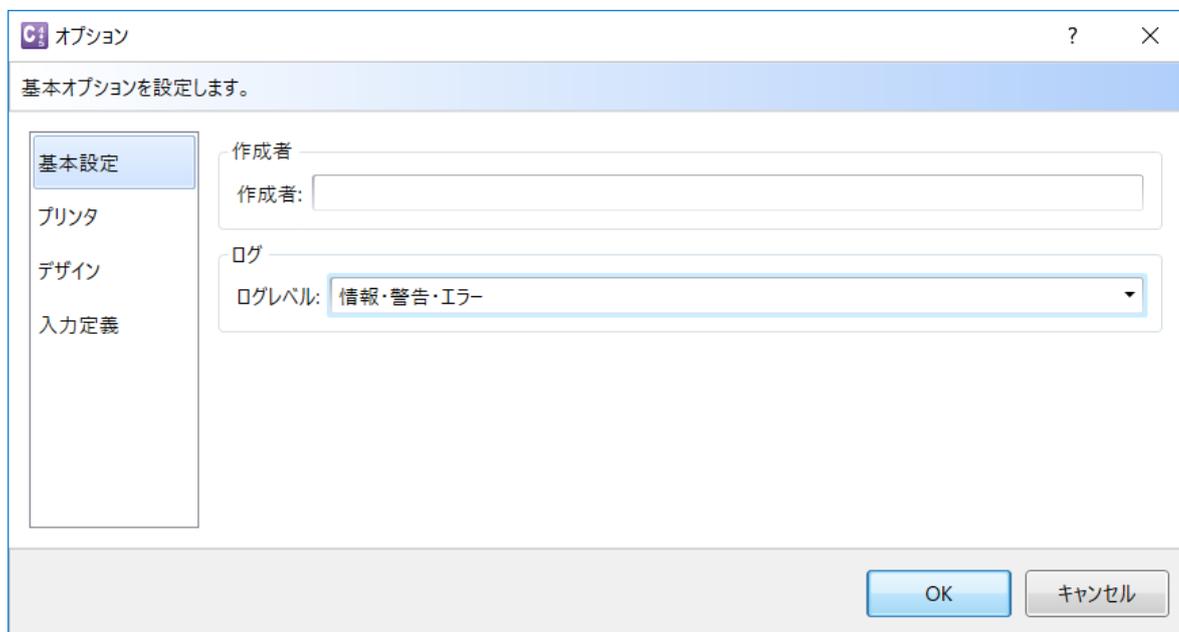
メニューの[ファイル]ー[オプション]を選択します



■基本設定

作成者は、ファイルの更新履歴に記録されます。コンバートを行った作成者を特定する場合に使用します。

ログは、コンバート時にログ出力ウィンドウに表示される内容をログレベルで変更します。「警告・エラー」にすることでレイアウトの確認が必要な情報のみに表示することができます。



■プリンタ割付

Multi LABELIST V4 の機種を V5 の機種に割付けます。

Multi LABELIST V4、V5 双方で対応しているプリンタ機種の場合、予め「新プリンタ機種」は指定されています。新プリンタ機種が「未選択」の機種が指定されているレイアウトをコンバートすると「プリンタ未選択エラー」となり、コンバートに失敗します。

新プリンタ機種一覧から、割付先のプリンタを選択します。

印字速度・濃度を変換する場合はチェックを有効にします。

プリンタの割付が終わったら、「OK」ボタンをクリックしオプション画面を終了します。



■デザイン設定

図形オブジェクト変換方法は、図形オブジェクトをグラフィックに変換することが可能です。

Windows フォント変換方法は、利用していた MLV4 のバージョンが Ver.4.4 以下か Ver.4.5 以上によってフォントサイズを調整します。

Windows フォント情報出力方法は、通常変更する必要はありません。Ver.5.9.0.1 より古いバージョンで変換したレイアウトと合せる場合は「WPF」に変更します。

グラフィックオブジェクトの伸縮モードは、通常変更する必要はありません。グラフィックが黒く潰れてしまう場合は「標準」に変更します。

■入力定義

入力項目の履歴保存は、入力項目に入力されたデータを発行履歴に出力する設定を有効にするか調整します。

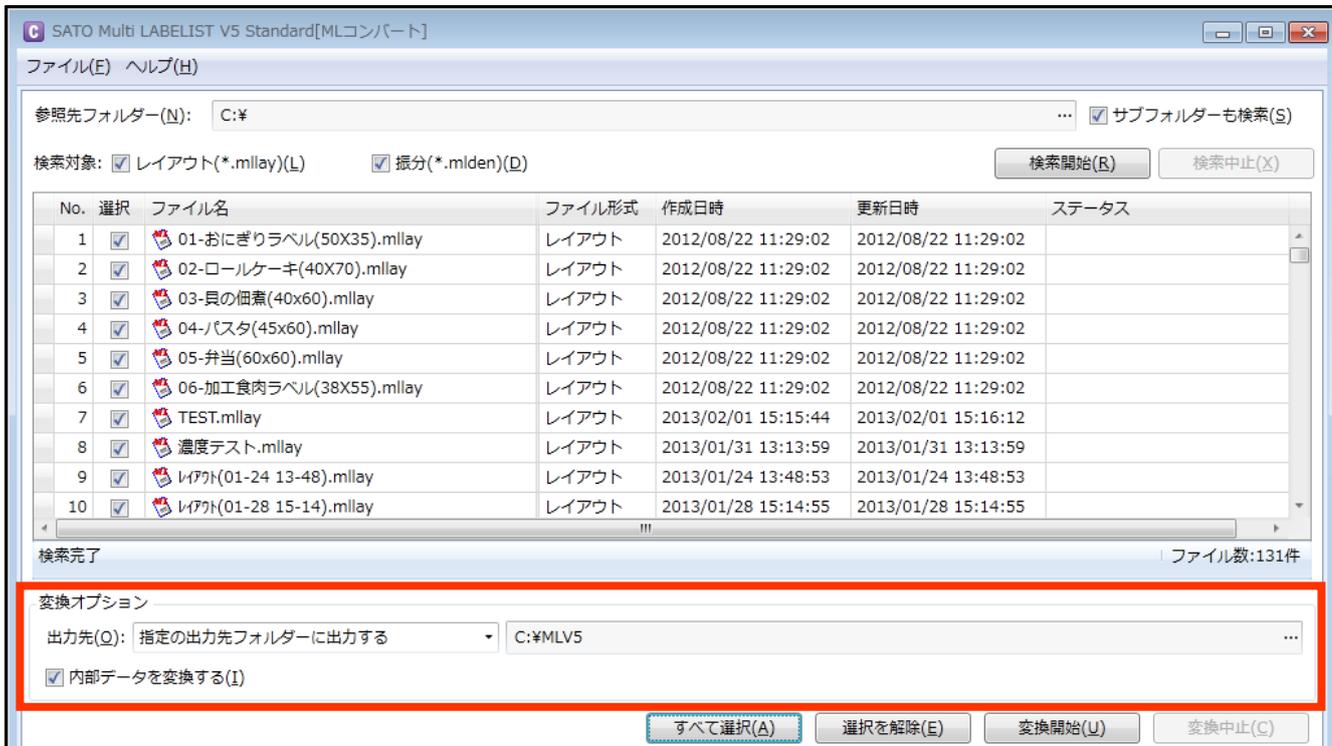
発行履歴の出力、連番履歴の出力は、履歴の出力有無を変更できます。

エラー処理は、発行時にエラーとしてメッセージ表示する動作を調整します。

4. コンバートを行う

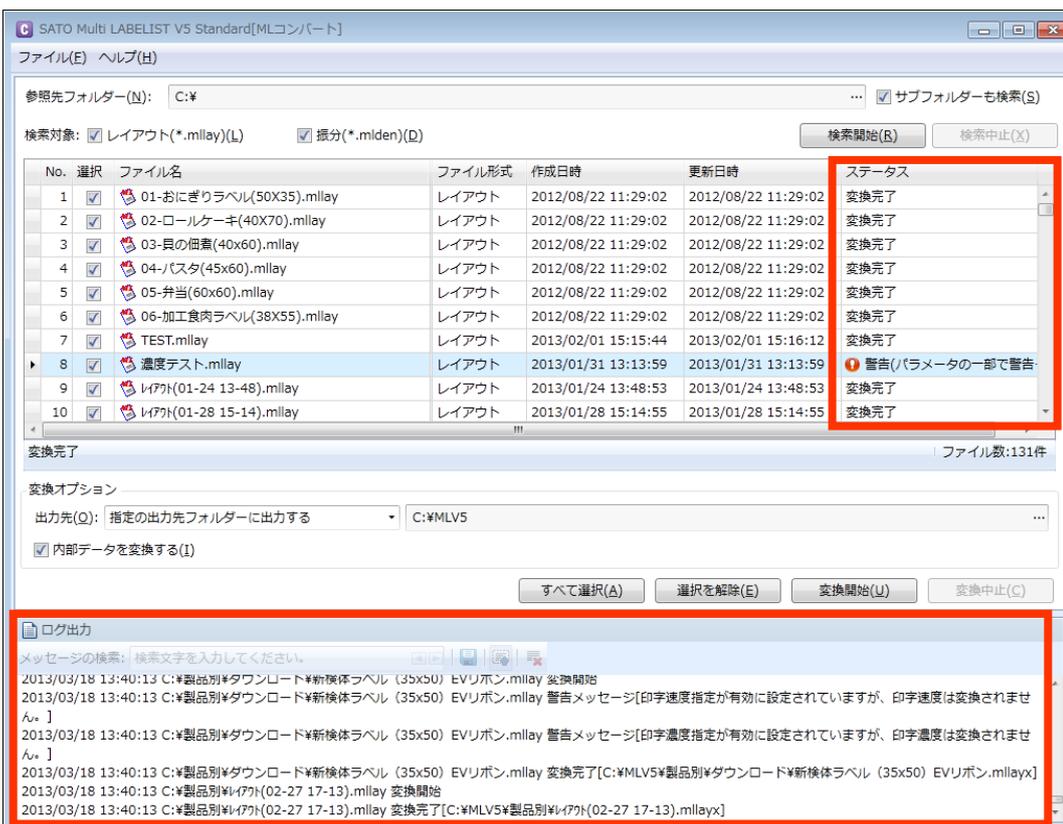
税率、プリンタ割付け設定が完了したら、変換オプションを設定します。

出力先、内部データを変換するかなどを指定し、検索ファイルから対象データを選択した上で変換を実行します。



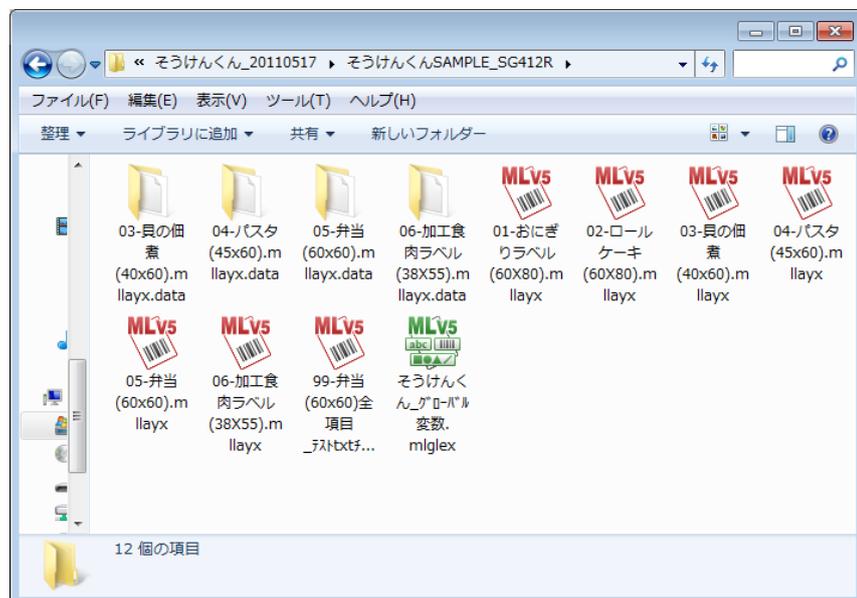
変換結果はステータスで確認できます。

「警告」はログ出力で詳細内容を確認してください。



コンバートが終了したら、出力先として指定した場所にフォルダが新たに作成されています。フォルダ構成は、元の Multi LABELIST V4 ファイルの保存先と同じ構成で作成され、その中に V5 用にコンバートされたレイアウトが作成されます。

レイアウト名に「.data」と付いたフォルダには内部データが保存されています。

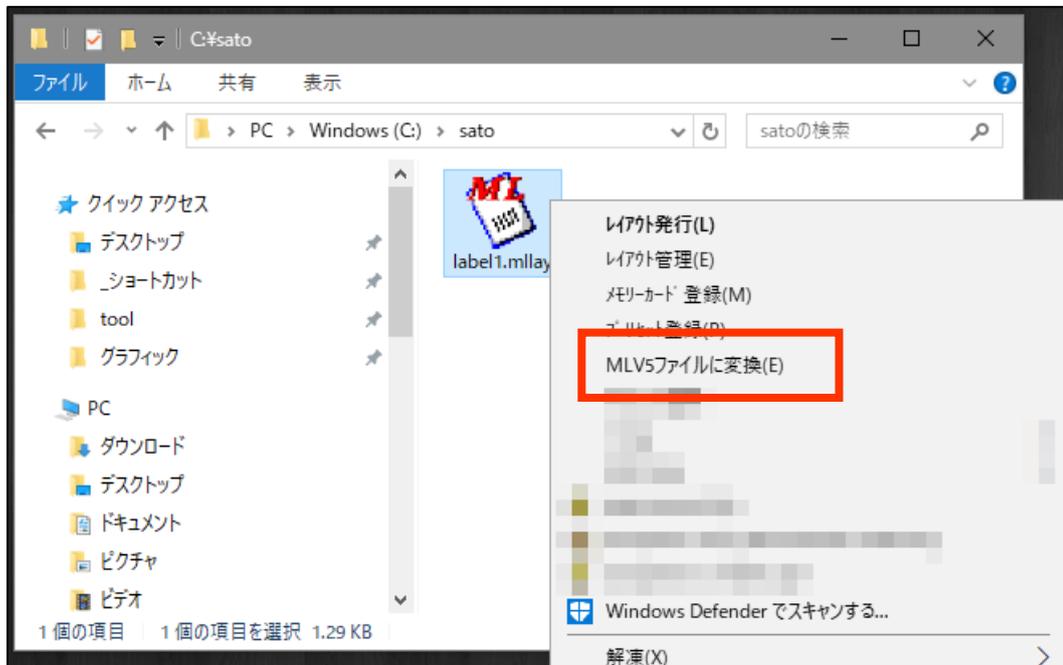


発行確認を行ってみましょう。

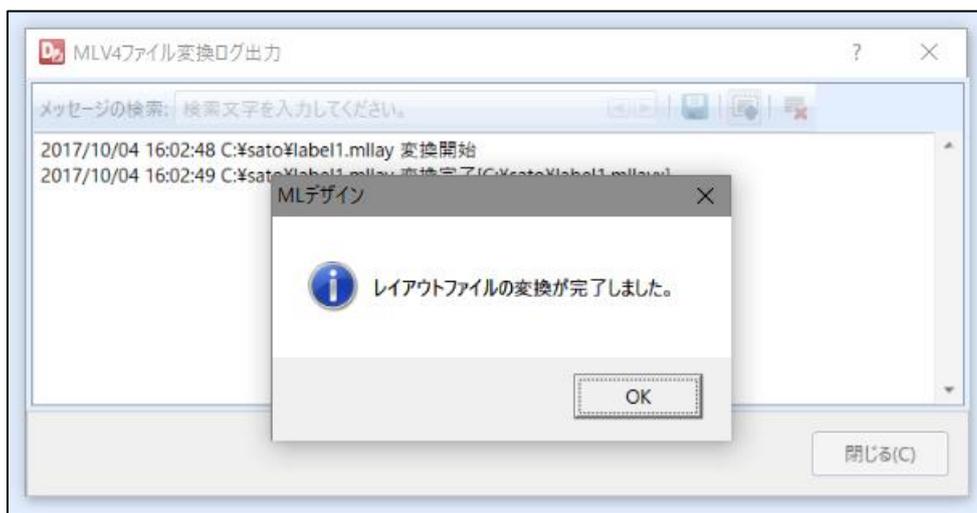
5. コンバートを行う（コンテキストメニュー）

税率、プリンタ割付け設定が完了している状態で、Multi LABELIST V4 のレイアウトファイルや振分ファイルのコンテキストメニュー（右クリックして表示されるメニュー）で、簡単にコンバートすることができます。（変換後のファイル出力先の指定や内部データの変換はできません。）

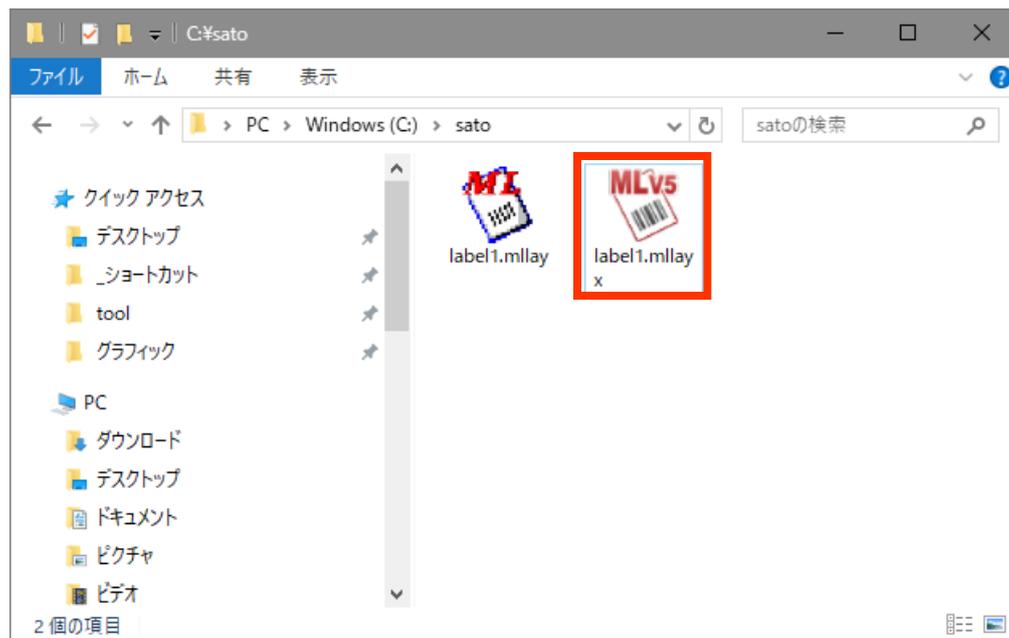
ファイルのコンテキストメニューで「MLV5ファイルに変換」をクリックします。



ML デザインが起動して変換が行われます。



変換後のファイルは、元のファイルと同じフォルダに同じ名称で保存されます。



以上で「[4 : ML コンバート](#)」は終了です。

5：連番を判別値毎に保存してみよう

1. 連番を判別値毎に保存するとは

入力データ（変数項目）の値により、連番値を判別して保存する機能です。
例としては、店舗名ごとに連番値を保存することが出来ます。

判別項目は最大3つまで登録でき、複数登録した場合はAND条件にて判別いたします。
キーと一致せずに判別できない場合は、設定外として共通の部分に連番値を保存します。
また、保存された連番値の修正は、MLプリント画面にて行えます。

2. 連番変数の設定

判別キーとして使用する入力変数を作成しておきます。

変数名：判別キー
桁数：10桁

次に、変数の追加にて「連番」を選択して各条件を入力して、「連番保存方法」で「判別値毎に保存する」を指定し、「判別値…」ボタンをクリックします。

D 変数設定

変数の設定を行います。

入力 連番種類(C): 数値(レイアウト) 発行時に入力する(I)

複写 数値(レイアウト) 編集パラメータ

結合 n進数: 10進数

連番 連番範囲: 1 ~ 99

日時 増減値: 1

計算 連番詳細

シンボル カウント条件: 指定した発行枚数毎にカウントする 指定枚数: 1 枚

連番保存方法: 判別値毎に保存する 判別値...

連番保存値の初期化: なし

変数名: 判別キー-1 桁数: 10

OK キャンセル

項目1に先ほど作成しておいた変数「判別キー」を指定し、値の種類を「文字」にします。判別値の入力が完了したら「OK」ボタンをクリックします。

判別値は「ファイルから取込み」ボタンでファイルのデータを利用することも可能です。

判別値設定を行います。最初に判別項目を選択して、選択した判別項目の判別値を入力します。

判別項目

項目1: 判別キー ▼ 判別条件1: 文字 ▼

項目2: ▼ 判別条件2: 数値 ▼

項目3: ▼ 判別条件3: 数値 ▼

判別値設定

No.	判別値1	判別値2	判別値3
1	渋谷		
2	恵比寿		
3	目黒		
4	田町		
*			

追加(A)

削除(D)

上へ移動(U)

下へ移動(W)

ファイルから取込み(I)...

ファイルへ出力(E)...

OK

キャンセル

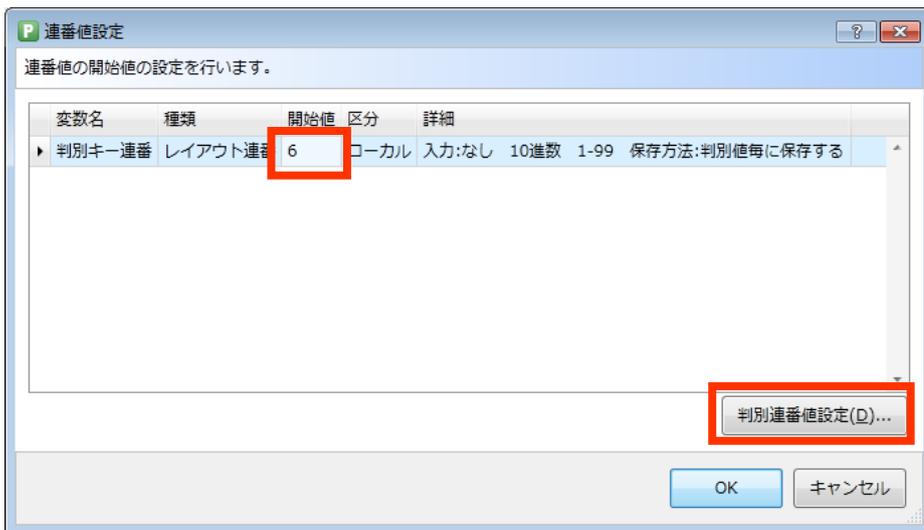
3. 連番保存値修正手順

ML プリント画面にて「連番設定」をクリックします。

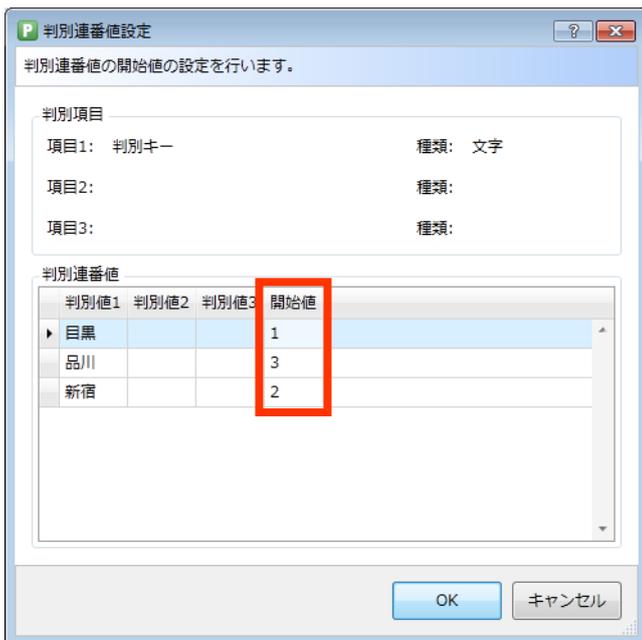


判別キーに一致しなかったデータの連番保存値が表示されます。

判別キー毎の連番保存値を修正する場合は「判別連番値設定」をクリックします。



設定した判別キーの値が表示され、それぞれの連番開始値を修正可能です。

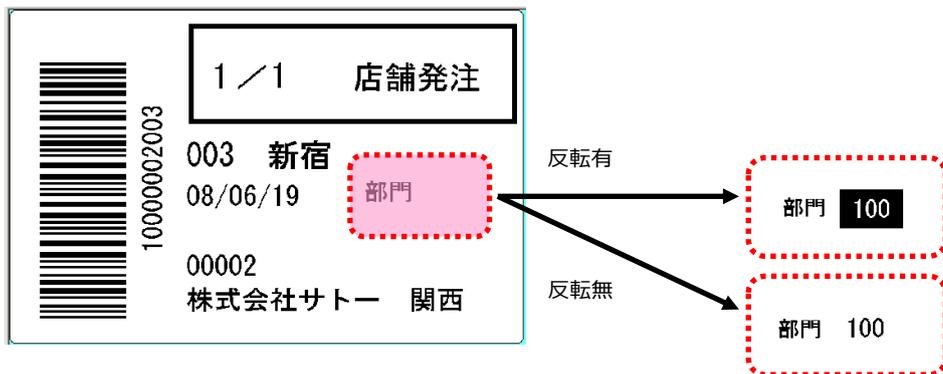


以上で「[5：連番を判別値毎に保存する](#)」は終了です。

6：オブジェクトの印字条件設定

1. オブジェクトの印字条件設定とは

白黒反転や罫線など、ラベルデザインを構成する各オブジェクトの印字有無を制御する事が出来る機能です。例えば、ラベルデザイン上の一部が入力項目の値によって白黒反転する・しないの双方のケースが存在する場合、条件を設定することで白黒反転の切り替えが可能となります。



今回は、既存のレイアウトファイルを使い、下記条件に合致した場合に部門コード（変数文字）を白黒反転させるパターンを例に説明いたします。

条件「①部門コード ②が ③1～99 → 白黒反転する（それ以外は白黒反転しない）」

- ①「変数」に登録
- ②「入力チェックテーブル」に印字有りの条件値を設定
- ③白黒反転オブジェクトに印字有無の条件（①、②）を設定

2. オブジェクトの印字条件設定に必要な事前設定

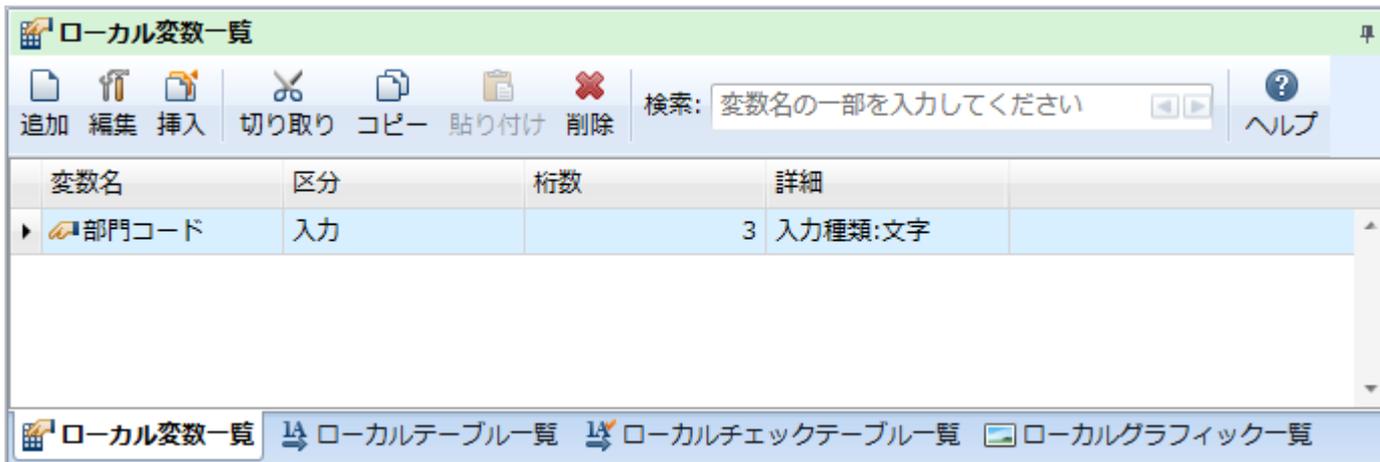
印字する・しないの切替え設定を行うためには、ML デザインの画面で、条件となる「変数」および「チェックテーブル」の作成が必要になります。

本マニュアルでは「変数」と「チェックテーブル」を条件に使用します。

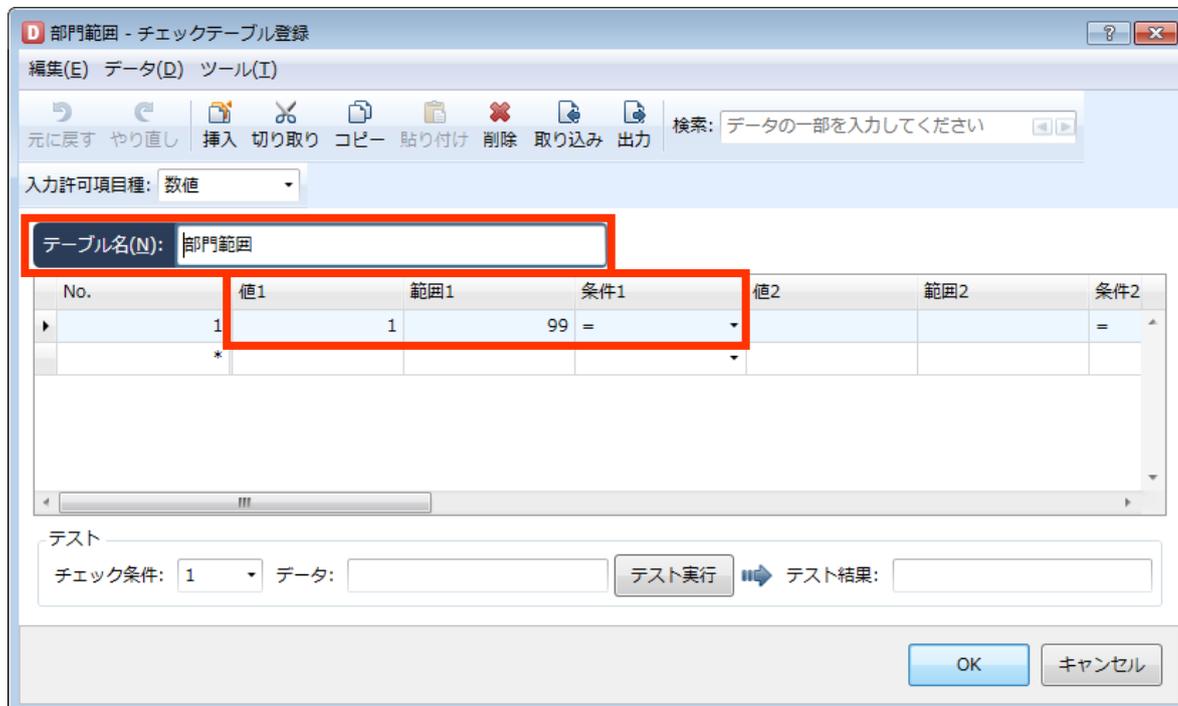
Windows スタートメニュー>すべてのプログラム>Multi LABELIST V5 の

D MLデザイン を選択します。

条件に使用する変数を以下のように設定します。

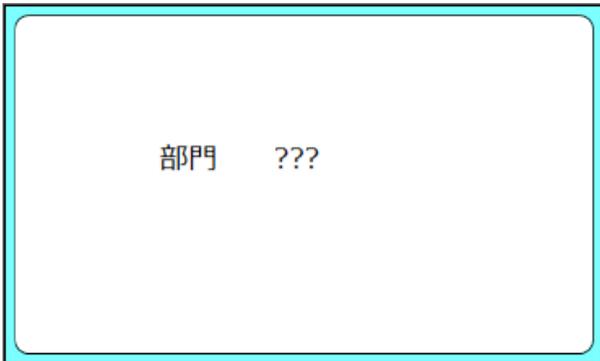


条件に使用するチェックテーブルを以下のように設定します。

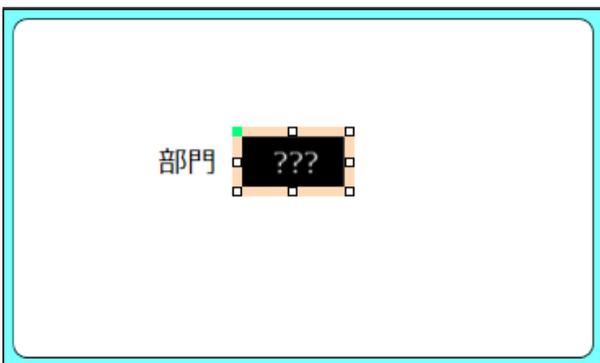


3. オブジェクト印字有無の条件設定

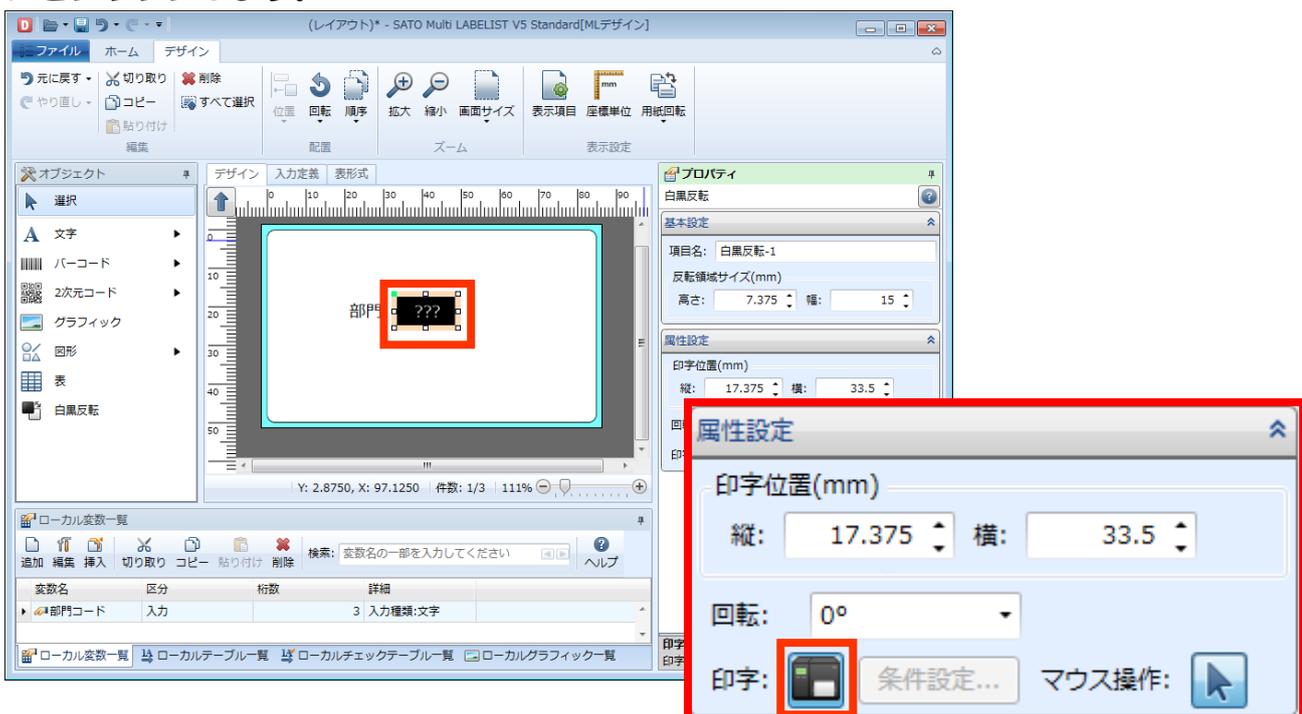
「ML デザイン」画面で、「2」で作成した部門コード変数と、固定の文字列を以下のようにレイアウト上に配置します。



印字有無の条件を設定するオブジェクトをレイアウトに配置します。
(本マニュアルでは、「白黒反転オブジェクト」を部門コード上に配置します)



「2」の白黒反転オブジェクトをクリックし、プロパティウィンドウの属性設定の「印字」アイコンをクリックします。

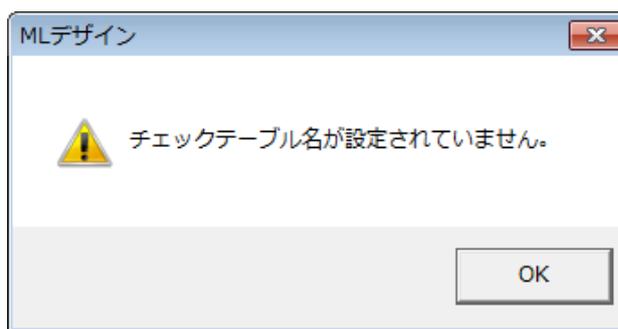
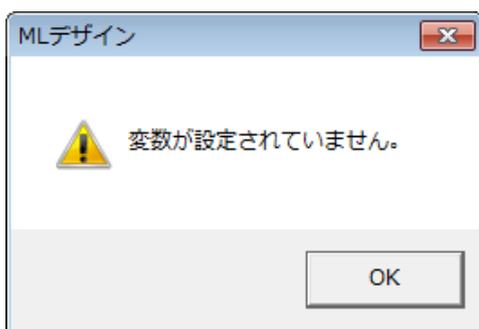


印字条件設定画面で、変数名にあらかじめ作成した「部門コード」を指定し、判定内容で「指定ローカルチェックテーブルの条件を満たす」を選択します。

ローカルチェックテーブルの指定画面で、作成しておいた「部門範囲」を選択し、使用するチェックテーブル条件 No(1~5)を選択します。

最後に「OK」ボタンをクリックすると登録します。

「変数」もしくは「条件」が設定されていない状態で「OK」ボタンをクリックすると、以下のようにエラーメッセージが表示されます。設定した上で、再度「OK」ボタンをクリックしてください。



以上で設定作業は完了です。レイアウトは名前を付けて保存し、次の項目で、発行やプレビュー画面を使い確認してみましょう。

4. 印字確認

先ほど作成したレイアウトファイルを ML プリントで起動します。
設定した条件どおりに印字されるか、実際に発行して確認を行って下さい。

今回は条件を「1～99」のチェックテーブルにしているため、
部門コードに「88」を入力して印字した結果は、白黒反転されます。



部門コードに「100」を入力して印字した結果は、白黒反転されません。



印字有無の判断は変数編集後の値で行います。

そのため、条件設定の参照変数には、編集後の結果が分かり難い変数の使用は極力さけ、入力文字変数（各種編集は一切設定しない）にて設計いただくことをお勧めいたします。

例として、条件設定で参照する変数に、テーブル変換編集が有効だった場合、変換後の値とチェックテーブルの値が比較されます。

入力値	テーブル変換後の値
001	目黒
002	品川
003	新宿

→ 条件の判別値として利用されます。

以上で、「[6. オブジェクトの印字条件設定](#)」は終了です。

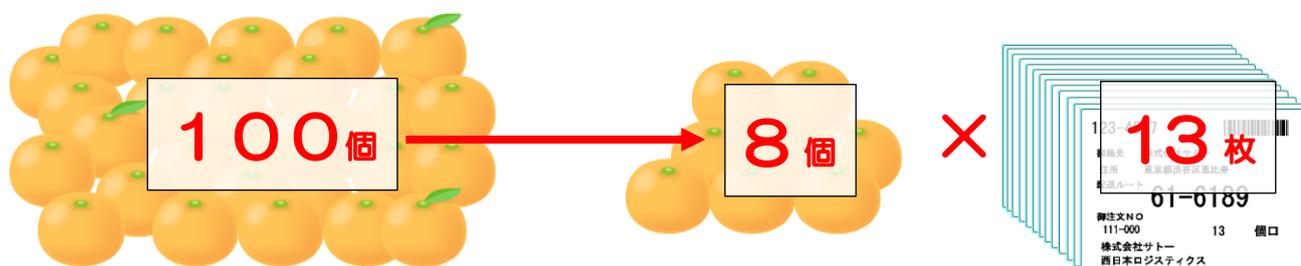
7：発行枚数計算機能

1. 発行枚数計算機能とは

変数の値や固定値などを計算項目として任意の計算式を設定し、発行枚数を算出できる機能です。例えば、100個ある商品を決まった数ずつ箱詰めした際の箱に貼るラベルの数などが算出できます。

100個ある商品を8個ずつ箱詰めし、箱の数分ラベルを発行したい場合、
 $100 \div 8 = 12$ 余り 4 → 13箱分のラベルが必要となります。

発行枚数計算機能を活用することで、枚数を手入力することなく必要な枚数で発行できます。



2. 変数の作成

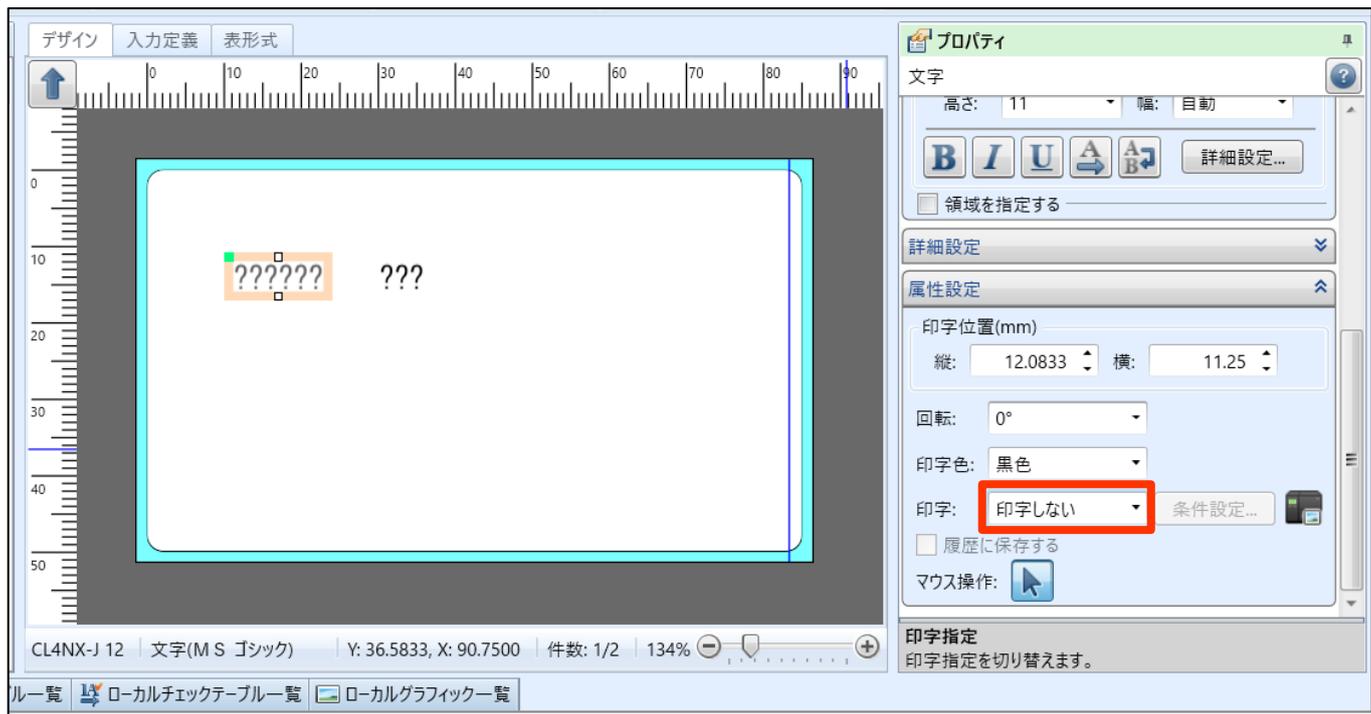
発行枚数演算の計算対象項目に「入力値」を使用するためには、予め「ML デザイン」画面で「変数」を作成し、「変数文字」としてレイアウト上に配置しておく必要があります。本マニュアルでは以下の「変数」を発行枚数計算の条件として使用します。

Windows スタートメニュー>すべてのプログラム>Multi LABELIST V5 の **MLデザイン** を選択します。

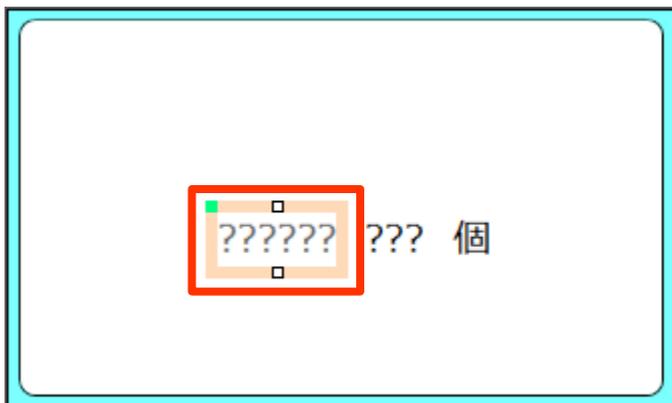
条件に使用する変数を以下のように設定します。

変数名	区分	桁数	詳細
個数	入力		6 入力種類:文字
入数	入力		3 入力種類:文字

発行枚数計算に使用する変数は、発行時に値を入力し計算するため、その値をラベルやタグに印字する必要が無い場合でも、「変数文字」としてレイアウト上に配置する必要があります。配置した上で印字不要の場合は、文字設定画面で「印字しない」を指定してください。



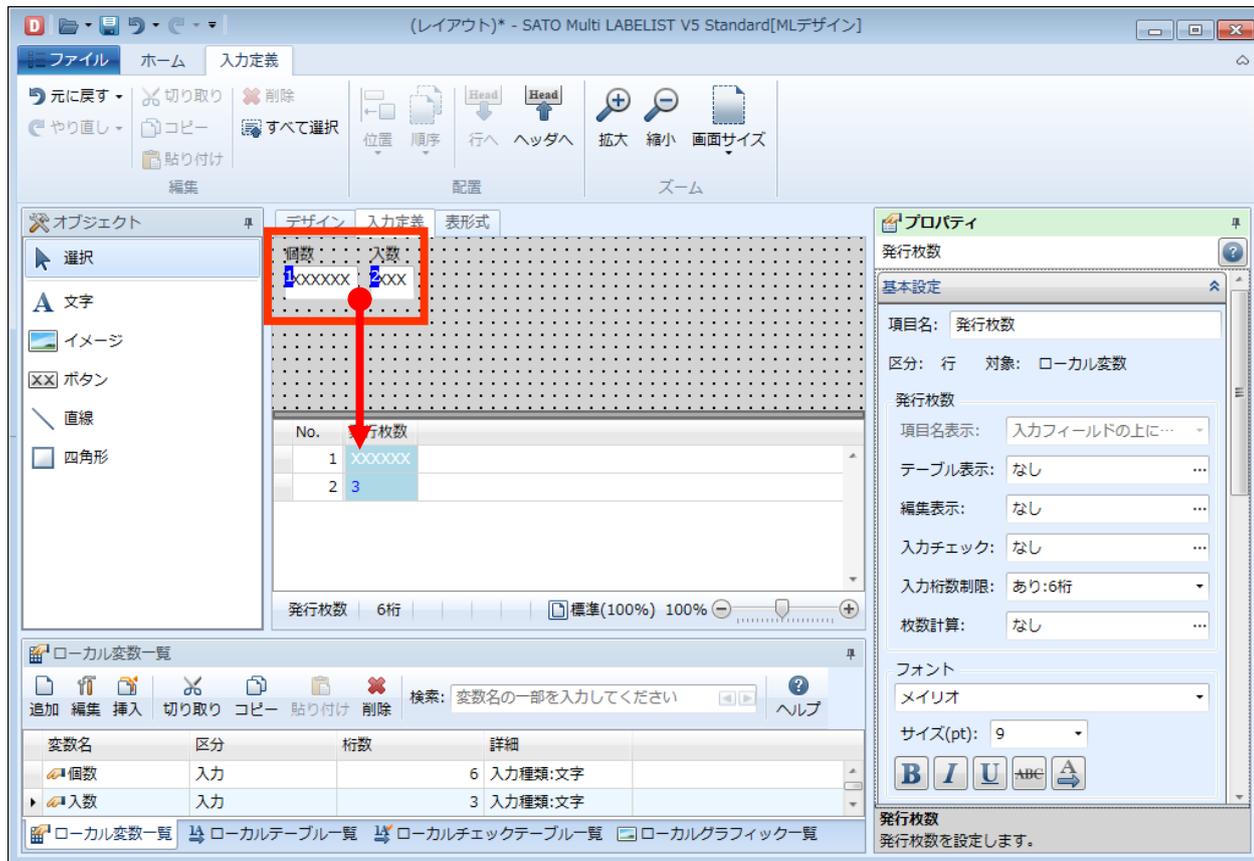
本マニュアルのレイアウトでは、作成した「入数」変数のみ印字し、「個数」はラベルやタグに印字不要のため、「印字しない」設定の変数文字とします。印字しない設定をすると文字がグレーの表示になります。



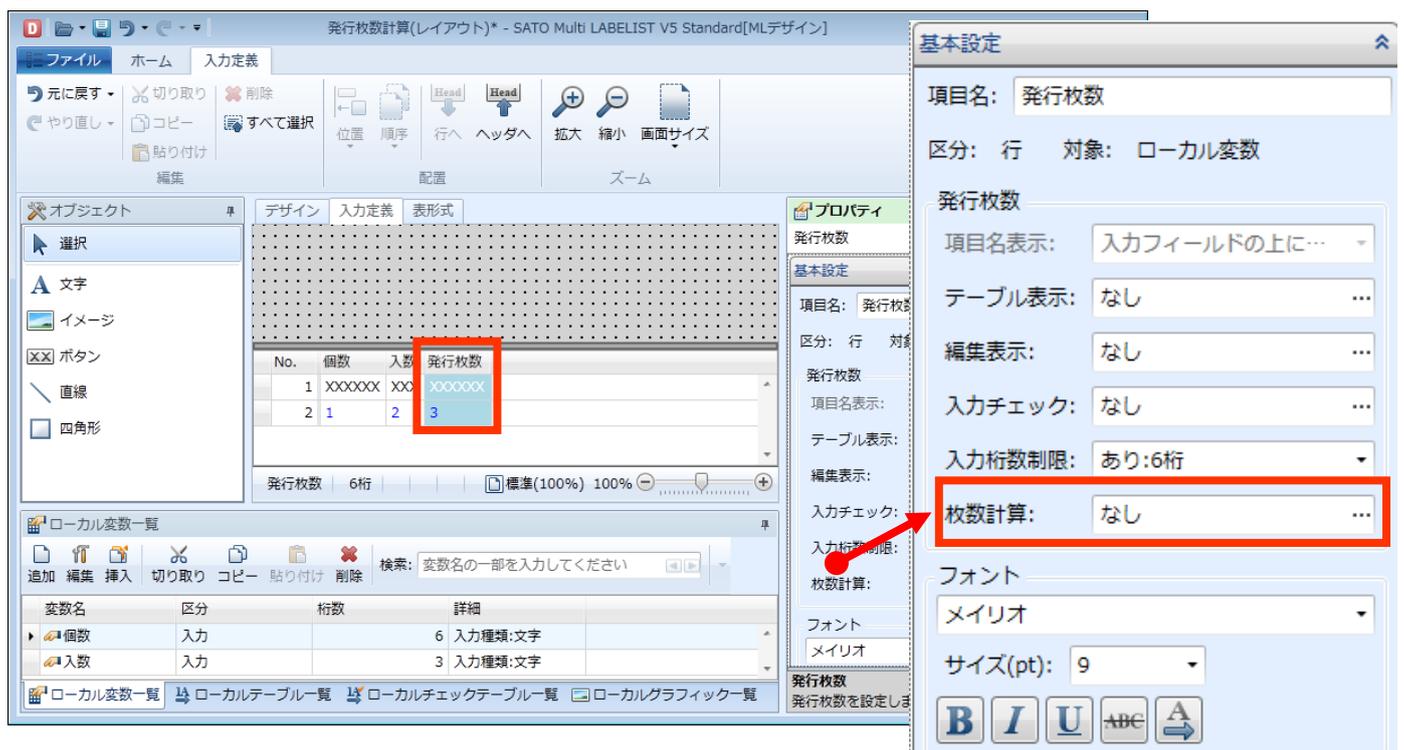
3. 発行枚数計算の条件設定

「2」で作成したレイアウトの「入力定義」画面を表示します。

ヘッダー部にある「個数」「入数」項目を行部に移し、項目の順番を入替えます。



「発行枚数」項目の上でクリックし、プロパティウィンドウの「基本設定」の中の「枚数計算」項目をクリックします。



「枚数計算」画面が開きますので、「発行枚数の計算を行う」チェックを有効にし、以下の項目を入力可能にします。

枚数計算

最初に発行枚数計算の有無を選択します。発行枚数の計算を行う場合は、計算式の設定をしてください。

発行枚数の計算を行う(C)

計算式(E): テスト(I)...

入力項目(I)

項目の挿入

開始位置: 終了位置:

関数(N)

関数の挿入

書式:

説明:

計算種類(K)

+ - × ÷ ¥

括弧(P)

()

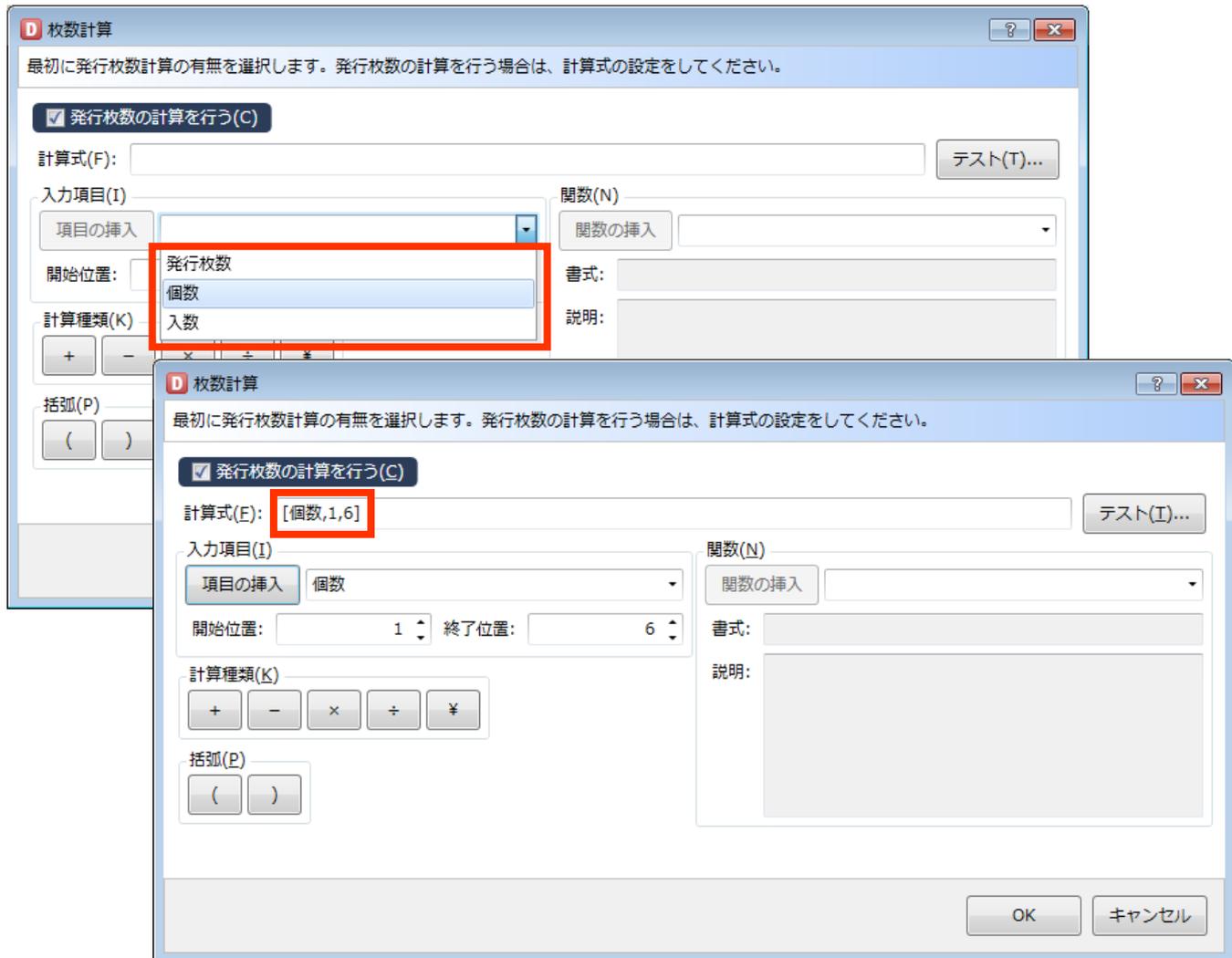
OK キャンセル

「計算式」項目に式を設定します。画面にある入力項目等から計算式を作成することが可能です。本マニュアルでは、「個数」÷「入数」の結果を切り上げた値を取得したいため、「ROUNDUP 関数」を使って計算式を記述します。

「ROUNDUP 関数」では、「引数」として、切り上げ処理をする値を指定します。今回切り上げ処理をしたいのは「個数」÷「入数」の計算結果の値なので、「引数」にはこの「個数」÷「入数」という計算式をセットする必要があります。

計算式を「引数」として手でセットすると、記述にミスなどが発生しやすいため、一度「計算式」の項目に「入力項目」で変数を選択して「個数」÷「入数」の計算式をセットし、その計算式をコピーすると、ミスも少なく設定が可能です。

ではまず、入力項目で変数「個数」を選択し「項目に挿入」で計算式に値をセットします。



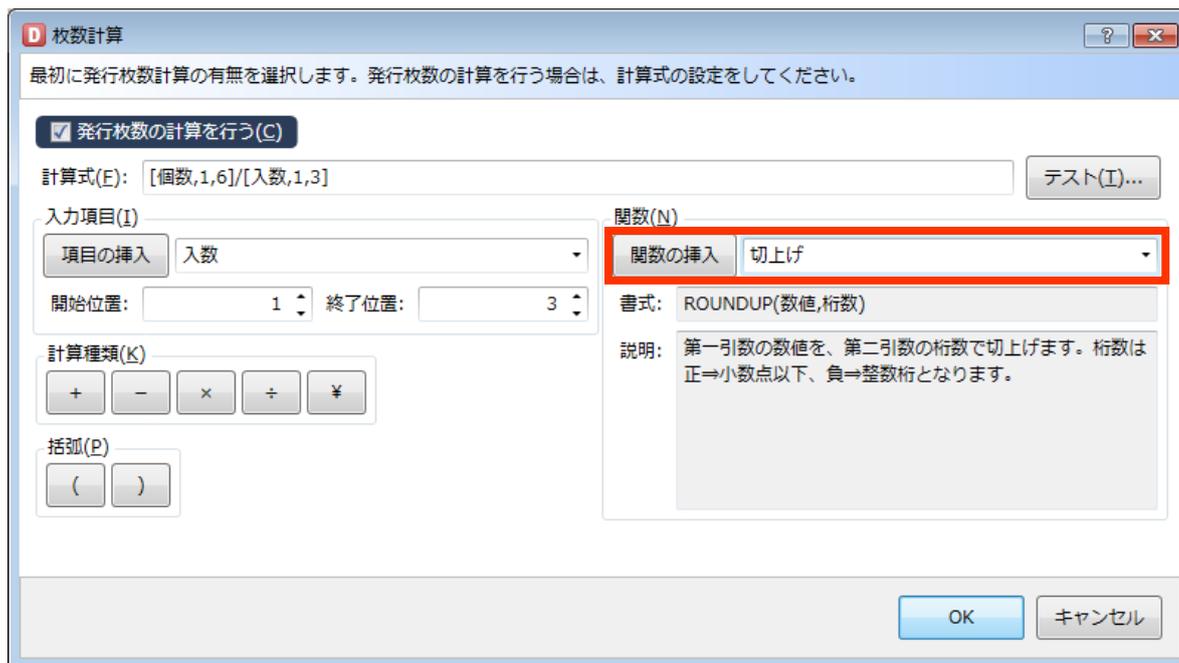
次に、「計算種類」の「÷」をクリックします。

計算式(E): [個数,1,6]/

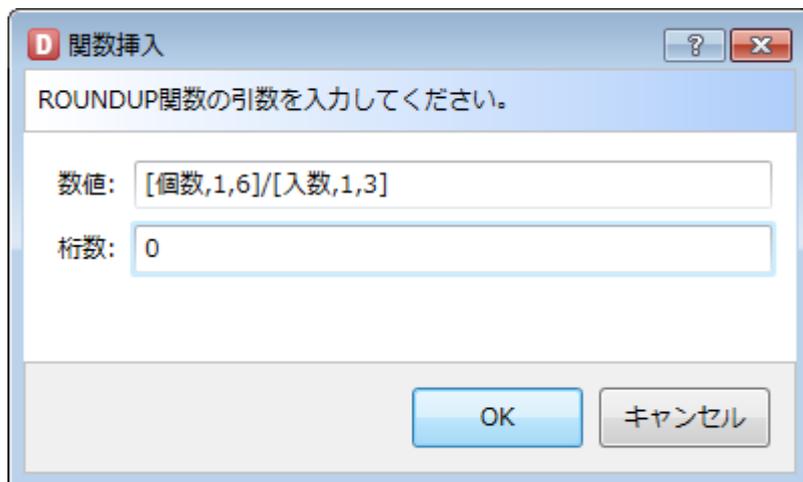
更に、変数「入数」を選択し、以下のような計算式を作成します。

計算式(E): [個数,1,6]/[入数,1,3]

これで、「ROUNDUP 関数」の引数に入れる計算式ができましたので、この計算式を切り取り、「関数」の一覧から「切上げ」を選択の上、「関数の挿入」ボタンをクリックします。



関数挿入画面が表示されますので、「数値」に先ほど切り取りした計算式をセットします。「桁数」は「0」とセットし、「OK」ボタンをクリックします。



計算式に、以下のような計算式がセットされましたか？
「テスト」ボタンをクリックし、計算結果を確認してみましょう。

枚数計算

最初に発行枚数計算の有無を選択します。発行枚数の計算を行う場合は、計算式の設定をしてください。

発行枚数の計算を行う(C)

計算式(E): ROUNDUP([個数,1,6]/[入数,1,3],0) テスト(I)...

入力項目(I)
項目の挿入 入数

開始位置: 1 終了位置: 3

計算種類(K)
+ - × ÷ ¥

括弧(P)
()

関数(N)
関数の挿入 切上げ

書式: ROUNDUP(数値,桁数)

説明: 第一引数の数値を、第二引数の桁数で切上げます。桁数は正⇒小数点以下、負⇒整数桁となります。

OK キャンセル

「個数」「引数」にそれぞれ「100」、「8」と入力し、「テスト実行」ボタンをクリックします。
計算結果に「13」と表示されましたか？確認が終わったら「閉じる」をクリックします。

計算式テスト

計算式のテストを行います。計算式で使用している入力項目の値を入力して、テスト実行を行うと計算結果を表示します。

計算式: ROUNDUP([個数,1,6]/[入数,1,3],0)

No.	入力項目	データ
1	個数	100
2	入数	8

計算結果: 13

テスト実行(I) 閉じる

枚数計算画面に戻ったら、「OK」をクリックし登録を完了します。

D 枚数計算

最初に発行枚数計算の有無を選択します。発行枚数の計算を行う場合は、計算式の設定をしてください。

発行枚数の計算を行う(C)

計算式(E): ROUNDUP([個数,1,6]/[入数,1,3],0) テスト(I)...

入力項目(I)

項目の挿入 入数

開始位置: 1 終了位置: 3

計算種類(K)

+ - × ÷ ¥

括弧(P)

()

関数(N)

関数の挿入 切上げ

書式: ROUNDUP(数値,桁数)

説明: 第一引数の数値を、第二引数の桁数で切上げます。桁数は正⇒小数点以下、負⇒整数桁となります。

OK キャンセル

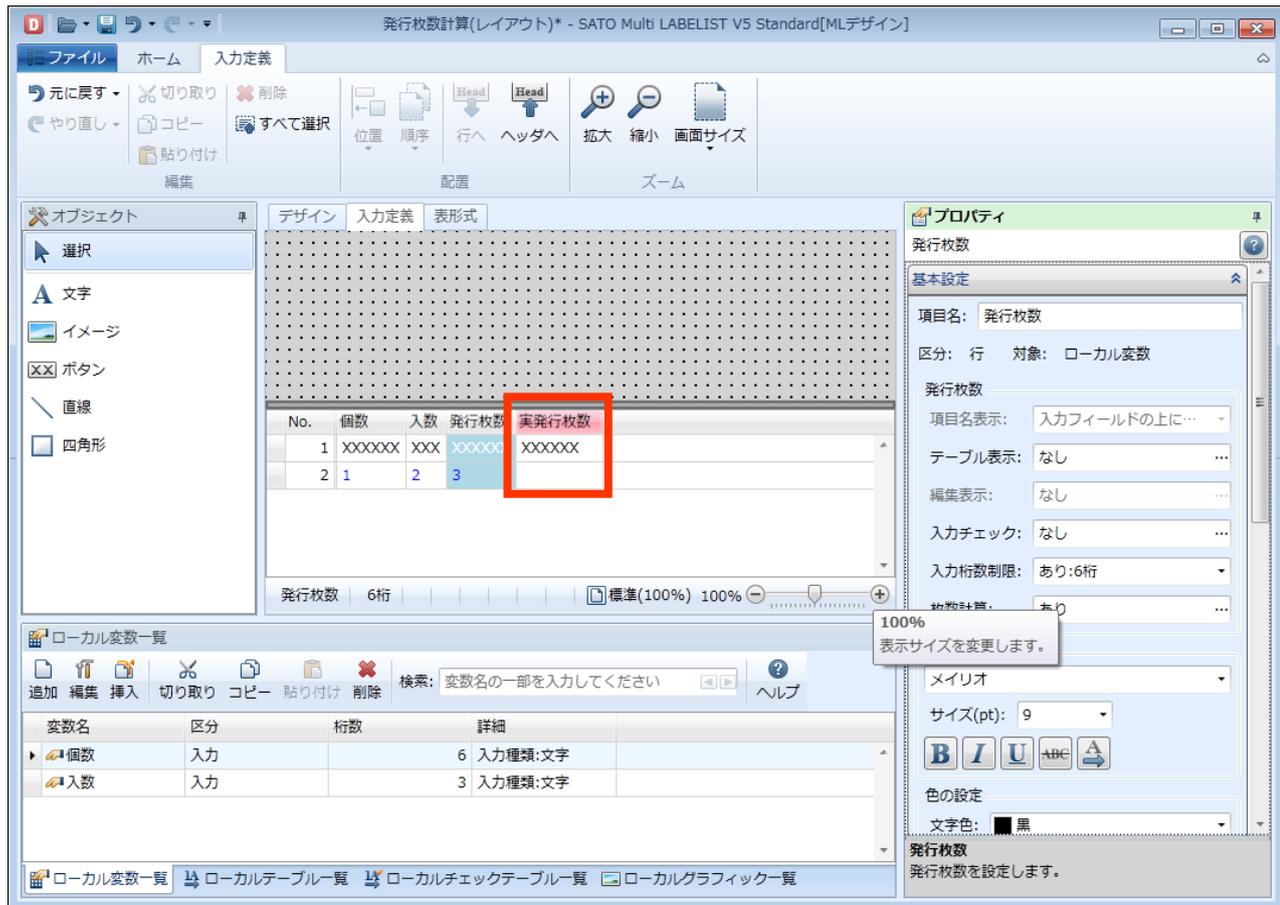
本マニュアルのように、「発行枚数」項目を計算に使用しない場合、以下のメッセージが表示されますので、「はい」を選択してください。

MLデザイン

発行枚数の入力が必要な設定です。
発行枚数を入力不可に変更しますか?

はい(Y) いいえ(N)

設定が完了すると、入力項目の「発行枚数」の右隣に「実発行枚数」項目が表示されます



以上で設定作業は完了です。

次の項目で、発行画面で値を入力し、実発行枚数（計算結果）を確認しましょう。

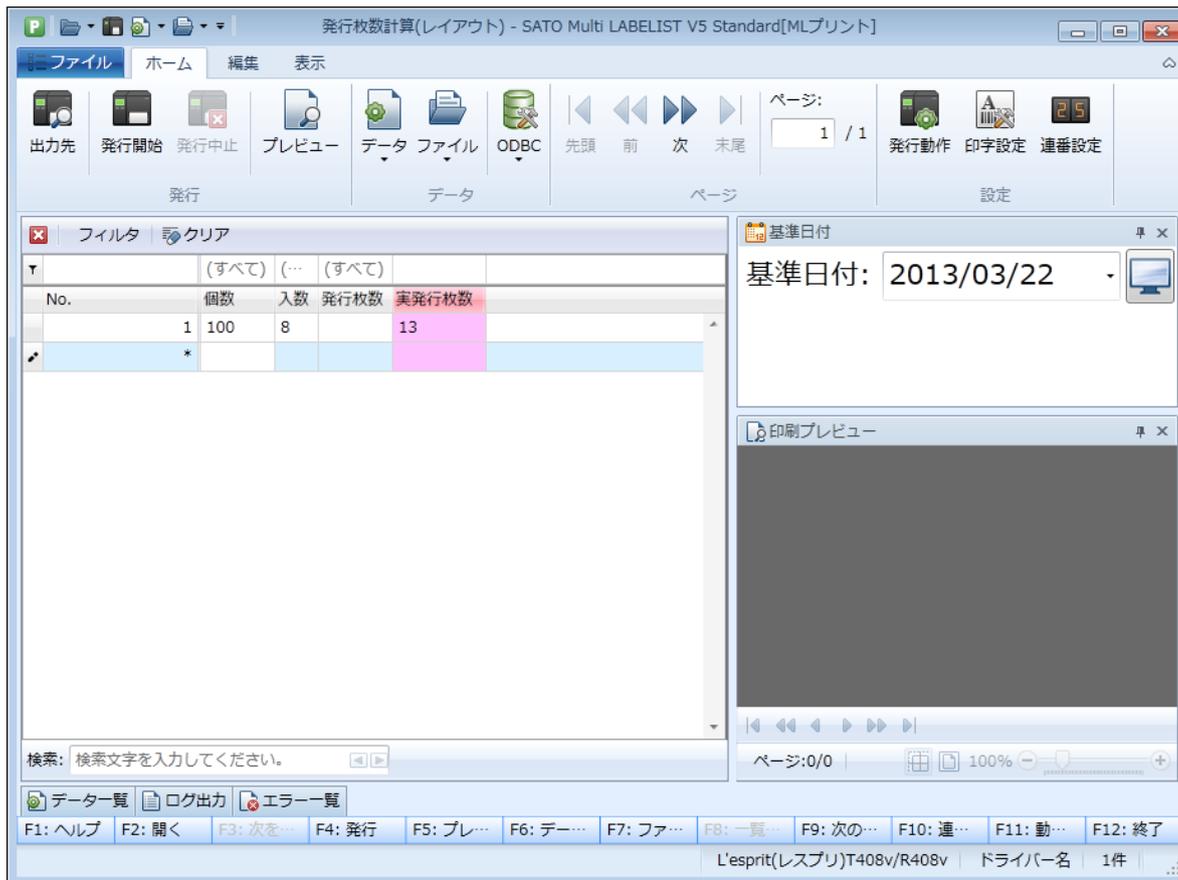
4. 発行枚数計算結果確認

レイアウト発行画面にて入力項目に値を入力し、「実発行枚数」項目に正しい演算結果の値がセットされるかを確認します。

本マニュアルでは以下の条件で計算していますので、計算結果は13枚となります。

<条件>

100個ある商品を8個ずつ箱詰めし、箱の数分ラベルを発行したい場合



以上で、「[7：発行枚数計算機能](#)」は終了です。

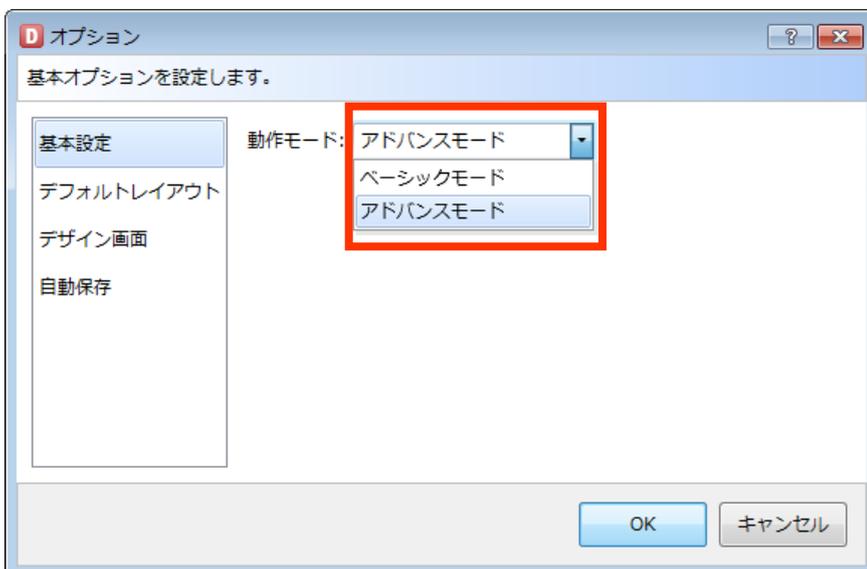
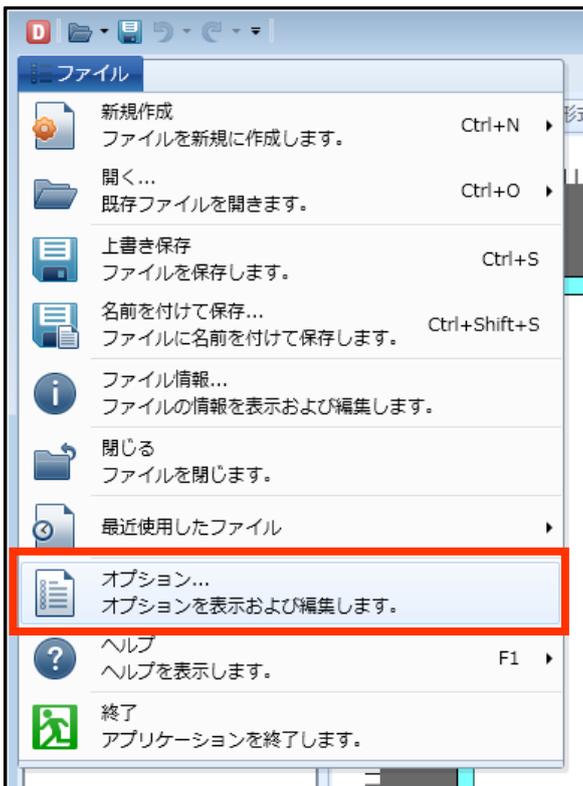
8：アドバンスモードを使ってみよう

1. アドバンスモードとは

白黒反転や合成シンボルなど拡張機能を利用することができる動作モードです。

2. アドバンスモードの設定方法

ML デザインのファイルメニューから「オプション」を選択し、動作モードを「アドバンスモード」に設定します。



合成シンボルやコマンドの機能が利用できるようになります。



3. アドバンスモードの機能

ベーシックモードとアドバンスモードは利用できるデザインオブジェクトと変数に違いがあります。

■デザインオブジェクト

デザインオブジェクト		サトープリンタ		カラープリンタ	
		ベーシック	アドバンス	ベーシック	アドバンス
選択		○	○	○	○
文字	(1)貼付	○	○	○	○
	(2)入力	○	○	○	○
	(3)連番	○	○	○	○
	(4)日時	○	○	○	○
バーコード	(1)貼付	○	○	○	○
	(2)入力	○	○	○	○
	(3)GS1 Databar	×	○	×	○
	(4)GS1-128	×	○	×	○
	(5)CODE128	×	○	×	○
2次元 コード	(1)貼付	○	○	○	○
	(2)入力	○	○	○	○
合成 シンボル	(1)貼付	×	○	×	○
	(2)入力	×	○	×	○
グラフィック	(1)貼付	○	○	○	○
	(2)呼出	×	○	×	○
	(3)入力	×	○	×	○
図形	(1)直線	○	○	○	○
	(2)四角形	○	○	○	○
	(3)三角形	○	○	○	○
	(4)円	○	○	○	○
表		○	○	○	○
白黒反転		○	○	×	×
赤色領域		×	○	×	×
コマンド		×	○	×	×
RFID		×	○	×	×
シンボルウ ィザード		×	○	×	○

■変数

変数	設定項目	アドバンスモードで使用可能になる項目
入力	変数種類	[16進文字コード]、[グラフィック]
	編集パラメータ	次表を参照。
固定	すべて	(ベーシックモードは非表示)
複写	編集パラメータ	次表を参照。
結合	編集	クロスチェック
連番	連番種類	[数値(プリンタ)]、[文字]
	数値(レイアウト)	[ループ時に増減値を引き継ぐ]
	数値(レイアウト) -カウント条件	[発行指示毎にカウントする]、[デザインオブジェクト毎に カウントする]、[他の変数値がループする毎にカウントす る]
	編集パラメータ	次表を参照。
日時	日時種類	[システム日付]
	文字種別	[全角]
	編集パラメータ	次表を参照。
計算	変数種類	[プライスC/D]、[プライス丸め]
	編集パラメータ	次表を参照。
条件	すべて	(ベーシックモードは非表示)
シンボル	すべて	(ベーシックモードは非表示)
システム	すべて	(ベーシックモードは非表示)

・編集パラメータ

アドバンスモードは全ての機能が利用できます。

編集パラメータ	ベーシックモードの制限				
	入力	複写 (編集前)	連番	日時	計算
テーブル変換	○	○	×	○	○
改行削除	×	×	×	×	×
指定文字削除	×	×	×	×	×
税編集	○	○	×	×	○
カンマ編集	○	○	○	×	○
通貨編集	○	○	×	×	○
1文字補填	×	×	×	×	×
桁寄せ	○	○	○	○	○
置換	×	×	×	×	×
前ゼロ補填	○	○	○	×	○

以上で「[8：アドバンスモードを使ってみよう](#)」は終了です。

9 : GS1 DataBar 及び GS1 合成シンボル印字機能

1. GS1 DataBar 及び GS1 合成シンボル印字機能とは

GS1 DataBar 及び GS1 合成シンボルを印字するための機能です。

GS1 DataBar 及び GS1 合成シンボルを利用する場合は、[動作モードをアドバンスモードに変更してください。](#)

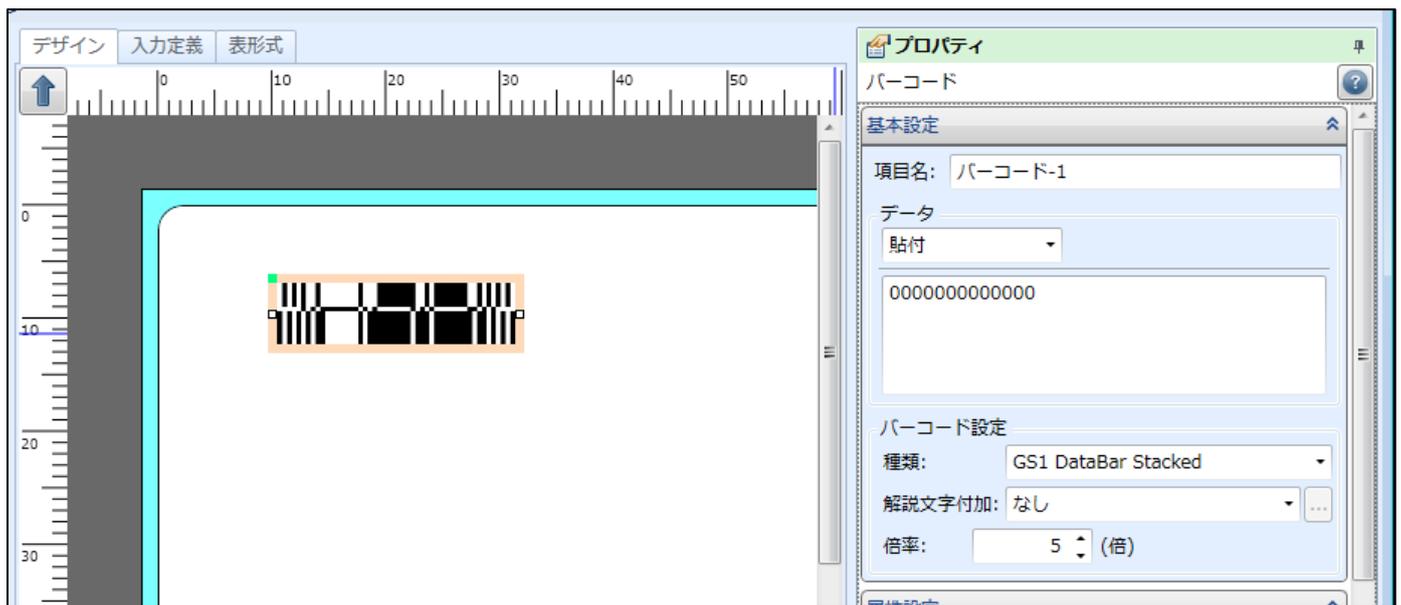
2. レイアウトの設計方法

■GS1 DataBar 指定

1. ML デザインでバーコードオブジェクトのプロパティにて、GS1 DataBar として作成可能なバーコード種類が「GS1 DataBar」という名称で一覧に表示されます。

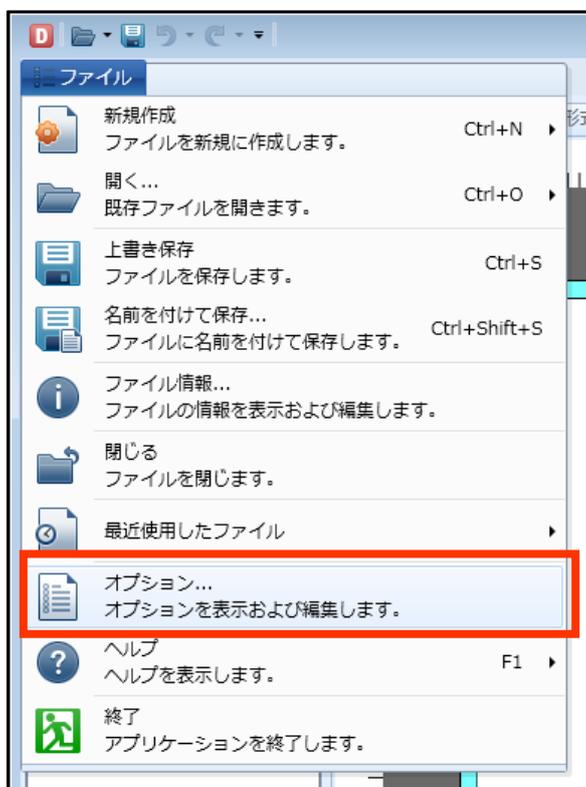


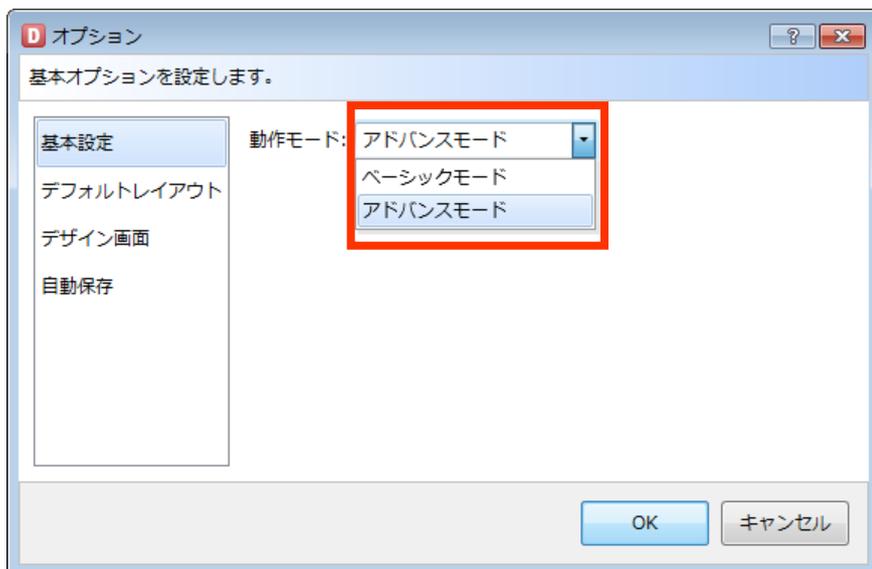
2. 印字を行うバーコード種類を選択し、1次元バーコード部の仕様に沿ったキャラクタコード、桁数を指定することでGS1 DataBarが表示されます。



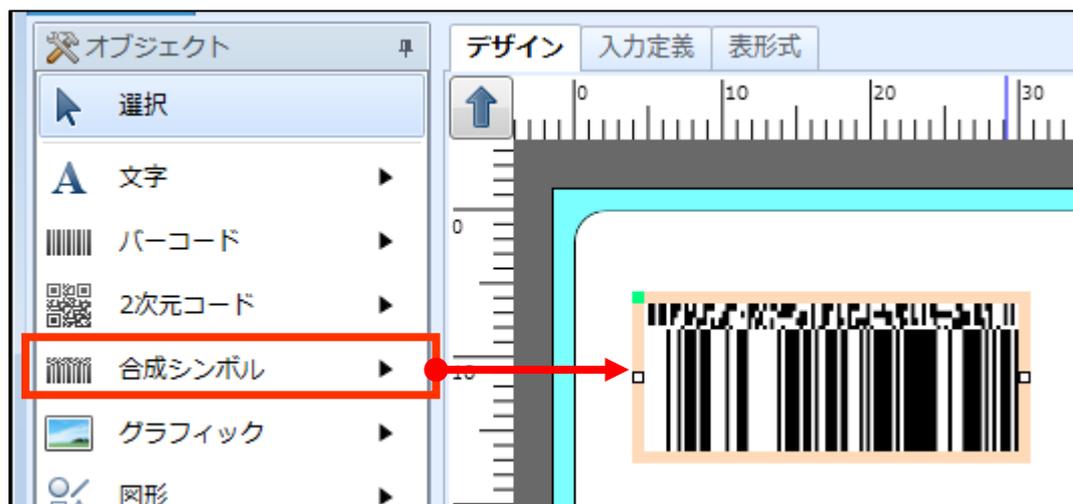
■GS1 合成シンボル指定

1. ML デザインのファイルメニューから「オプション」を選択し、動作モードを「アドバンスモード」に設定します。





2. 合成シンボルオブジェクトをデザイン画面に配置します。



3. 合成シンボルオブジェクトを選択し、プロパティのプルダウンメニューからバーコード種を選択します。



3. GS1 合成シンボルの印字を指定する場合には 1 次元バーコード部のデータと 2 次元コードのデータをそれぞれ指定してください。

例) 1 次元 : 0123456789012、2 次元 : 1707100210ABC123



3. 補足説明

■GS1 合成シンボルでの FNC1 指定

- FNC1(GS)をデータとして指定する場合は、'#' (23h)を指定してください。

■GS1-128 Composite(CC-A/CC-B)及び(CC-C)のバー高さ指定

- GS1-128 Composite (CC-A/CC-B)及び(CC-C)のみ 1次元バーコード部のバー高さが指定できます。倍率を「1」とした場合のバーコード高さとなります。

なお、指定できる最大値はプリンタのヘッド密度によって異なります。

ヘッド密度	高さ	プリンタ機種
8本/mm	62.5mm	CL4NX-J 08、T/R408v-ex など
12本/mm	41.7mm	CL4NX-J 12、T/R412v-ex など
24本/mm	20.85mm	CL4NX-J 24、HA224R など

- 合成シンボルオブジェクトを選択し、プロパティの基本設定で指定します。
例) 高さ「5」、倍率「2」の場合、1次元バーコード部の高さは「10mm」になります。



■変数の使用について

- GS1-128、GS1 DataBar、GS1 DataBar Composite でシンボル変数が利用できます。その他のGS1 合成シンボルは結合変数を使用してください。
- シンボル変数は、自動的に先頭や可変データの末尾に FNC1 を付加し、AI に応じて最適なコードセットに切り替えてデータを生成します。

■データに関する制限について

- 各バーコード種類によって指定できるデータの種類が異なりますのでご注意ください。

指定可能な文字
1次元バーコード部
バーコード種：GS1 DataBar Expanded、GS1-128 Composite
英数字、半角スペース、記号 29 文字* * 「!」 「"」 「\$」 「%」 「&」 「'」 「(」 「)」 「*」 「+」 「,」 「-」 「.」 「/」 「:」 「;」 「<」 「=」 「>」 「?」 「@」 「[」 「¥」 「]」 「_」 「`」 「{」 「}」 「~」
バーコード種：上記以外の GS1 DataBar、GS1 合成シンボル
数字のみ
2次元コード部（すべて共通）
英数字、半角スペース、記号 21 文字* * 「!」 「"」 「%」 「&」 「'」 「(」 「)」 「*」 「+」 「,」 「-」 「.」 「/」 「:」 「;」 「<」 「=」 「>」 「?」 「_」 「#」（#は FNC1 となります。）

- GS1-128 Composite (CC-A/CC-B)の場合、2次元コード部が56桁以下の場合はCC-A、57～338桁の場合はCC-Bとして自動的に判別されます。
- GS1 DataBar の1次元バーコード部の先頭に必要な AI (アプリケーション識別子)「01」は、スキャナが付加して出力します。但しGS1 DataBar Expanded にはデータとして指定する必要があります。
- GS1 Databar の1次元バーコード部のチェックディジットは自動補填されます。但し、GS1 Databar Expanded のみデータとしてチェックディジットを指定する必要があります。

■読み取り確認のお願い

- GS1 合成シンボルは種類が多く、データの構成も複雑になりますので、必ずスキャナによるデータのご確認をお願いいたします。

以上で「[9：GS1 DataBar 及び GS1 合成シンボル印字機能](#)」は終了です。

10：バイナリ情報設定機能（二次元コード）

1. バイナリ情報設定機能とは

バイナリ情報とは文字では表現できないキャラクタのことであり、二次元コードではこれらを表現して利用することができます。このような、バイナリ情報を付加した各種シンボルの印字を可能とするのがバイナリ情報設定機能になります。

バイナリ情報設定を利用する場合は、[動作モードをアドバンスモードに変更してください。](#)

例えば、原材料メーカなどでは QR コードにバイナリ情報を付加してトレーサビリティシステムを実現している事例があります。以下はその QR コード仕様の一例になります。次ページより具体的な設定方法を記載しておりますが、内容は以下の QR コード仕様に基づき記載しています。

表示シンボル：QR コード

格納データ：

[]>【RS】05【GS】01（GTIN）【GS】17（保証年月日）【GS】11（製造年月日）【GS】10（ロット番号）【GS】412（GLN）【RS】【EOT】

※以下がバイナリ情報になります。

- 【RS】 レコードセパレータ（1E）
- 【GS】 グループセパレータ（1D）
- 【EOT】 メッセージトレーラ（04）

2. レイアウトの設計方法

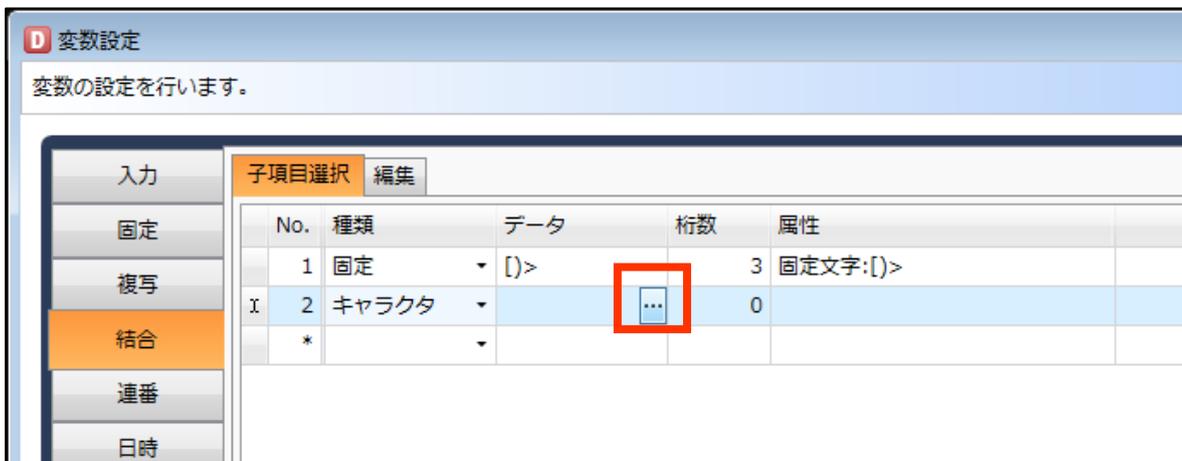
■結合変数で作成する

1. データを入力する変数を作成します。
以下の設定で各項目の入力変数を作成します。



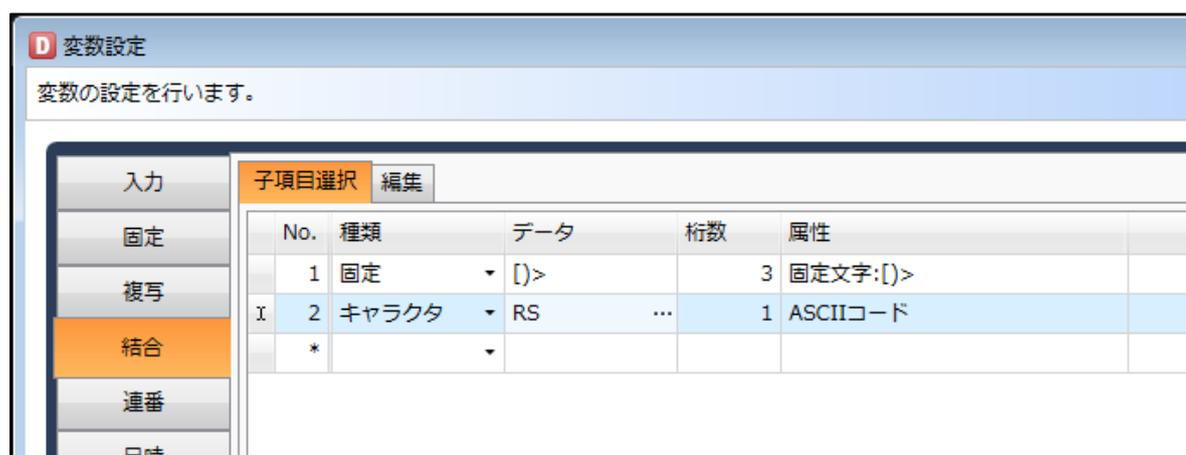
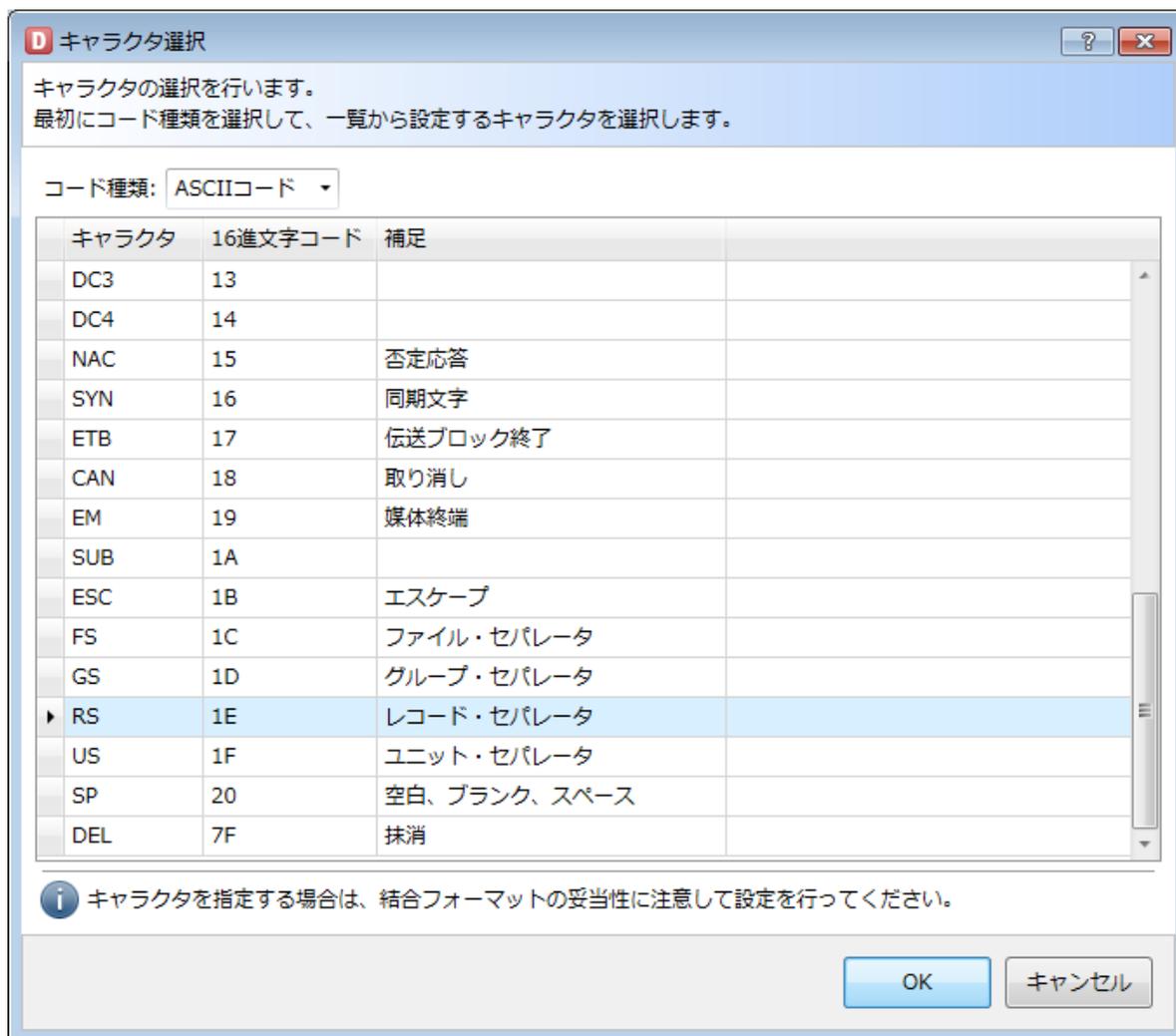
変数名	区分	桁数	詳細
GTIN	入力	14	入力種類:文字
保証年月日	入力	6	入力種類:文字
製造年月日	入力	6	入力種類:文字
ロット番号	入力	20	入力種類:文字
GLN	入力	13	入力種類:文字

2. QR コードで利用するために、各項目を結合変数にて一つにまとめます。
バイナリ情報の入力は、種類で「キャラクタ」を選択し、データの「…」をクリックします。

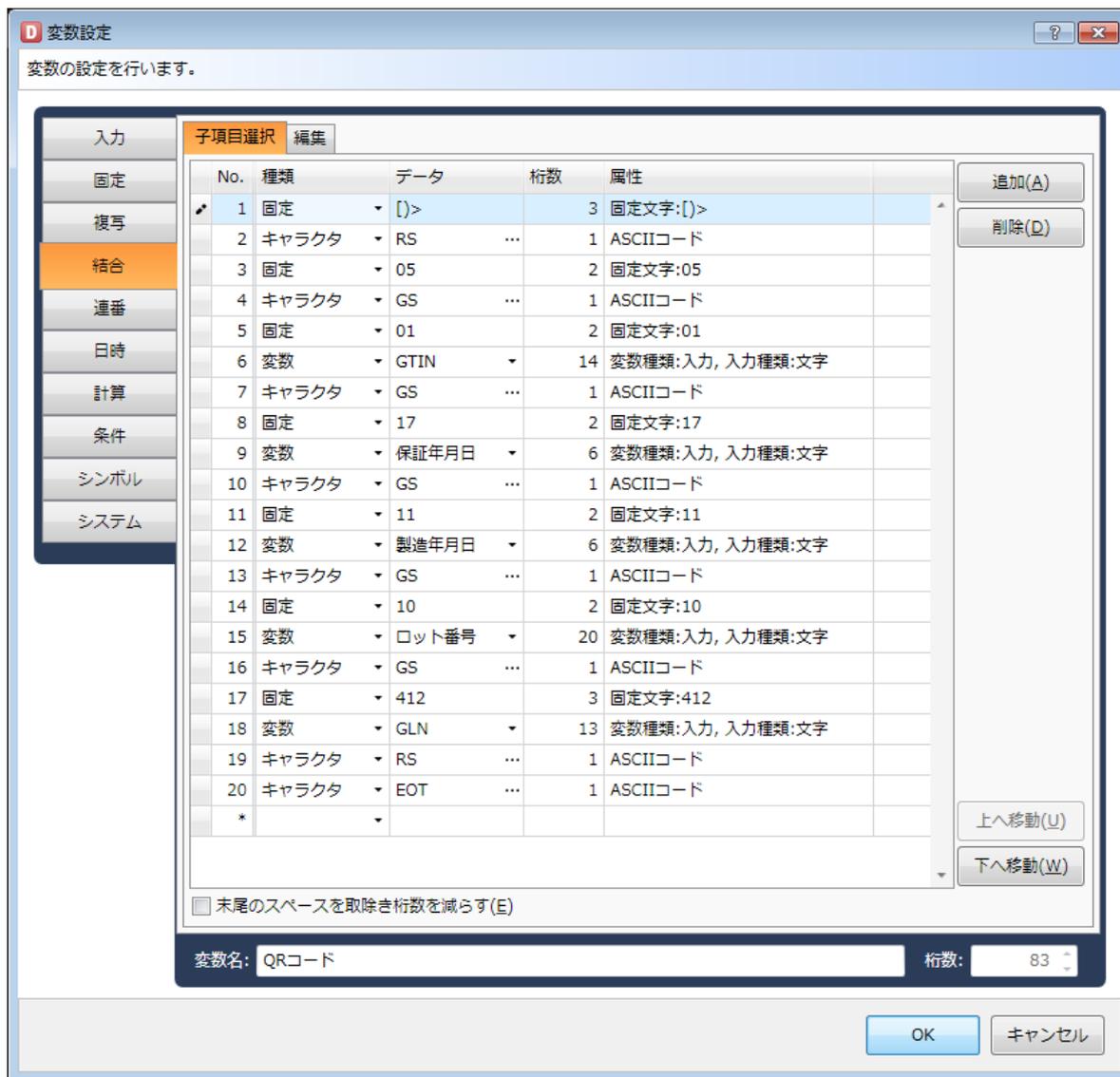


No.	種類	データ	桁数	属性
1	固定	[]>	3	固定文字:[]>
2	キャラクタ	...	0	
*				

3. バイナリ情報のキャラクタを選択して、「OK」ボタンをクリックします。



4. 同様の手順で、バイナリ情報と入力項目を組合せてデータを作成します。



5. デザイン画面でQRコードに作成した結合変数を設定して完成です。



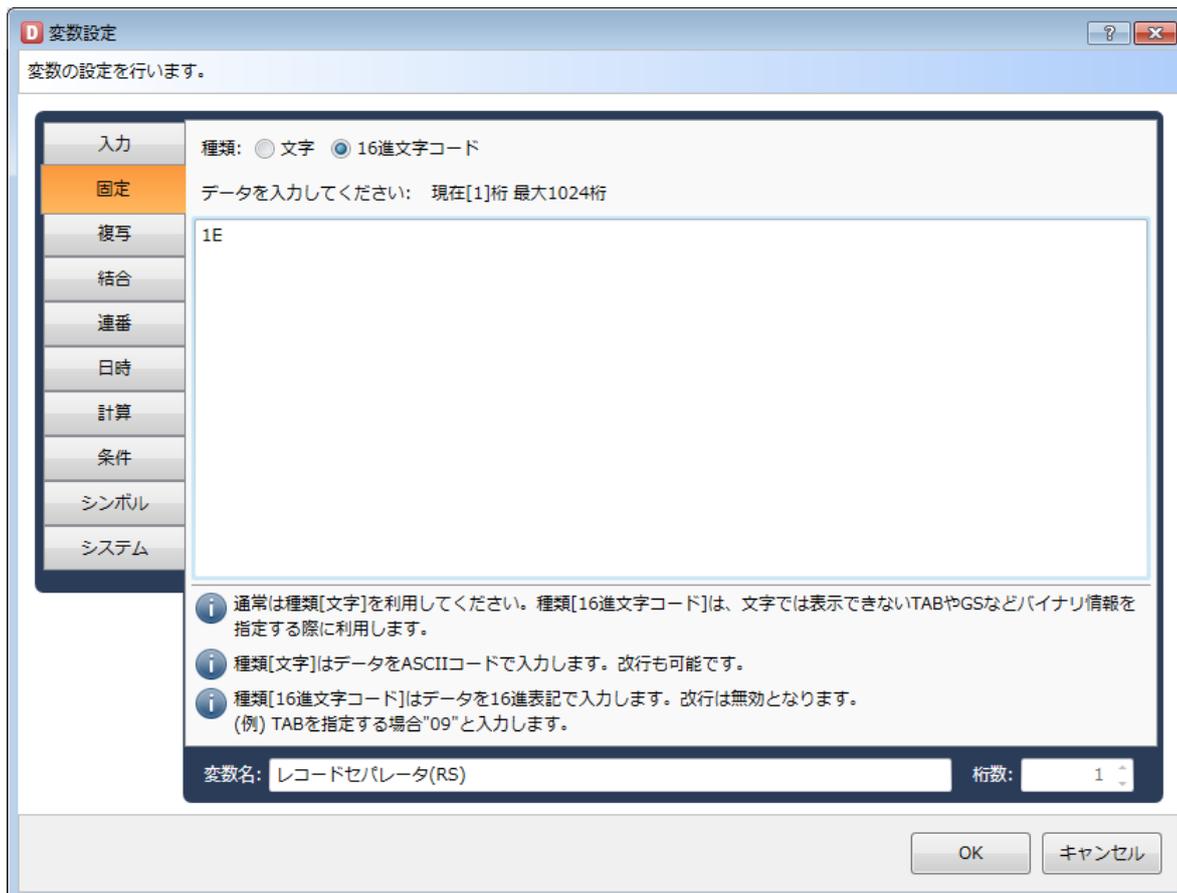
■固定変数で作成する

1. データを入力する変数を作成します。
以下の設定で各項目の入力変数を作成します。



変数名	区分	桁数	詳細
GTIN	入力	14	入力種類:文字
保証年月日	入力	6	入力種類:文字
製造年月日	入力	6	入力種類:文字
ロット番号	入力	20	入力種類:文字
GLN	入力	13	入力種類:文字

2. バイナリ情報を固定変数で作成します。
種類で「16進文字コード」を選択し、データに16進数の文字列を入力します。
ここではレコードセパレータを「1E」で入力しています



変数の設定を行います。

入力 種類: 文字 16進文字コード

固定 データを入力してください: 現在[1]桁 最大1024桁

1E

通常は種類[文字]を利用してください。種類[16進文字コード]は、文字では表示できないTABやGSなどバイナリ情報を指定する際に利用します。

種類[文字]はデータをASCIIコードで入力します。改行も可能です。

種類[16進文字コード]はデータを16進表記で入力します。改行は無効となります。
(例) TABを指定する場合"09"と入力します。

変数名: レコードセパレータ(RS) 桁数: 1

OK キャンセル

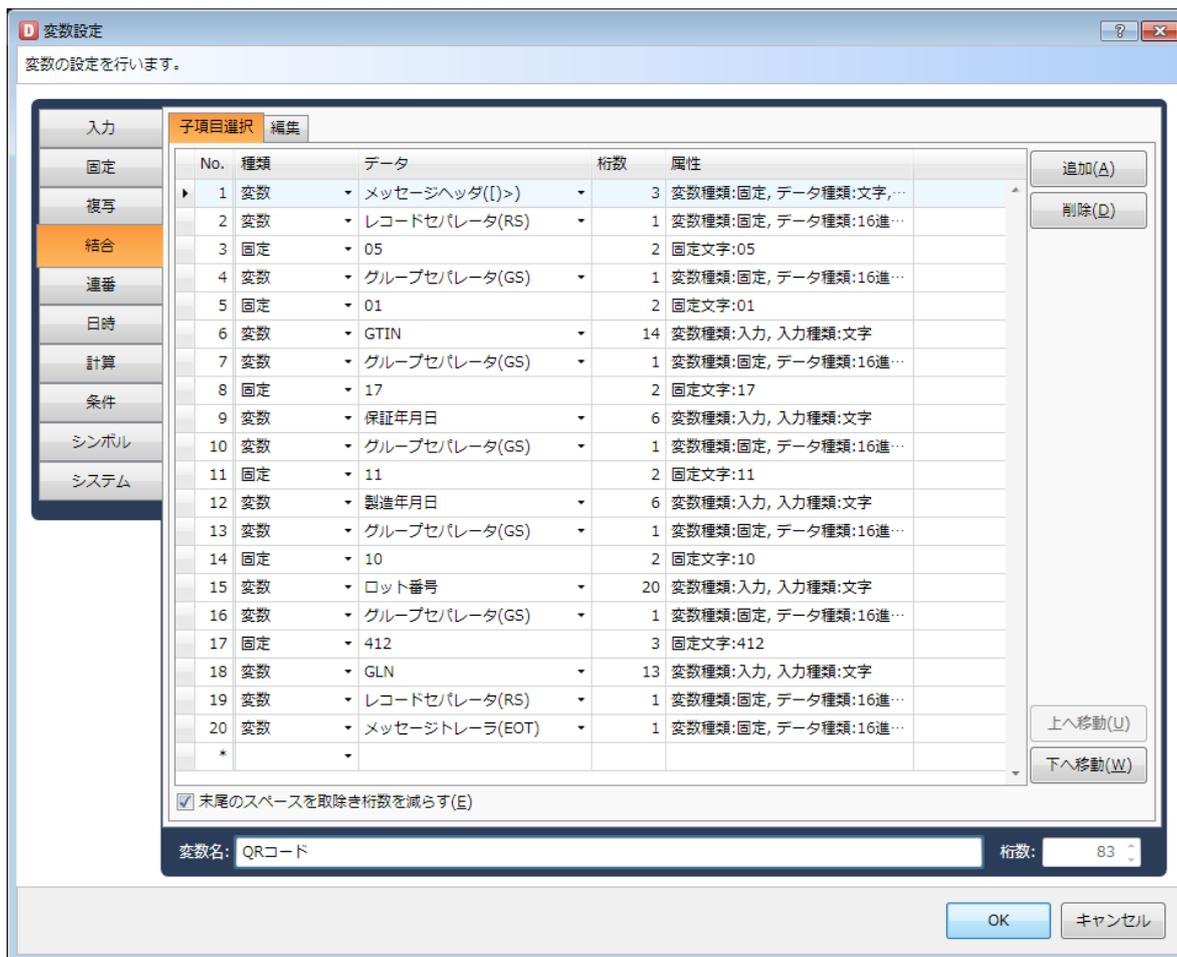
3. 変数の作成が完了すると変数一覧に表示されます。

変数名	区分	桁数	詳細
GTIN	入力	14	入力種類:文字
保証年月日	入力	6	入力種類:文字
製造年月日	入力	6	入力種類:文字
ロット番号	入力	20	入力種類:文字
GLN	入力	13	入力種類:文字
レコードセパレータ(RS)	固定	1	データ種類:16進...

引続き同様の手順で、必要な変数を作成していきます。

変数名	区分	桁数	詳細
GTIN	入力	14	入力種類:文字
保証年月日	入力	6	入力種類:文字
製造年月日	入力	6	入力種類:文字
ロット番号	入力	20	入力種類:文字
GLN	入力	13	入力種類:文字
レコードセパレータ(RS)	固定	1	データ種類:16進文字コード, 固定文字:1E
グループセパレータ(GS)	固定	1	データ種類:16進文字コード, 固定文字:1D
メッセージトレーラ(EOT)	固定	1	データ種類:16進文字コード, 固定文字:04
メッセージヘッダ(I)>	固定	3	データ種類:文字, 固定文字:[]>

4. QRコードで利用するために、各項目を結合変数にて一つにまとめます。
作成した変数を順番に選択します。



5. デザイン画面でQRコードに作成した結合変数を設定して完成です。



以上で「[10: バイナリ情報設定機能 \(二次元コード\)](#)」は終了です。

11：RFID 書込み機能を使ってみよう

RFID 書込み機能を使って IC タグにデータを書込んでみましょう。

1. RFID 書込み機能とは

RFID 書込み機能とは、IC タグにデータを書込む機能のことです。

RFID 書込みを利用する場合は、[動作モードをアドバンスモードに変更してください。](#)

対応プリンタは、CL4NX-J シリーズ、レスプリ R412v-RFID、エヴィ EV212R-RFID のRFID 対応機になります。

2. IC タグの種類と容量

各 IC タグのメーカーと容量は次のようになります。

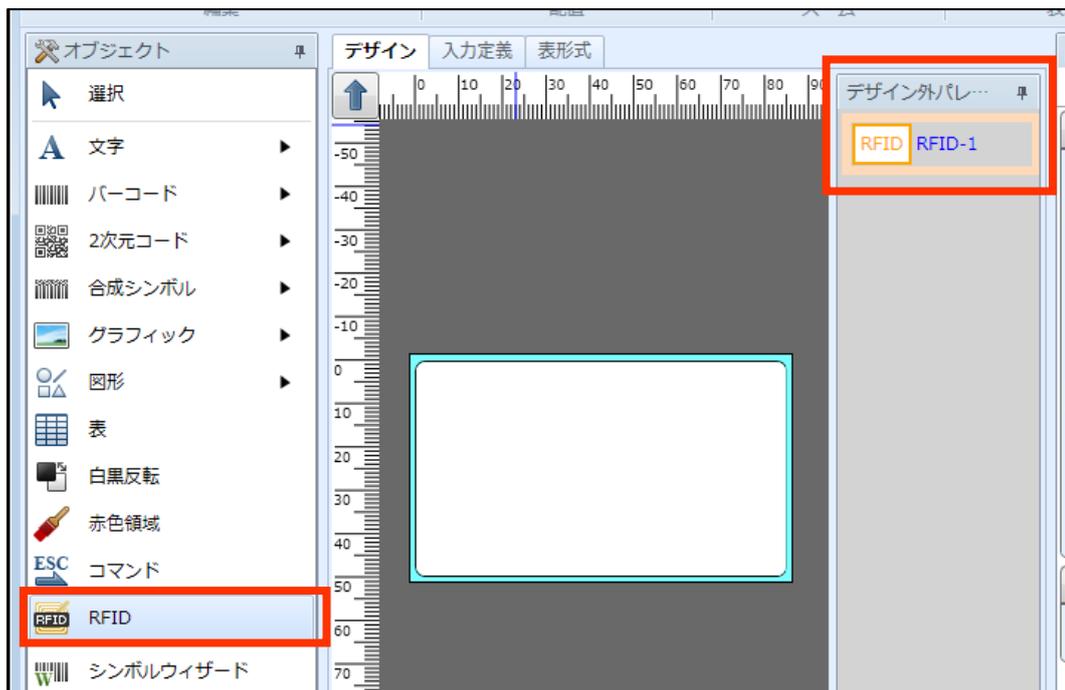
容量を超えた情報を指定した場合、超過分のデータは書込みを行いませんのでご注意ください。

IC タグの種類	メーカー	容量 (byte)
ICODE SLI	Philips Semiconductors	112
Tag-it HF-I	Texas Instruments	256
my-d *	Infineon	1000

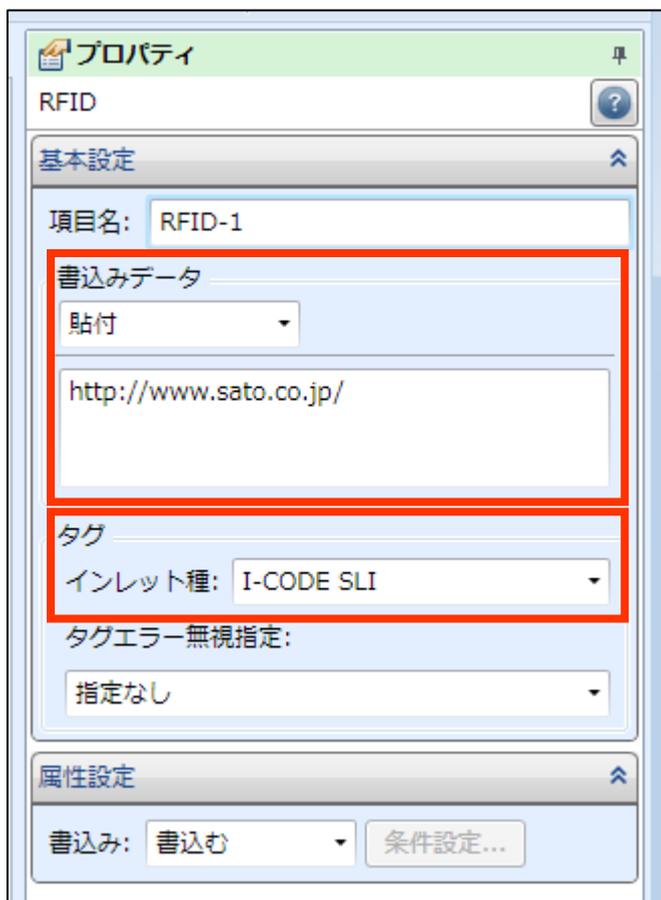
* my-d は最大登録容量が 232byte と 1000 byte の 2 種類のインレットがあります。
232byte のタグに対して最大登録容量を超えてデータ書込みを行うと RFID タグエラーとなります。

3. RFIDに書込んでみよう

デザインオブジェクトから「RFID」を選択してデザイン画面に配置すると、デザイン外パレットが表示され、RFIDのアイコンが追加されます。



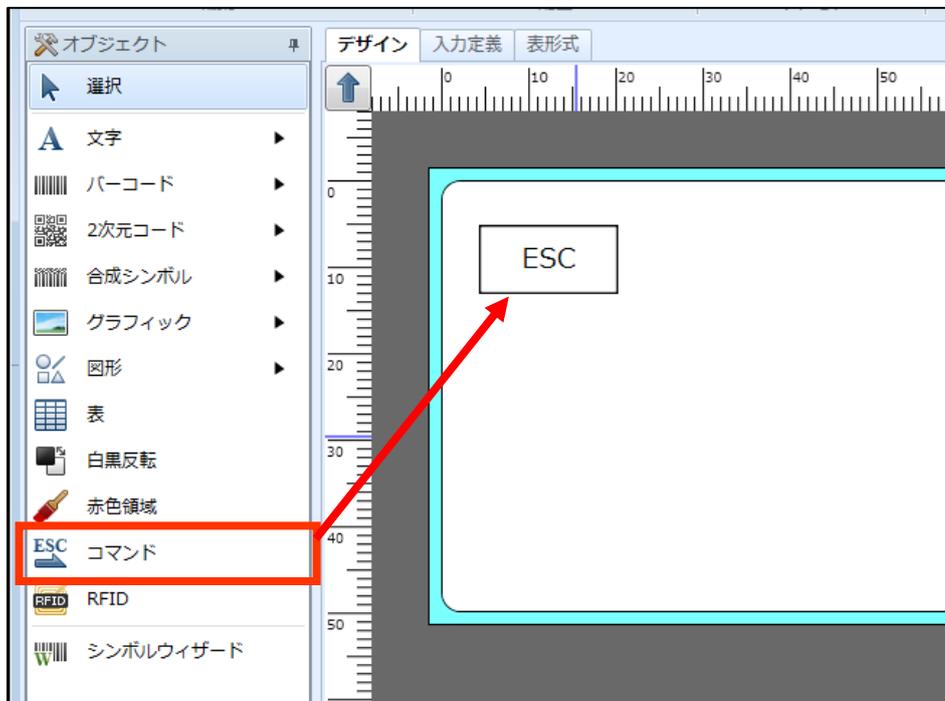
プロパティウィンドウで書き込むデータを入力し、使用する「インレット種」を選択します。



4. その他の IC タグへの書込み

I-CODE SLI、Tag-it HF-l、my-d 以外の IC タグ種を利用する場合は、コマンドオブジェクトでプリンタ用コマンド（SBPL）を記述して書込みを行います。

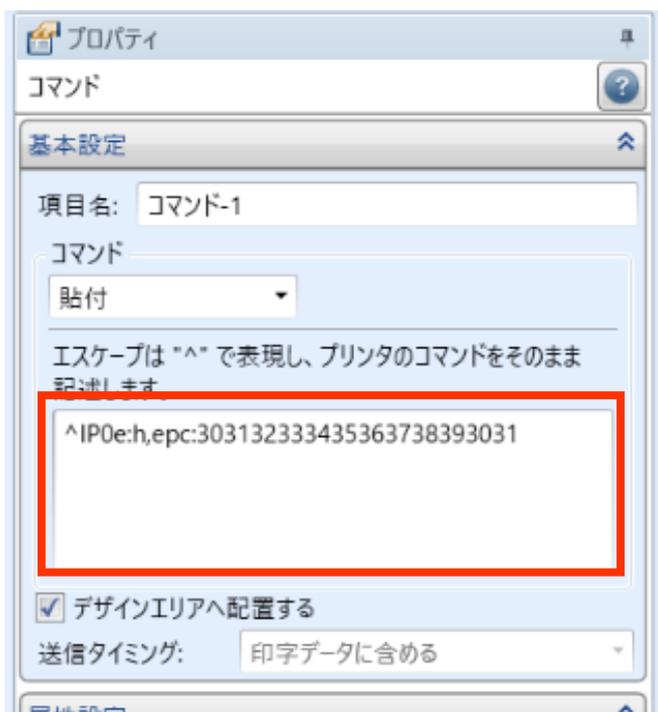
デザインオブジェクトから「コマンド」を選択してデザイン画面に配置すると、ESC のアイコンが追加されます。



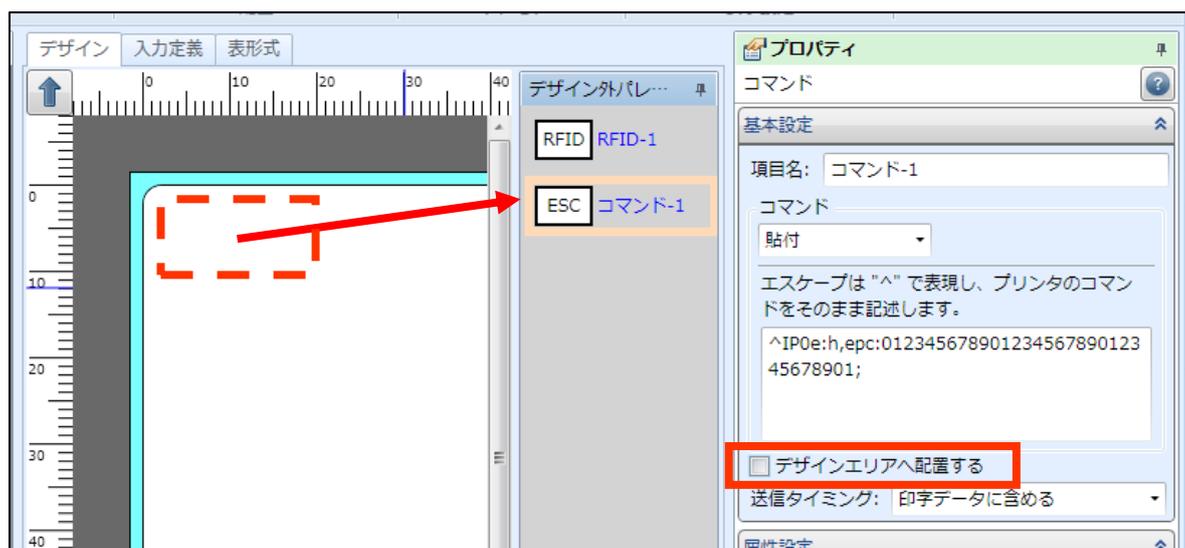
プロパティウィンドウで書き込むプリンタコマンド（SBPL）*を設定します。

*SBPL の作成方法については弊社販売拠点までご連絡ください。

例：IC チップメーカー拡張仕様タグの EPC 領域にデータを任意の長さで書き込むコマンドを指定



RFID コマンドは位置情報が必要ないため、「デザインエリアへ配置する」のチェックをはずします。ESC アイコンがデザイン画面からデザイン外パレットに移動します。



発行時に、設定した内容が IC タグに書き込まれます。

以上で「[1 1 : RFID 書込み機能を使ってみよう](#)」は終了です。

12：条件変数を使ってみよう

条件変数を使って、入力桁数によって文字のサイズを変更する、入力項目数によって結合する文字を変更するなど、複雑な条件に従って印字内容を変えることができます。

1. 条件変数とは

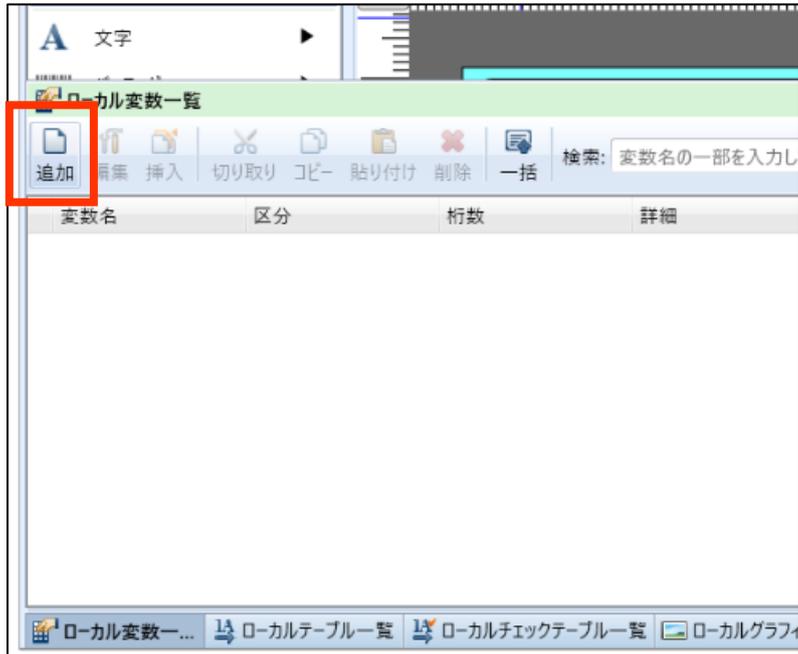
条件変数とは、条件式を設定して真偽判定（True/False）を行う機能のことです。条件に応じた値を印字したり、オブジェクトの印字条件として利用することもできます。

条件変数を利用する場合は、[動作モードをアドバンスモードに変更してください。](#)

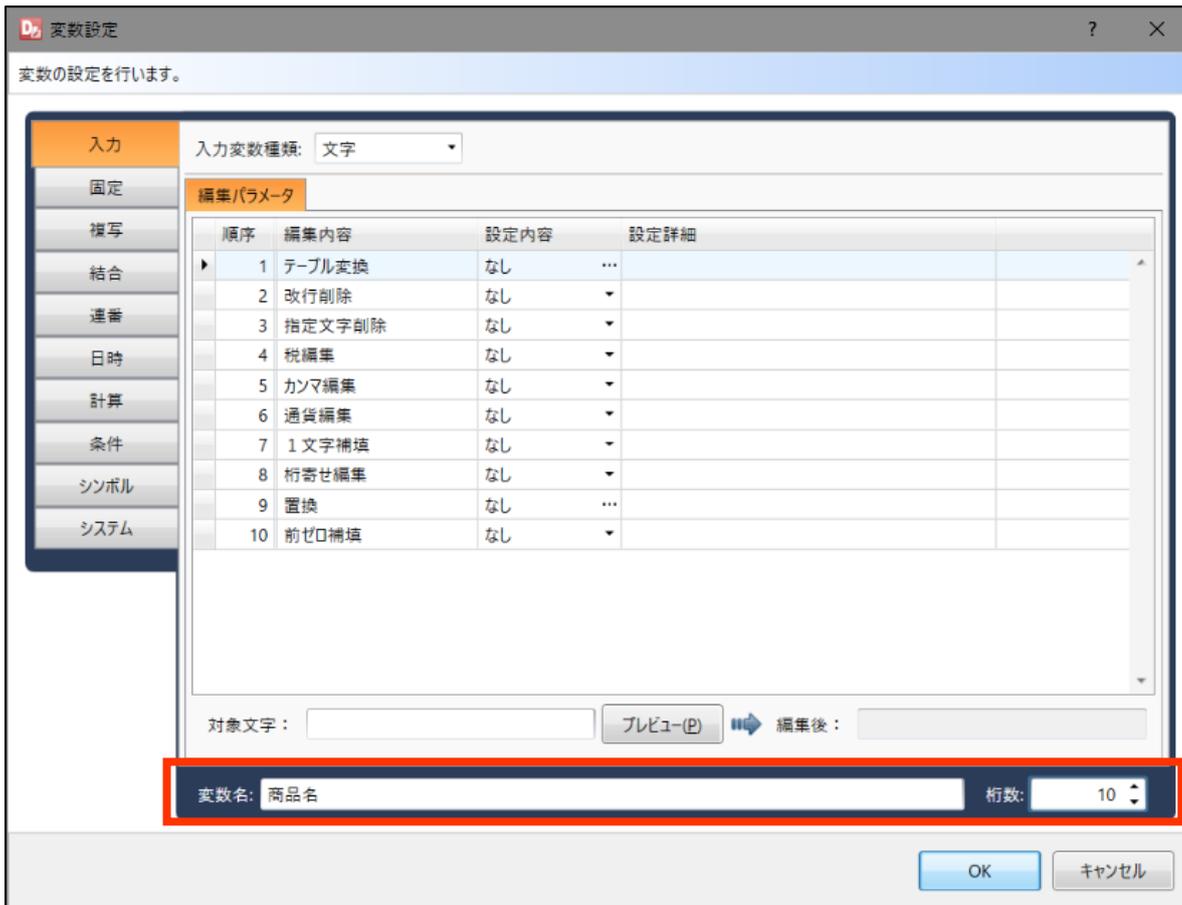
2. 条件変数の作成

入力桁数が 5 桁より大きい場合に、フォントサイズが小さくなるように、「条件変数」と「オブジェクトの印字条件」を作成しましょう。

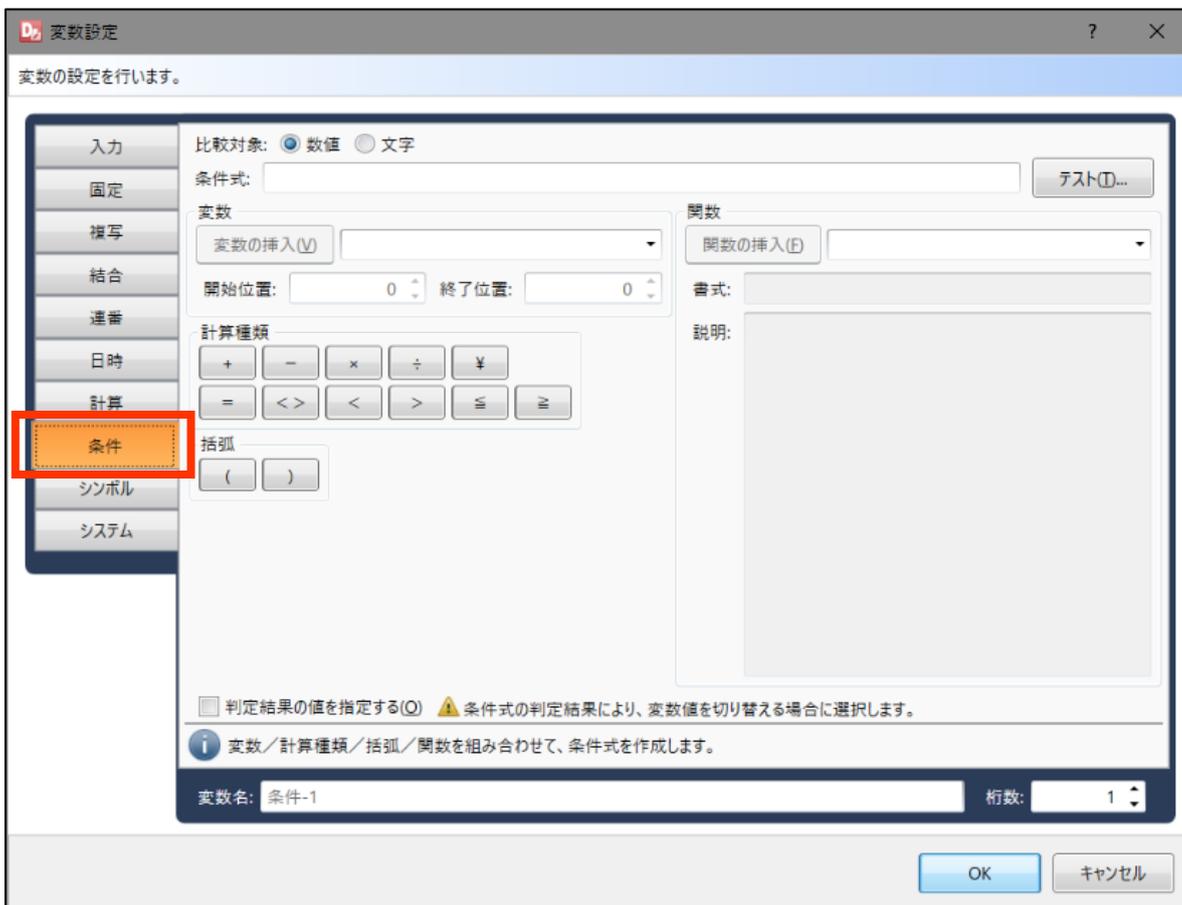
1. 変数の追加を選択します。



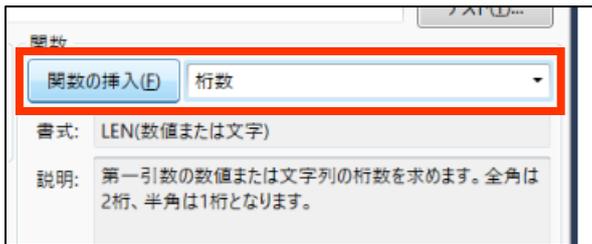
2. 入力変数で変数名「商品名」、桁数「10」で作成します。



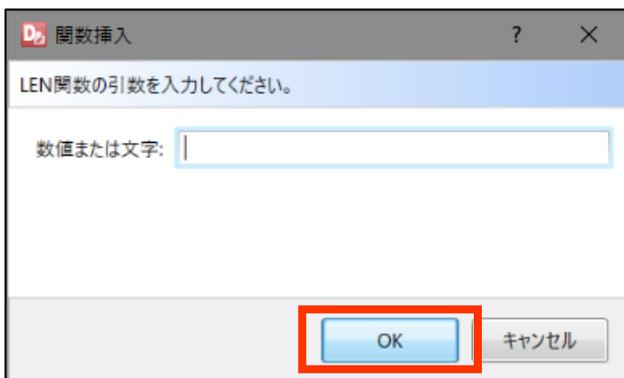
3. 続けて、条件変数を作成します。



4. 関数リストから「桁数」を選択し、「関数の挿入」を押します。



5. 「OK」ボタンをクリックします。



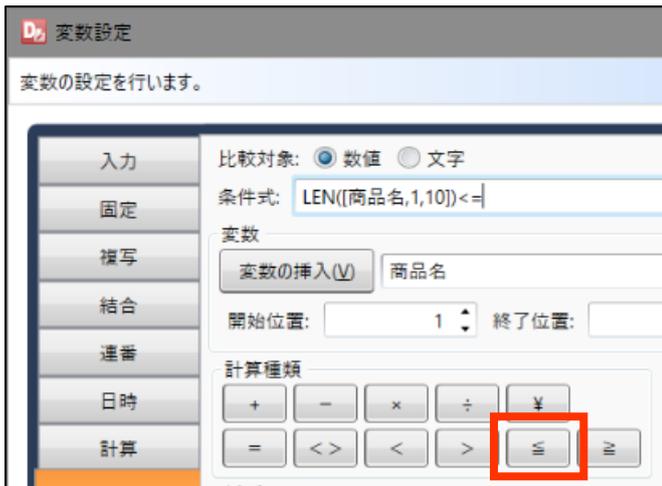
6. 条件式の「LEN()」の「()」の中にカーソルを合わせます。



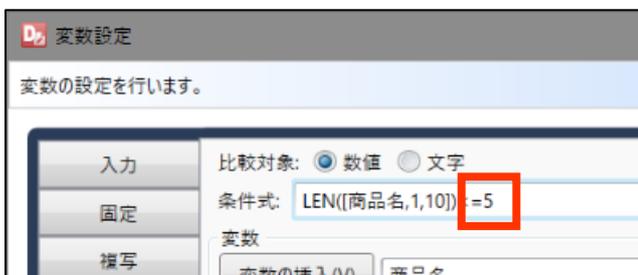
7. 変数リストから「2」で作成した入力変数を選択し、「変数の挿入」を押します。



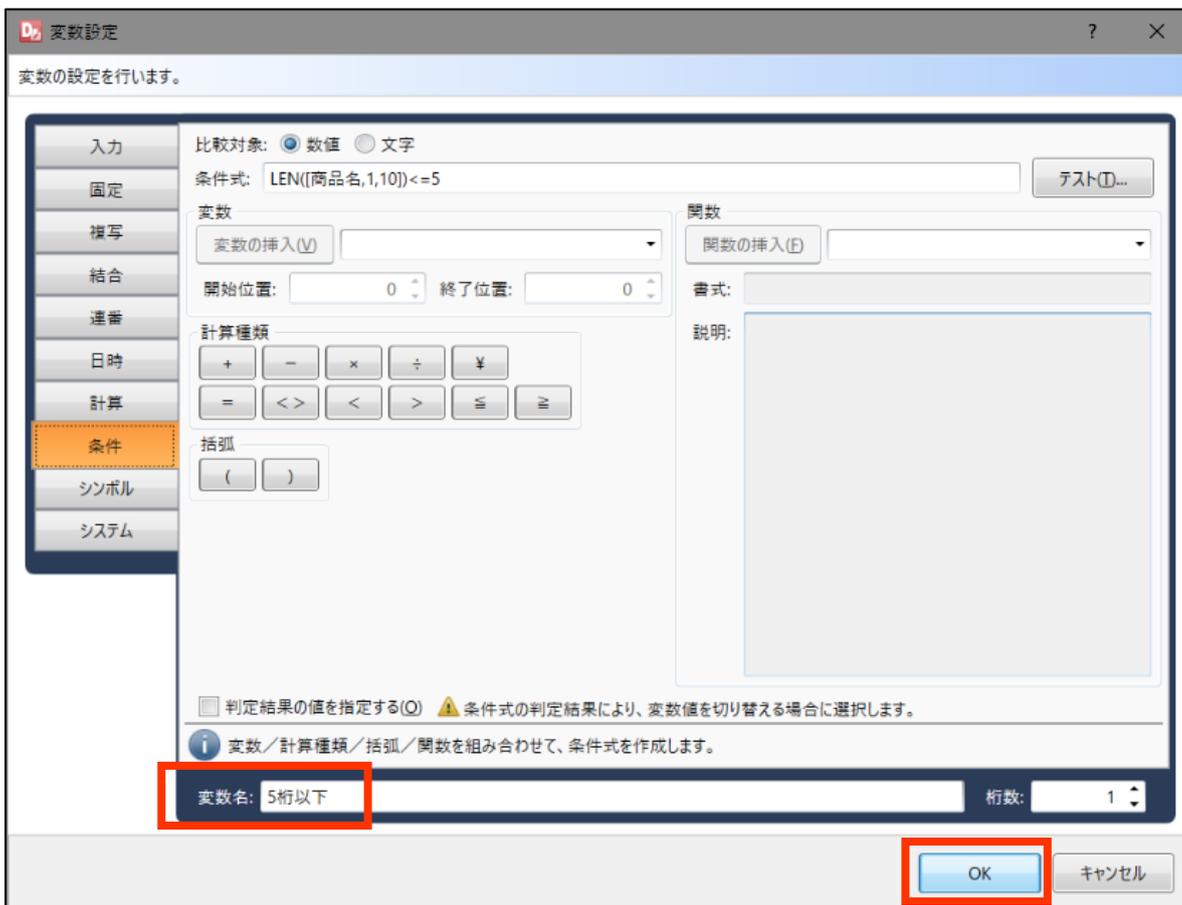
8. 条件式の末尾をクリックし、計算種類で「≤」を押します。



9. 条件式の末尾に「5」を入力します。



10. 変数名「5桁以下」で「OK」ボタンをクリックします。



以上で変数作成は完了です。

次の項目で、条件変数と印字条件を組合せましょう。

条件式について

LEN([商品名,1,10]) ≤ 5

① ② ③

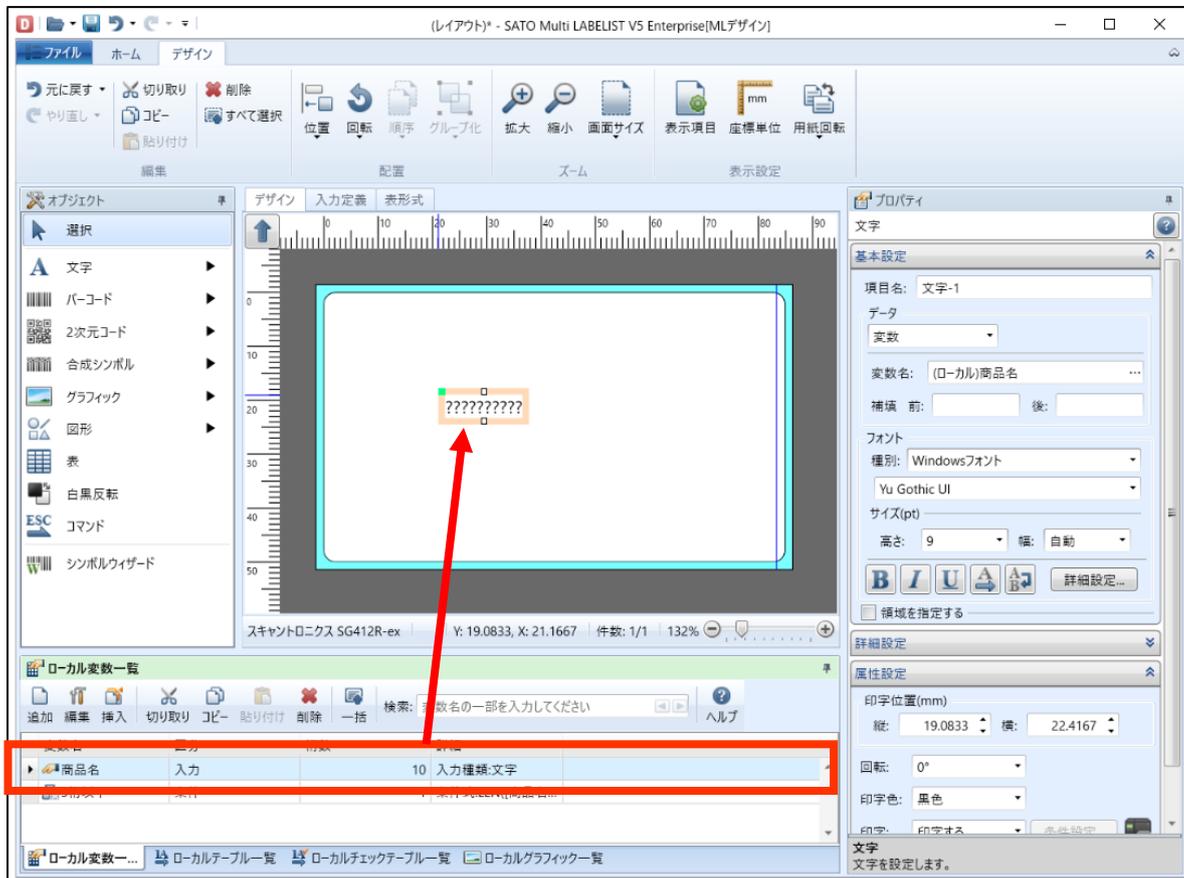
- ① **LEN(xxx)** : 括弧内に指定されたデータの桁数を求めます。
- ② **[xxx,a,b]** : 入力変数「xxx」の a 桁目から b 桁目を使用します。
- ③ **A ≤ B** : B より A が小さいか同じ場合は True、大きい場合は False となります。

設定された条件式は、商品名が 5 桁以下で True、5 桁より大きいと False となります。

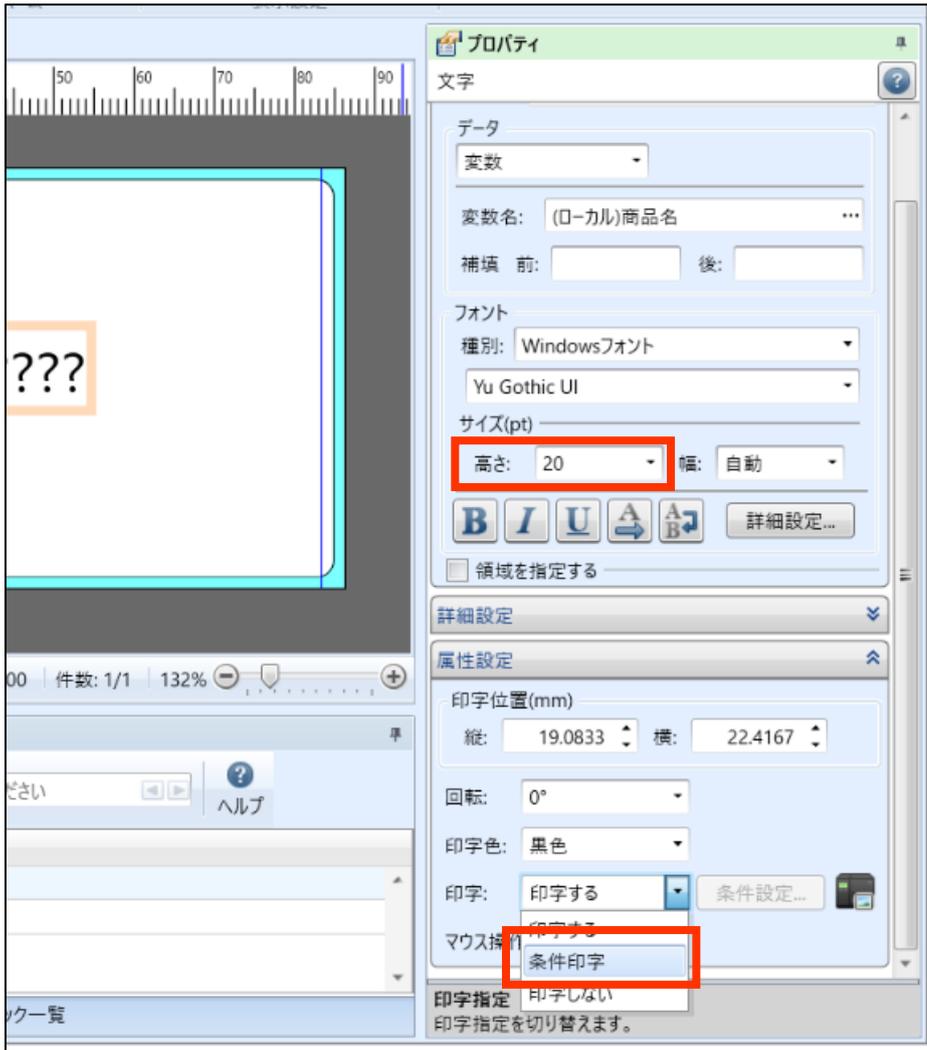
3. オブジェクトの作成

入力桁数が 5 桁より大きい場合に、フォントサイズが小さくなるように、フォントサイズの異なる文字を同じ位置に配置し、印字条件を設定しましょう。

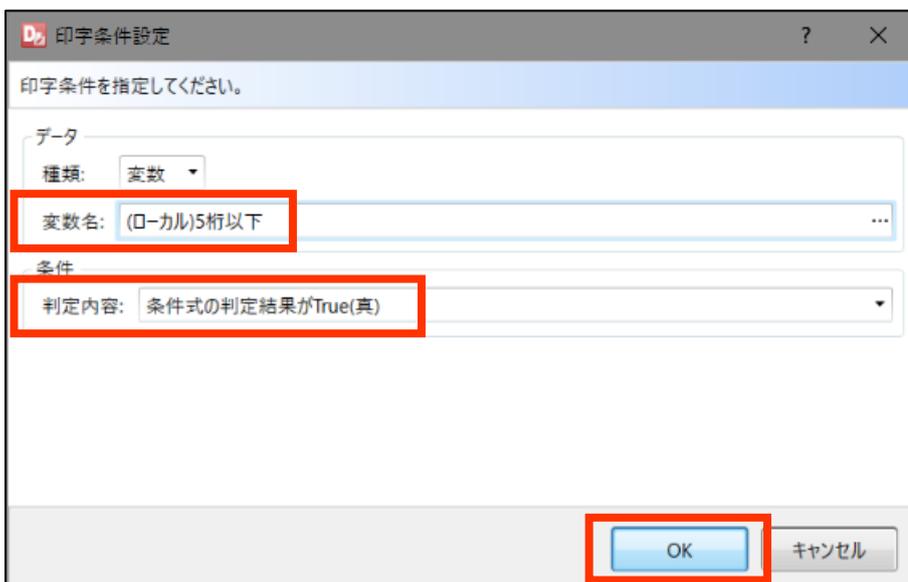
1. 変数一覧から入力変数をデザイン画面にドラッグ&ドロップします。



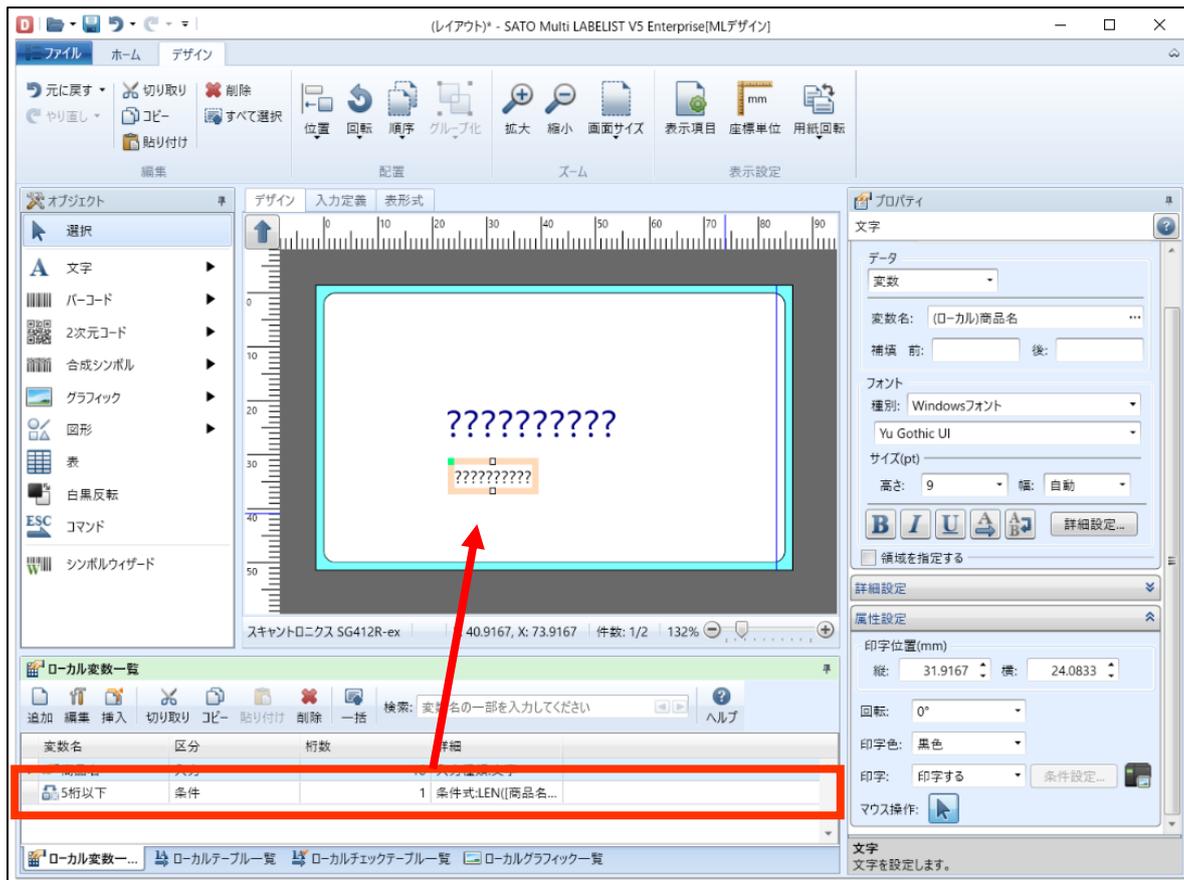
2. フォントサイズを「20」に設定し、印字設定で「条件印字」を選択します。



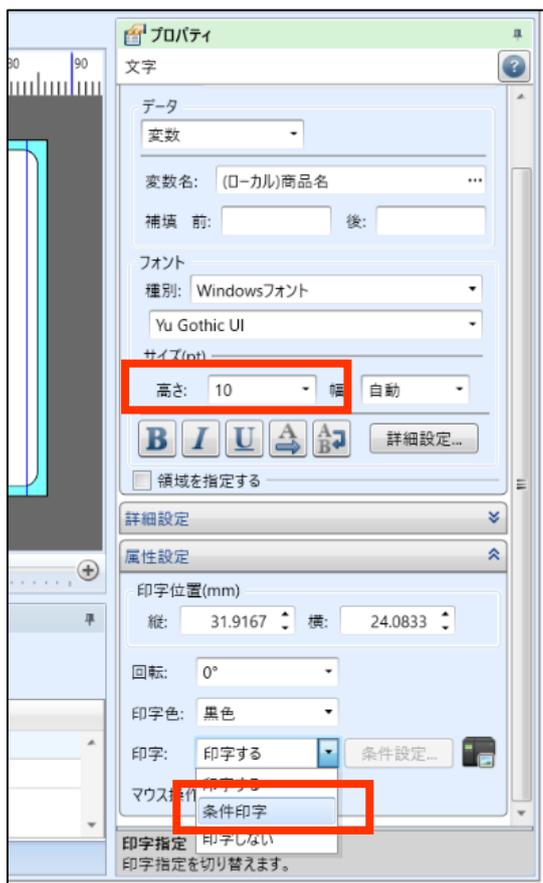
3. 変数名に条件変数を指定、判定内容で「条件式の判定結果が True(真)」を選択し、「OK」ボタンをクリックします。



4. 変数一覧から入力変数をデザイン画面にドラッグ&ドロップします。



5. フォントサイズを「10」に設定し、印字設定で「条件印字」を選択します。



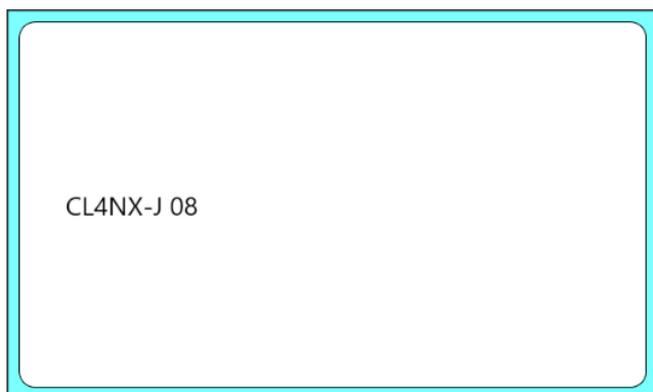
4. 印字確認

先ほど作成したレイアウトファイルを ML プリントで起動します。
設定した条件どおりに印字されるか、実際に発行して確認を行って下さい。

商品名に「PW208」を入力して発行すると、
商品名が 5 文字以下の条件に合致し、フォントサイズが 20 で印字されます。



商品名に「CL4NX-J 08」を入力し発行すると、
商品名が 5 文字以下の条件に合致し、フォントサイズが 10 で印字されます。



以上で「[12：条件変数を使ってみよう](#)」は終了です。

1 3 : GS1 DataMatrix を使ってみよう

1. GS1 DataMatrix とは

GS1 DataMatrix とは、AI（アプリケーション識別子）を使用して、決められた書式に従って作成したデータを DataMatrix で印字したシンボルです。

2. 結合変数の作成

GTIN、保証期限日、ロット No. を結合変数で作成しましょう。

1. データを入力する変数を作成します。
以下の設定で各項目の入力変数を作成します。

変数名	区分	桁数	詳細
GTIN	入力	14	入力種類:文字
保証期限日	入力	6	入力種類:文字
ロットNo.	入力	20	入力種類:文字

2. 結合変数で種類「固定」のデータ「01」を入力します。

No.	種類	データ	桁数	属性
1	固定	01	2	固定文字:01

3. 次項目で種類「変数」、データ「GTIN」を選択します。

D₂ 変数設定
変数の設定を行います。

入力 子項目選択 編集

No.	種類	データ	桁数	属性
1	固定	01	2	固定文字:01
2	変数	GTIN	14	変数種類:入力, 入力種類:文字
*				

結合 連番 日時 計算 シンボル

4. 同様な方法でAI17で「保証期限日」、AI10で「ロットNo.」を設定します。

D₂ 変数設定
変数の設定を行います。

入力 子項目選択 編集

No.	種類	データ	桁数	属性
1	固定	01	2	固定文字:01
2	変数	GTIN	14	変数種類:入力, 入力種類:文字
3	固定	17	2	固定文字:17
4	変数	保証期限日	6	変数種類:入力, 入力種類:文字
5	固定	10	2	固定文字:10
I 6	変数	ロットNo.	20	変数種類:入力, 入力種類:文字
*				

結合 連番 日時 計算 シンボル

5. 「各項目の末尾のスペースを取除き桁数を減らす」を有効にし、変数名「GS1 DataMatrix」を入力し、「OK」ボタンをクリックします。

変数の設定を行います。

No.	種類	データ	桁数	属性
1	固定	01	2	固定文字:01
2	変数	GTIN	14	変数種類:入力, 入力種類:文字
3	固定	17	2	固定文字:17
4	変数	保証期限日	6	変数種類:入力, 入力種類:文字
5	固定	10	2	固定文字:10
6	変数	ロットNo.	20	変数種類:入力, 入力種類:文字
*				

各項目の末尾のスペースを取除き桁数を減らす(E)

変数名: GS1 DataMatrix 桁数: 46

OK キャンセル

以上で変数作成は完了です。
次にオブジェクトに変数を設定して印字してみましよう。

■可変桁の末尾に FNC1 を設定する場合

可変桁の AI の後に別の AI を指定する場合は、データセパレータの FNC1 を指定する必要があります。結合変数で、種類「キャラクタ」を選択し、コード種類「シンボル」の一覧から「GS1 DataMatrix - FNC1」を選択することで FNC1 が利用可能です。

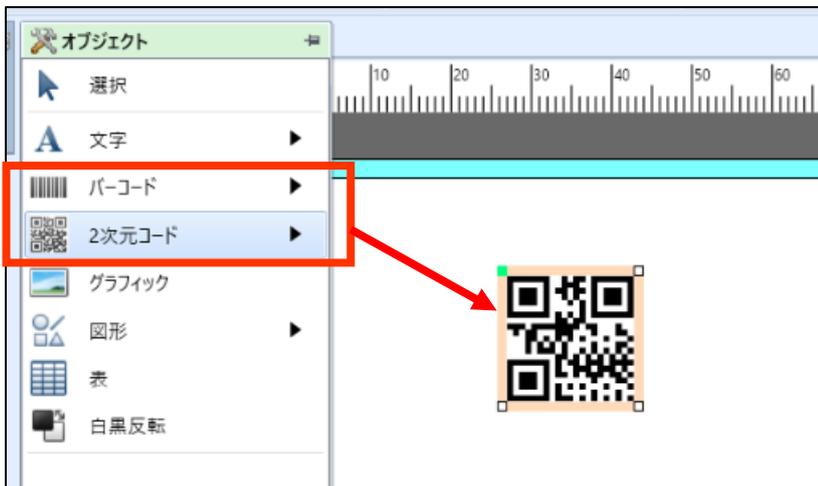
The screenshot shows the 'Character Selection' dialog box in the software. The 'Code Type' is set to 'Symbol'. The list below shows various character types, with 'GS1 DataMatrix - FNC1' selected. The 'Data' column for this selection is '1B,31'.

分類	キャラクタ	データ	補足
合成シンボル	FNC	#	
合成シンボル	区切り		1次元データと2次元データの区切り。
DataMatrix	~	7E,7E	
GS1 DataMatrix	~	7E,7E	
GS1 DataMatrix	FNC	1B,1B	이스ケープ
▶ GS1 DataMatrix	FNC1	1B,31	
識別子	識別子(AI/DI)	(nnnn)	バーコードオブジェクトでは0が省略されます。

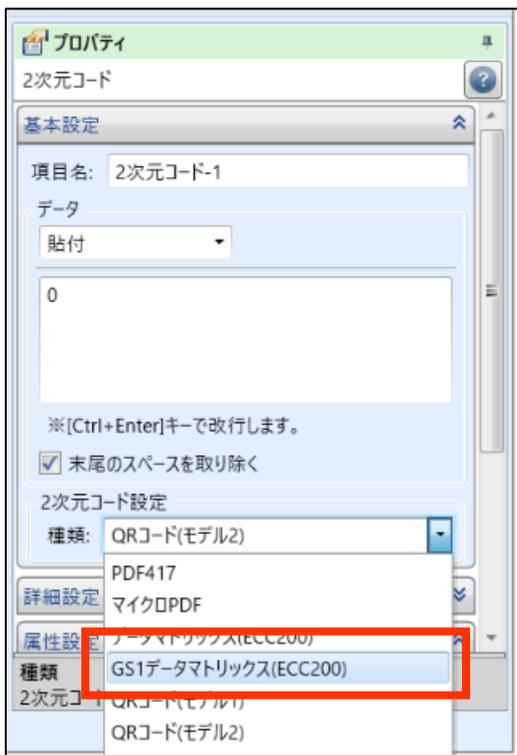
3. オブジェクトの作成

シンボル変数を2次元コードオブジェクトに設定して、GS1 DataMatrix を印字しましょう。

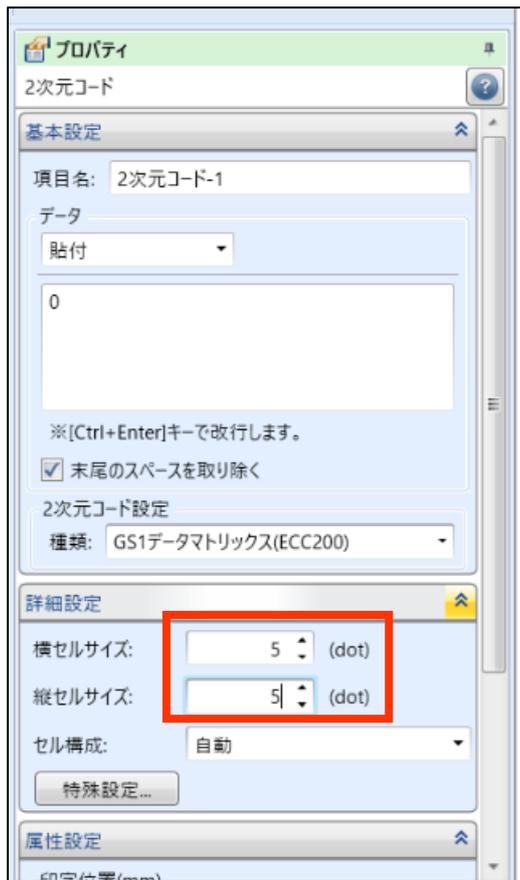
1. 2次元コードオブジェクトをデザイン画面に配置します。



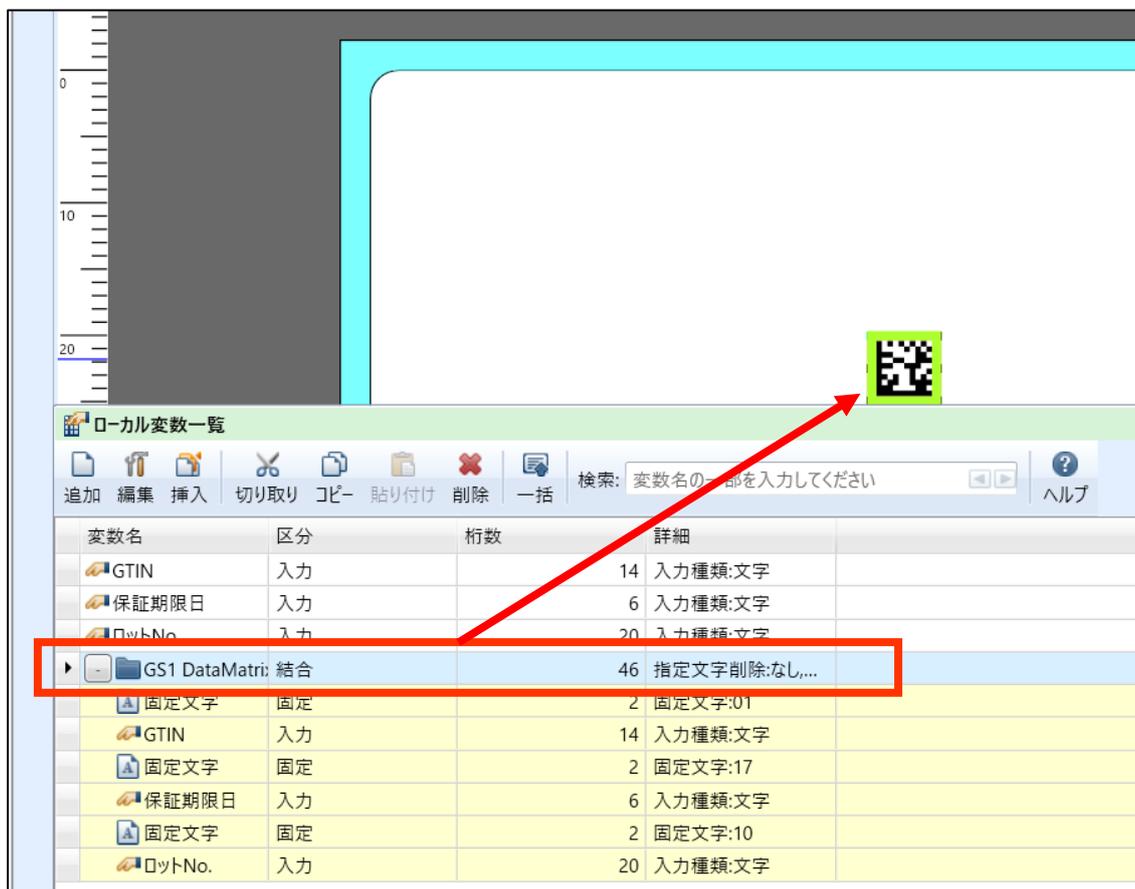
2. プロパティウィンドウで種類「GS1 データマトリックス(ECC200)」を選択します。



3. プロパティウィンドウで詳細設定を開き、横と縦のセルサイズを「5」(dot)にします。



4. 結合変数をドラッグ&ドロップします。



以上で 2 次元コードオブジェクトの作成は完了です。

最後に ML プリントでレイアウトを開き、GTIN、保証期限日、数量、ロット No.が格納された GS1 DataMatrix が印字されるか、実際に発行して確認を行って下さい。

GS1 の規格に従っているかは「GS1 Japan Scan」で簡易チェックすることが可能です。

<http://www.dsri.jp/application/js1japanscan/>

(例) GTIN「04993191000002」、保証期限日「201231」、ロット No.「A0123」を入力した場合



以上で「[1 3 : GS1 DataMatrix を使ってみよう](#)」は終了です。