Roboticsware



Automation シリーズ

# タグマニュアル

RWMN-5-21-910

**ようこそ** ..... **4**

本マニュアルについて ..... 4

**１章　タグの概要** ..... **5**

１－１　タグの概要 ..... 5
タグの位置付けと目的 ..... 5
タグを利用する場面 ..... 6
タグの階層構造とタグパス ..... 7
タグのデータ型 ..... 8
仮想デバイスについて ..... 9
スタティックタグとアクティブタグ ..... 10
ネットワークタグ ..... 11
命名規約について ..... 12

**２章　タグの作成手順と基本操作** ..... **13**

２－１　タグの作成とモニタリング ..... 13
数値タグの作成 ..... 13
現在値を画面に表示する ..... 14
ビットタグと文字列タグの作成 ..... 15
ネットワークタグの作成 ..... 17
２－２　タグを一括で作成する方法 ..... 19
タグの連続追加 ..... 19
CSV ファイルによるタグの編集 ..... 21

**３章　機能解説** ..... **25**

オンライン状態の切り替え ..... 25
値のモニタ ..... 26
値の書き込み ..... 26
更新周期と高速化についてのヒント ..... 28
更新周期の計測手順 ..... 29
接続・切断について ..... 30
仮想通信について ..... 31
デバイスダウンについて ..... 32
２重化について ..... 33
２重化における通信パラメータの切り替えについて ..... 34
書き込み後のキャッシュ更新について ..... 34
データ型と型変換について ..... 35
配列について ..... 36
デバイスの解釈 ..... 37
フィルタ（型変換／工学値変換など） ..... 41
パラメータ文字列 ..... 47
エイリアスタグについて ..... 49
タグコメントについて ..... 51

**４章　システムプロパティ** ..... **56**

概略 ..... 56
システムプロパティの使用例 ..... 56
システムプロパティ一覧 ..... 57

４－１　ユニット ..... 58
４－２　フォルダ ..... 60
４－３　タグ ..... 63

# ようこそ

## 本マニュアルについて

本書「タグマニュアル」は、本パッケージを使用する上での基礎となるものです。
ユーザーズマニュアルと併せてお読みいただくことを推奨いたします。

# １章　タグの概要

## １－１　タグの概要

### タグの位置付けと目的

本パッケージは、タグ、イベント、アクション、ビュー、インターフェースの５つの大きな概念から成り立ちます。本マニュアルは「タグ」について記述されています。

| | | |
|---|---|---|
|  | タグ | PLC との通信の設定です。 |
|  | イベント | アクションを起動するトリガの設定です。 |
|  | アクション | ロガー、メールなどのアクションの設定です。 |
|  | ビュー | タグリスト、タグモニタなどのビュー（画面）の設定です。 |
|  | インターフェース | OPC/DDE/IPLink などの、外部アプリケーションから接続するためのインターフェースの設定です。 |

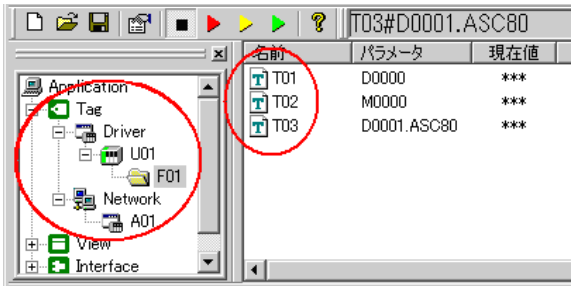


本パッケージは、現在１００機種相当の PLC との通信に対応しております。PLC 以外にも、ODBC データベースや Excel などに通信接続することもできます。そして、これらのデバイスとの通信はタグを介して行われます。また、タグは PLC との通信を行うという目的以外にも、内部計算用のワークエリアとしての利用や、複数のノード間でネットワークを介したデータのリンクを行うなど、さまざまな用途に利用することができます。

タグを定義することによる共通的な最大のポイントは、例えば、三菱 PLC 上にある信号やオムロン PLC 上にある信号、あるいは内部計算用のワークエリアについても、情報を利用する側から見ると全て同等に「タグ」としてフラットに扱うことができるという点です。

タグを設定することにより、PLC との通信が行えるようになります。例えば三菱 PLC で、「D0000」や「M0000」に対して通信を行いたい場合、それぞれにタグを作成します。

右の画面は、タグを設定している画面イメージです。



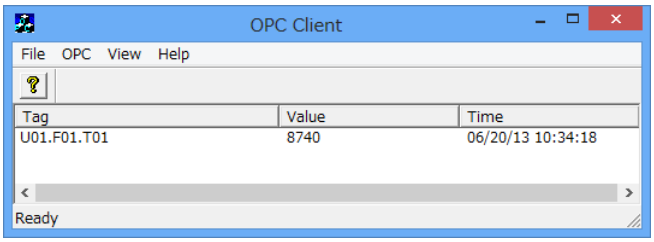
## タグを利用する場面

例えば「U01.F01.T01」というタグを 1 つ設定すると、下記のように様々な場面から利用することができます。

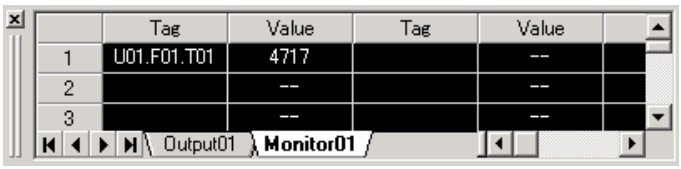
例）IPLink クライアントで　　IPLink1.ReadVal("U01.F01.T02", value, 0)

例）DDE で接続する際に　　=FASERVER|U01.F01!T01

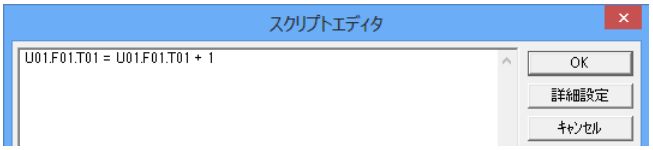
例）OPC のアイテム名として



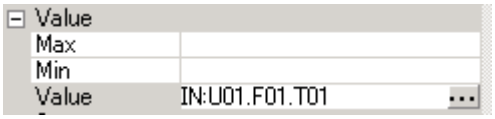

例）タグモニタのビューで




例）スクリプト Ver1 アクションで




例）FA-Panel のバインド式で




例）FA-Panel のスクリプトで

```
0001
0002  event OnMouseUp(button)
0003  {
0004    var vReadVal;
0005
0006    vReadVal = parent.GetVal("U01.F01.T00");
0007
0008    parent.LblReadVal.Text = vReadVal;
0009  }
```

## タグの階層構造とタグパス

タグは、「ユニット」「フォルダ」「タグ」という３階層の構造を持ちます。

■ユニット
ユニットは通信を行う PLC 毎に作成します。各 PLC 固有の接続設定情報を定義します。




■フォルダ
フォルダでは、各 PLC 固有のネットワークに関連する設定を行います。フォルダは必ずユニットの下に作成します。１つのユニットの下に複数のフォルダを作成することもできます。但し、フォルダのネスト構造（フォルダの下にさらにフォルダを作成すること）はできません。




PLC との通信周期をフォルダ単位で指定することもできます。例えば、複数のフォルダを作成して収集周期の異なるタグを分けて管理することなどができます。

■タグ
タグは必ずフォルダの下に作成します。個々のタグは、タグ名、データ型、データサイズ（ワード、ダブルワードなど）、PLC デバイスのアドレス、フィルタ情報（型変換や工学値変換など）などの詳細な設定値を持ちます。PLC に接続されたタグを登録した場合、タグを利用するアプリケーションからタグの値を参照する事によって、アプリケーションは PLC デバイスの値を知ることができます。同様に、アプリケーションからタグの値を変更する事によって、PLC デバイスの値を書き換えることができます。

■タグパス
上記の階層構造において、タグの参照は“タグパス”と呼ばれる書式に従って行われます。タグパスの書式は、ユニット、フォルダ、タグをピリオド「.」でつなげた文字列です。

タグパスの例：

```
U01.F01.T01
```




フォルダはタグをまとめて管理する用途以外にも重要な役割を持っています。例えば、RS-422 でのマルチドロップ接続や、各メーカー固有のネットワーク（MELSECNET など）で複数の PLC が連結されている場合、PLC 固有のネットワーク上に接続された PLC をフォルダの「ネットワーク設定」によって個別に管理することができます。詳細は「接続マニュアル」を参照ください。

## タグのデータ型

タグには、大きく分けて「ビットタグ」「数値タグ」「文字列タグ」の３つのタイプがあります。

標準 PLC では「アドレス設定」タブの「タイプ」によりタグの型が決定します。

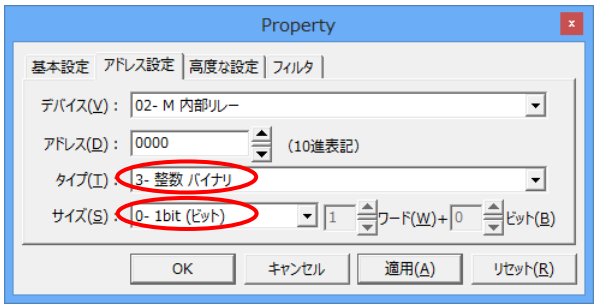
| タイプ | 型 |
|---|---|
| 0 - ビット | ビットタグ |
| 1 - 整数 BCD | 数値タグ |
| 2 - 整数 BCD (符号付き) | 数値タグ |
| 3 - 整数 バイナリ | 数値タグ |
| 4 - 整数 バイナリ (符号付き) | 数値タグ |
| 5 - 単精度浮動小数型(IEEE) | 数値タグ |
| 6 - 倍精度浮動小数型(IEEE) | 数値タグ |
| 7 - アスキー | 文字列タグ |

ビットタグは TRUE か FALSE の値をとります。TRUE は ON の状態で、FALSE は OFF の状態です。数値タグは数値を取扱います。文字列タグは、文字（全角文字も含む）を取り扱います。

---



ビットデバイスをビットタグ（TRUE/FALSE）ではなく、数値タグ（1/0）として扱いたい場合は下記のように設定します。その他、M0000 から M0015 までを BCD 値として扱うなど、柔軟に定義できるようになっています。詳しくは本書中、「機能解説」の「デバイスの解釈」を参照下さい。




---

## 仮想デバイスについて

本パッケージは、現在 100 機種相当の通信に対応しており、その対応機種の 1 つとして、「仮想デバイス」があります。「仮想デバイス」は PLC をシミュレートしたアドレス空間であり、特定の PLC との通信を行わないという点を除き、標準的な PLC デバイスと同等に扱えます。本マニュアルでは、特定の PLC に限定して説明することを避けるため、「仮想デバイス」を用いて説明を進めます。「仮想デバイス」での扱いを理解すれば、その他の標準的な PLC でも同様の手順で設定が行えます。

仮想デバイス一覧

| デバイス | アドレス | 説明 | サイズ | 読み | 書き |
|---|---|---|---|---|---|
| 入力リレー | X0000～X9999 | 10 進数 | ビット | ○ | ○ |
| 出力リレー | Y0000～Y9999 | 10 進数 | ビット | ○ | ○ |
| 内部リレー | M0000～M9999 | 10 進数 | ビット | ○ | ○ |
| 特殊リレー | SM0000～SM9999 (※1) | 10 進数 | ビット | ○ | ○ |
| データレジスタ | D0000～D9999 | 10 進数 | ワード | ○ | ○ |
| 特殊レジスタ | SD0000～SD9999 (※2) | 10 進数 | ワード | ○ | ○ |

(※1)特殊リレー（SM）について

SM0000 10ms ごとに On/Off
SM0001 50ms ごとに On/Off。
SM0002 100ms ごとに On/Off
SM0003 500ms ごとに On/Off
SM0004 1000ms ごとに On/Off
SM0005 2000ms ごとに On/Off
SM0006 3000ms ごとに On/Off
SM0007 5000ms ごとに On/Off

(※2)特殊レジスタ（SD）について

SD0000 10ms ごとにインクリメント
SD0001 50ms ごとにインクリメント
SD0002 100ms ごとにインクリメント
SD0003 500ms ごとにインクリメント
SD0004 1000ms ごとにインクリメント
SD0005 2000ms ごとにインクリメント
SD0006 3000ms ごとにインクリメント
SD0007 5000ms ごとにインクリメント

SD0008～SD0015 は、それぞれ SD0000～SD0007 の値を用い、0～100 の間でサインカーブを描く。




仮想デバイスで作成されたタグは、内部メモリや共有メモリ的なタグとして色々と応用ができます。例えば、計算結果を一時的に格納するためのワーク的な利用や、OPC や DDE などの外部クライアント間での共有メモリとして利用することができます。

## スタティックタグとアクティブタグ

PLC と通信接続する際には、必ずしも全てのタグを事前に登録しておく必要はありません。タグを利用するアプリケーション側から直接 PLC アドレスを指定する事によって、タグの事前登録を行わずに動的に PLC との通信を確立する事が可能です。ここで、あらかじめ登録しておくタグの事を“スタティックタグ”と呼びます。それに対し、事前のタグ登録無しに画面などから PLC アドレスを直接指定する方法を“アクティブタグ”と呼びます。

スタティックタグのタグパスは、ユニット、フォルダ、タグをピリオド「.」でつなげます。

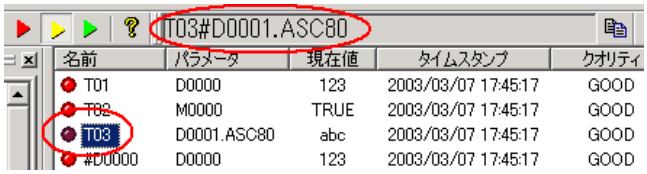
```
U01.F01.T01
U01.F01.T02
U01.F01.T03
：　：　：
```

アクティブタグのタグパスは、アドレス指定の前に必ず“#”を 1 文字付けます。

```
U01.F01.#D0          ※PLC の D0 アドレスを指定
U01.F01.#M0          ※PLC の M0 アドレスを指定
：　：　：
```

スタティックタグとアクティブタグは混在して使用することもできます。

タグの参照方法によっては、アクティブタグの後にパラメータ文字列を指定する必要があります。パラメータ文字列に何を指定するべきかについては、ツールバーの表示から知ることができます。例えば以下の「T03」のパラメータ文字列は「D0001.ASC80」です。よって、アクティブタグ「U01.F01.#D0001.ASC80」はスタティックタグ「U01.F01.T03」と同じ値になります。詳しくは本書中「パラメータ文字列」を参照ください。



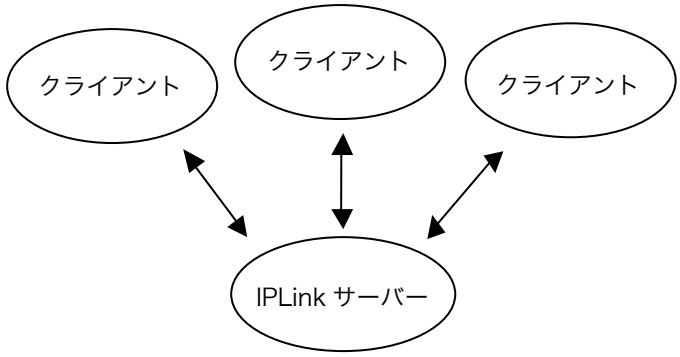

お薦め

スタティックタグの場合、あらかじめタグを設定しておくという手間が生じてしまいますが、通信対象のデバイスアドレスが変更されたり、PLC の機種が変わったりした場合などでも、タグの設定変更のみで変更対応が可能であるため、上位側のシステムの変更が生じないという利点があります。また、接続要求が無いときや PLC と接続されていない場合でもタグリスト上にタグの一覧を確認できるため、デバッグ作業が容易になります。スタティックタグ、アクティブタグのどちらを使うべきか迷う場合には、スタティックタグの使用を推奨いたします。

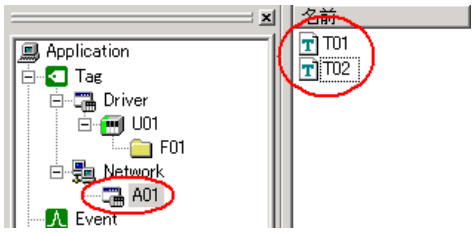
## ネットワークタグ

ネットワークタグを使用すると、TCP/IP のネットワーク環境下で、IPLink サーバー上のタグを共有できます。

ネットワークタグでタグを共有するためには、タグを保有するアプリケーションを IPLink サーバーとして起動しておく必要があります。IPLink サーバー上のタグは、ネットワークタグを介して複数のクライアントから共有することができます。




ネットワークタグの階層構造は２階層です。




■アプリケーション
どのアプリケーション（IPLink サーバー）と接続を行うのかを定義します。

■タグ
IPLink サーバー上のどのタグと接続したいのかを定義します。IPLink サーバー内に定義されているタグを、タグパスで指定します（「U01.F01.T01」や「U01.F01.T02」などを設定します）。

尚、ネットワークタグを参照する際のタグパスは、アプリケーション、タグを「/」（スラッシュ）でつなげた文字列になります（通常のタグパスの様に「ピリオド」ではない点に注意してください）。

例）A01/T01




IPLink サーバーとして起動するには IPLink インターフェースの定義を行う必要があります。

## 命名規約について

ユニット名、フォルダ名、タグ名を設定する際は、以下の規約に注意してください。

1. アルファベット（A～Z, a～z)、数字（0～9)、アンダーバー（_）が推奨文字列。
2. 「#./!」は使用できない。また、上記以外その他半角文字、または全角文字は利用する機能により制限がある場合があり非推奨。
3. 第１文字目に数字は設定出来ない。

# ２章　タグの作成手順と基本操作

## ２－１　タグの作成とモニタリング

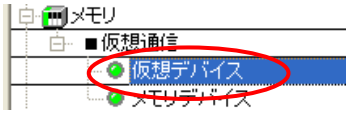
### 数値タグの作成

タグにはビットタグ、数値タグ、文字列タグの３つのタイプがあります。ここでは、数値タグを１つ作成します。

1．左側のツリーの「Driver」を右クリックし「ユニット追加」を選択




2．「メモリ」の「仮想通信」「仮想デバイス」を追加する




表示された一覧から、「メモリ」の「仮想通信」「仮想デバイス」を選択し、「OK」を押します。尚、この例では「仮想デバイス」を選択しましたが、通常は実際に接続する PLC を選択してください。

3．「U01」を右クリックし「フォルダ追加」を選択




この操作で、フォルダ「F01」が追加されます。

4．「F01」を右クリックし「タグ追加」を選択

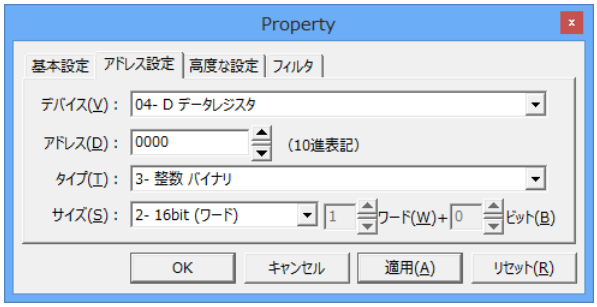



この操作で、「T01」というタグが 1 点追加されます。

5．左側のリストから「T01」を右クリックし、「プロパティ」を選択




6．「アドレス設定」タブから「04 - D データレジスタ」を選択し、「OK」を押す

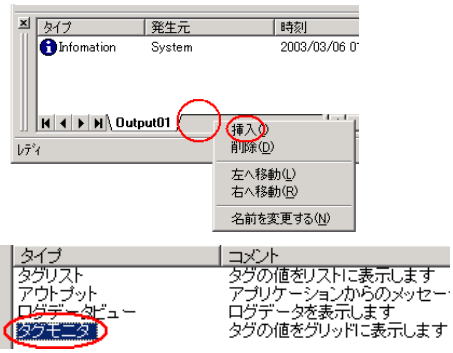



この例では、D0000（ワードレジスタ）を数値タグ（整数バイナリ）として設定しています。

## 現在値を画面に表示する

続いて、さきほど作成した数値タグの値を画面に表示してみます。

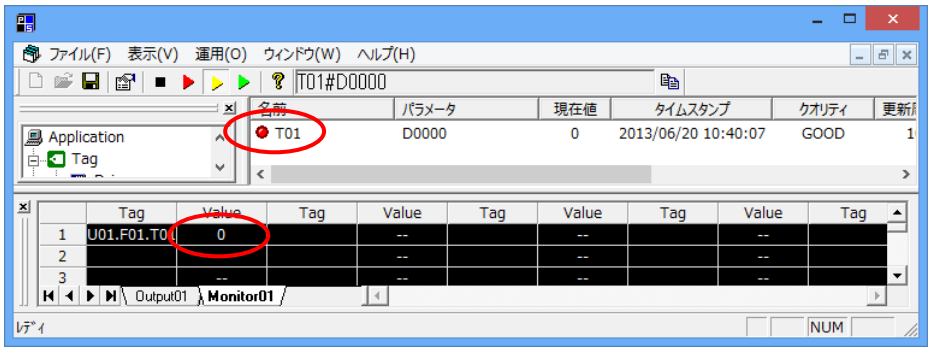
1．ビューにタグモニタを追加する
下のアウトプットビューの下記位置で右クリックし、「挿入」を選択し、「タグモニタ」を追加する



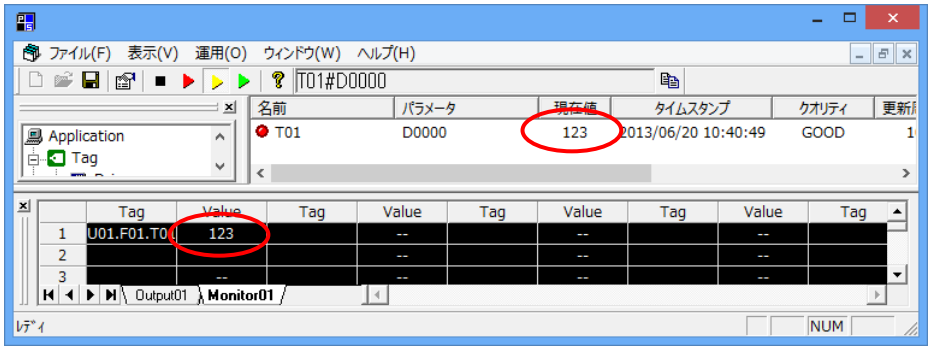

2．オンライン（モニタ）にする



3．セルに「U01.F01.T01」と入力する
T01 のランプが赤くなり、セルに値が表示されたら設定完了です。




Value の列に最初は「0」が表示されています。ここで適当な数値を書き込んでみてください。すると、「U01.F01.T01」へ値が書き込まれます（一覧の現在値が変化します）。



## ビットタグと文字列タグの作成

続いて、ビットタグと文字列タグを作成します。

1. 「F01」を右クリックし「タグ追加」を２回選択し２つのタグを追加する
   「T02」「T03」が追加されます。



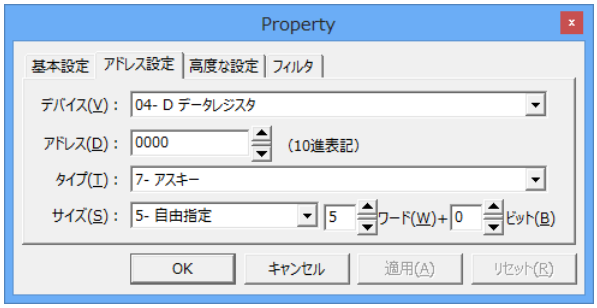

2. 「T02」「T03」の「プロパティ」を下記のように設定する

   (T02)




   M0000（ビットレジスタ）をビットタグとして設定しています。

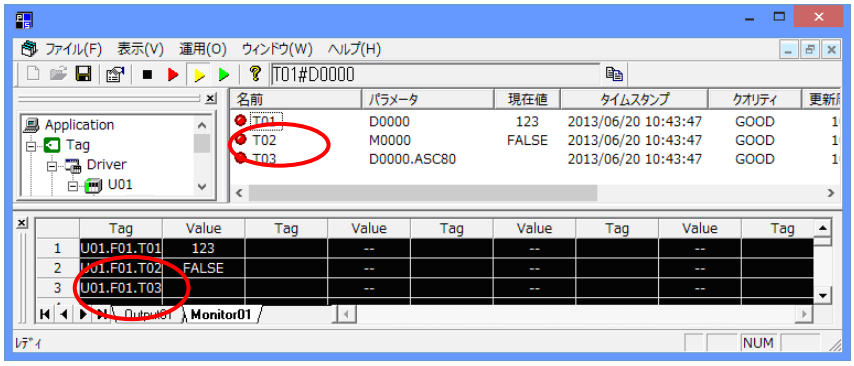
   (T03)




   D0001 から D0005（アスキー文字１０文字分）を文字列タグとして設定しています。

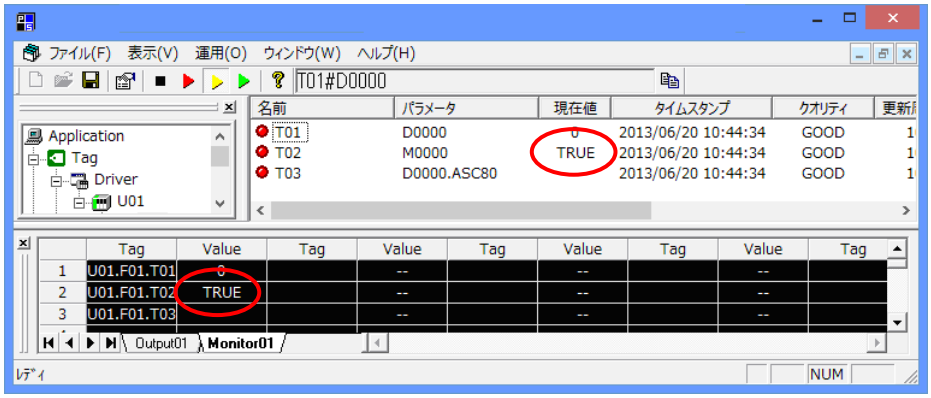
   続いて、この２点のタグをタグモニタに追加します。

3. オンライン後、タグモニタのセルに「U01.F01.T02」及び「U01.F01.T03」を追加する



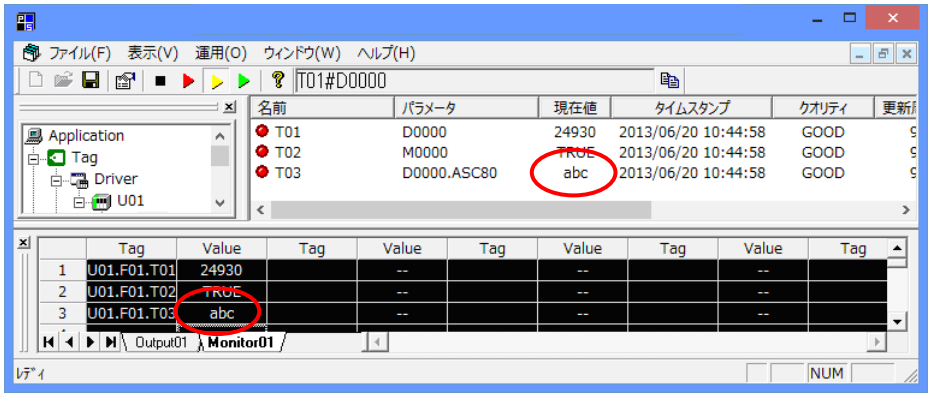
4．タグモニタからビットの値を編集する

ビットタグ「U01.F01.T02」の Value を、「TRUE」または「FALSE」と変更してみてください。タグモニタから変更した内容が一覧に反映されます。




5．タグモニタから文字列の値を編集する

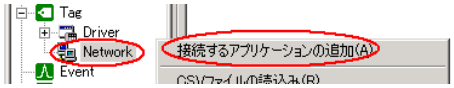
同様に、文字列タグ「U01.F01.T03」の Value に、「abc」などの任意の文字列を入力してみて下さい。タグモニタから変更した内容が一覧に反映されることを確認してください。



## ネットワークタグの作成

ここでは、ネットワークタグの作成手順について解説します。

1．Network にアプリケーションを追加する
左側のツリーから「Network」を右クリックし「接続するアプリケーションの追加」を選択します。

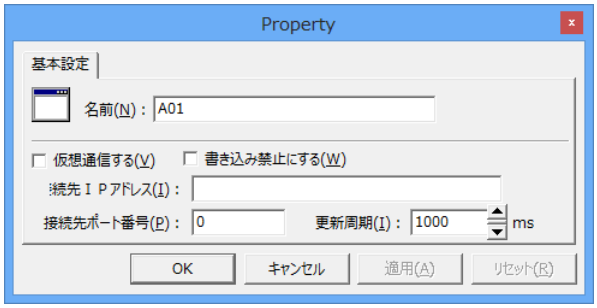



「A01」が追加されます。

2．「A01」を右クリックし「プロパティ」を選択

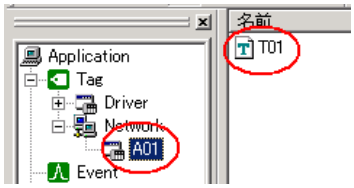



3．接続先の IPLink サーバーの設定を行う



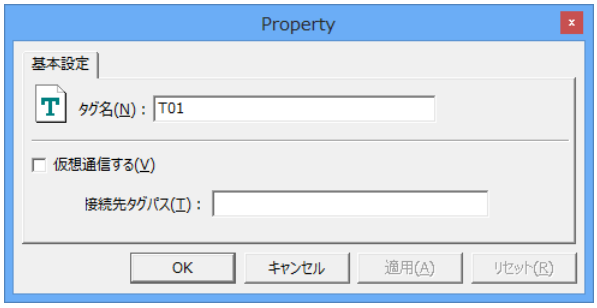

・接続先 IP アドレス　：　接続先 IPLink サーバーの IP アドレス
・接続先ポート番号　：　接続先 IPLink サーバーのポート番号
・更新周期　　：　IPLink サーバーから値を取得・更新する周期

4．「A01」を右クリックし「接続するタグの追加」を選択




タグが追加されます。

5．「T01」を右クリックし「プロパティ」を選択し、接続先のタグパスを指定する




・接続先タグ名に、接続先 IPLink サーバー内のタグパス（U01.F01.T01 など）を設定




テスト的に IPLink の動作確認を行いたい時に他に起動させるアプリケーションが無い場合には、自身のアプリケーションを IPLink サーバーとして設定し、自分自身のタグをネットワークタグとして登録して接続してみて下さい。

# ２－２　タグを一括で作成する方法

## タグの連続追加

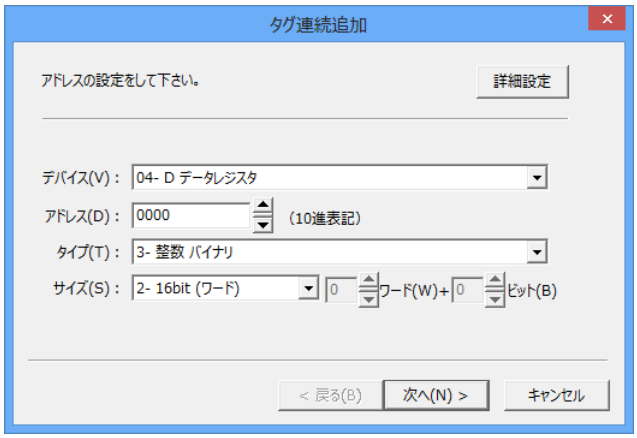
「タグの連続追加」機能を用いると、大量のスタティックタグを簡単に作成することができます。

1.　「F01」を右クリックし「タグ連続追加」を選択する

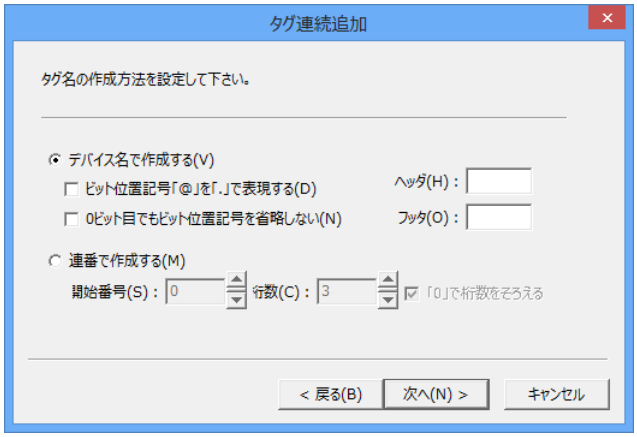



2.　タグ連続追加ウィザードに従って設定を進める

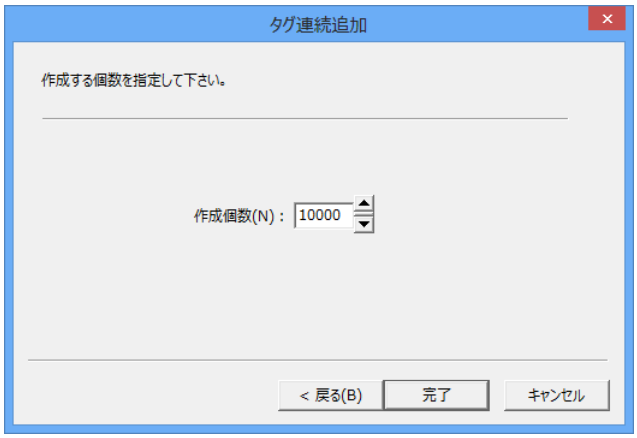
まず、連続追加を行う先頭のアドレスを設定します。詳細設定ボタンを押すと、フィルタの定義ができます。




続いて、タグ名の作成方法を定義します。「デバイス名で作成する」を選択すると、「D0000」などのアドレスがそのままタグ名になります。「連続で作成する」を選択すると、連番タグ（例えば T001 から T999 など）で作成できます。



タグの作成個数を設定して完了を押します。




3． 自動生成されたタグの設定を確認する

ダイアログから指定した内容に従って、タグが一括で登録されました。下記は、「デバイス名で作成する」で D0000 から D9999 までの 10000 個のタグを作成した例です。正しくタグが作成されていることを確認してください。





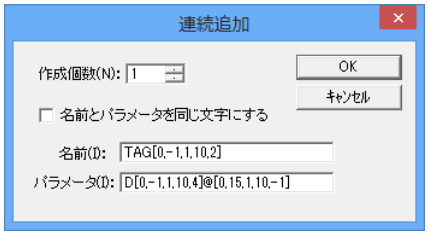

登録可能なスタティックタグの数には特に制約はありませんが、１つのフォルダに１万点を目安としてください。それ以上のタグを設定する場合には複数のフォルダに分けて設定することを推奨します。尚、スタティックタグは通信要求が発生したタグに対してのみ PLC との通信を行いますので、多くのタグを定義していても通信の負荷は上がりません。



連続追加に失敗したタグは手動で削除して下さい。一度にたくさんのタグを選択したい場合には、「Ctrl」と「A」キーを同時に押すことによって全てのタグを選択する事ができます。この状態で「Delete」キーを押すと、全てのタグを一括で簡単に削除することができます。



OPC、DDE、IPLink など一部のユニットでタグの連続追加を行う場合、パラメータ設定ダイアログが表示され、ダイアログで設定したパラメータに沿ってタグが追加されるものがあります。



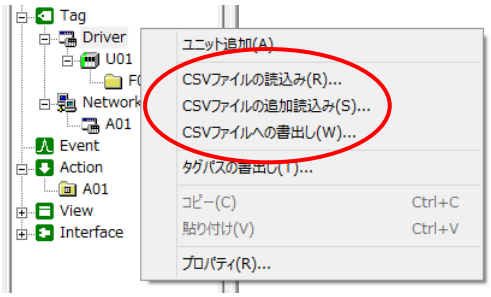

パラメータの詳細は以下の通りです。

**名称 [ 開始アドレス, -1(固定値), 増加値, 進数, 桁数 ]**

## CSV ファイルによるタグの編集

タグの設定は CSV ファイルへ書き出すことができます。CSV ファイルは Excel などで編集し、再び読み込むことができます。大量のタグを管理する場合に便利です。

CSV ファイルへの書き出しと読み込みは下記のメニューから行うことができます（ファイル名の拡張子を「.csv」としておくと Excel で直接開けます）。




書き出された CSV ファイルには、タグの情報が一行ずつ保存されています。１行ごとに１つのタグの情報が含まれており、１行は３列で構成されています。フォーマットは以下のとおりです。

```
“ユニット名#パラメータ文字列” , ” フォルダ名#パラメータ文字列” , ” タグ名#パラメータ文字列”
```

タグが含まれるユニットやフォルダが、１つ上のタグと同一だった場合、そのユニットやフォルダを、「*」で置き換えることができます。尚、パラメータ文字列に関する詳細ついては、本書中「パラメータ文字列」を参照してください。

```
“ユニット 1” , ”フォルダ 1” , ”タグ 1”
“*” , ” *” , ”タグ 2”
“*” , ” *” , ”タグ 3”
“ユニット 2” , ”フォルダ 1” , ”タグ 1”
“*” , ” *” , ”タグ 2”
...
```

（例）三菱 A シリーズ RS-232C




```
"U01#MIT.A.R:1,19200,7,1,E,0,10,500,10,500,0,Y,Y,Y,1" , "F01#FF,00" , "T01#D0000"
"*" , "*" , "T02#D0001"
"*" , "*" , "T03#D0002"
```

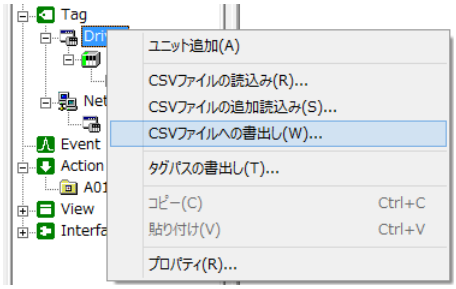


CSV ファイルとはテキストファイルの一種で、収められているデータが「,」（カンマ）と改行によって区切られているファイルです。単純なファイル形式のため、多くのアプリケーション、特に表計算ソフト（Excel など）でサポートされています。また、基本的に単なるテキストファイルですので、メモ帳などのテキストエディタでも開くことができます。

CSV ファイルによるタグの編集に際しては、以下の手順を参考にしてください。

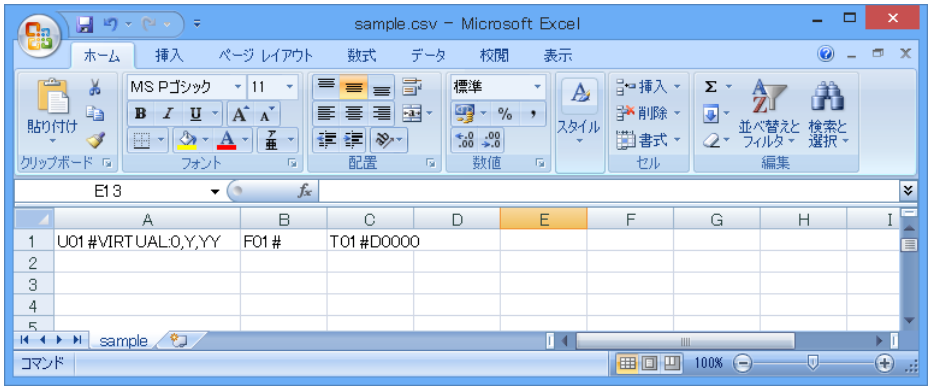
※既に数点のタグが登録されていること前提に解説します。

1． 左側のツリーの「Driver」を右クリックし「CSV ファイルへの書き出し」を選択




2． 適当なファイル名を指定して OK を押す
ファイル名を「.csv」としておくと Excel で直接開けるので便利です。

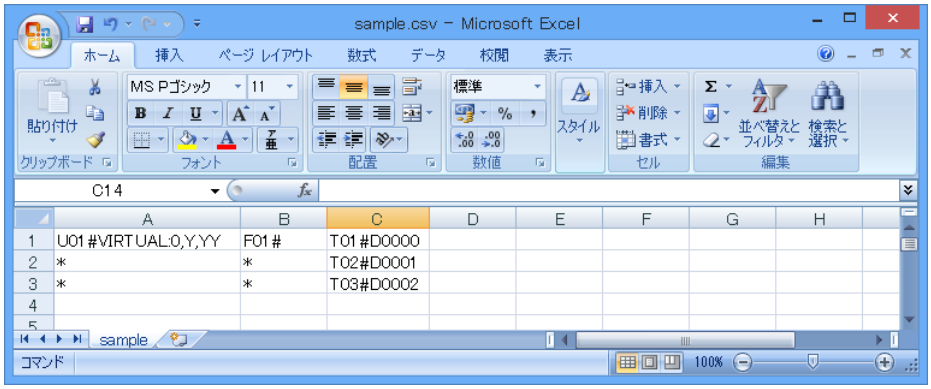
3． Excel でファイルを開く




この例では、既に数値タグが 1 点登録されている状態です。例として 2 点の数値タグを追加します。

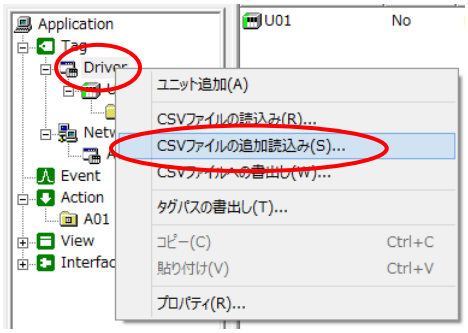
4． 下記の 2 行を追加し、ファイルに保存する

| | | |
|---|---|---|
| * | * | T02#D0001 |
| * | * | T03#D0002 |



5. 編集した CSV ファイルを読み込む

左側のツリーの「Driver」を右クリックし「CSV ファイルの読込み」を選択します。



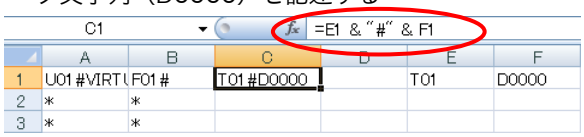

6. 「T02」「T03」が追加されていることを確認する



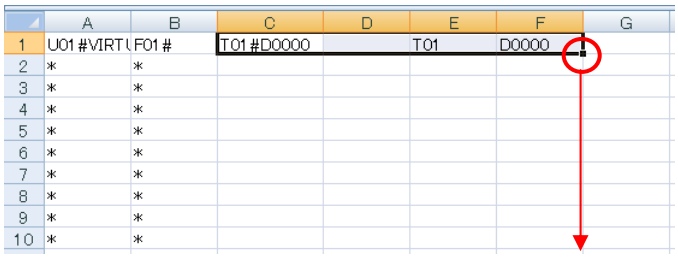
Excel には、自動的に数値をインクリメントして、行をコピーする機能があります。この機能の利用により簡単に大量のタグを編集することができます。

例）

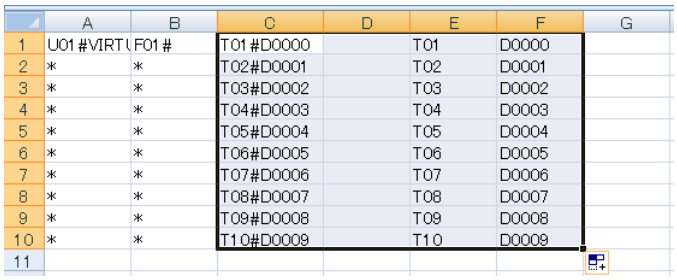
1．C1 のセルに「=E1 & “#” & F1」という式を書く。また E1、F1 にそれぞれタグ名（T01）とパラメータ文字列（D0000）を記述する




2．E1、F1 にそれぞれタグ名（T01）とパラメータ文字列（D0000）を記述する



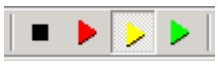

すると下記のようにコピーされますので、A 列から C 列までを他のシートにコピーし、CSV ファイルへ保存して下さい。



# 3章　機能解説

## オンライン状態の切り替え

アプリケーションをオンラインに切り替えることにより、タグは通信可能な状態になります。
オンラインには３つの状態があります。



赤：オンライン（デバック）　仮想的に通信します。PLC が手元に無いときに使用します。
黄：オンライン（モニタ）　アプリケーションで現在値の確認や設定が行えます。
緑：オンライン（最速）　アプリケーションは最小化され、タスクバーへアイコンとして表示されます。

タグの通信状況に応じて、タグの左側に表示される丸いランプの状態が変化します。アプリケーションがオンラインになると、まず PLC との接続処理が行われます。但し、タグに対する通信要求がない限り、ランプは黒色のままで待機している状態です。タグに対する通信要求が発生するとランプが黒以外の色に変化します。

ここでいう要求とは、色々なケースがあります。
・クライアントからの通信要求（OPC/DDE/IPLink など）
・各アクションからタグが使用されている場合
・タグモニタビューなどから参照されている場合

各タグが受け取っている要求の種別は、ランプの色の状態によって判別できます。

・（緑）
内部から「自動更新あり」でタグに接続されています。ここで言う内部とは、「タグ操作」「モニタリング開始」により、モニタリングの要求をした場合です。「モニタリング終了」により、再び黒色に戻ります。

・（黄）
外部から「自動更新なし」でタグに接続されています。

・（赤）
外部から「自動更新あり」でタグに接続されています。

「自動更新あり」の場合はフォルダで設定した更新周期で PLC と通信を行います。この自動更新（ポーリング）はフォルダ単位で通信が行われ、「自動更新あり」のタグのみ通信対象となります。例えばフォルダの中に１万点のタグが定義されていたとしても、１点しか「自動更新あり」のタグがない場合は、その１点のみ通信を行います。

| アイコン | 色 | 通信要求 | 要求元 | 自動更新 |
|---|---|---|---|---|
| | 黒 | 無し | ー | ー |
| | 緑 | 有り | 内部※ | する |
| | 黄 | 有り | 外部※ | しない |
| | 赤 | 有り | 外部※ | する |

※「内部」とは、「タグ操作」「モニタリング開始」により、モニタリングの要求を行った場合です。それ以外は外部からの要求になります。

<OPC>
OPC 接続の場合、クライアントからアクティブタグとして接続されるとランプが赤色になり、アクティブでないタグとして接続された場合は黄色になります。

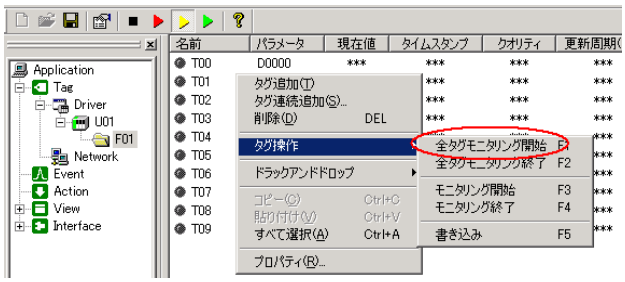
<アクション>
例えばロガーアクションの場合、タグフィールドの設定で「キャッシュ値でログをとる」のチェックを外しておくと、キャッシュ値を用意する必要がないため、ランプが黄色になり自動更新が行われません。

## 値のモニタ

オンラインが開始している状態においても、タグへの通信要求が発生していない時はランプの表示は黒のままです（通信を開始していない、待機状態）。この状態で PLC の値を確認したい場合には、下記の操作を行うことにより強制的に通信を開始し、値を確認することができます。

1．オンライン（モニタ）にする

2．タグリスト上で右クリックし、メニュー「タグ操作」「全タグモニタリング開始」を選択する




この操作により通信が開始され、ランプが緑色になったら通信が行われている状態です。画面から現在値を確認することができます。

## 値の書き込み

画面から強制的に PLC の値を変更したい場合は次の手順で書き込みを行えます。

1．タグのランプが赤もしくは緑色になっていることを確認する
それ以外の色の時は、上記手順で値のモニタを開始して下さい。

2．書き込みを行いたいタグ上で右クリックし、メニュー「タグ操作」「書き込み」を選択する




書き込みダイアログが表示されます。

3．書き込みを実行する

配列タグへ値を書き込みたい場合は「11,22,33,44,55」などのように値を「,」カンマで区切ります。また、値の入力後に他のタグに対して書き込みを再度実行したい場合は、書き込みダイアログを表示したままで対象のタグを選択して書き込みを実行して下さい。




また、下記のように複数のタグを選択して書き込みを実行すると、複数のタグに対して同じ値を一度に書き込むことができます。



## 更新周期と高速化についてのヒント

更新周期の設定はフォルダで行います。更新周期にすべて 0 を指定すると出来る限り早く値を更新します。

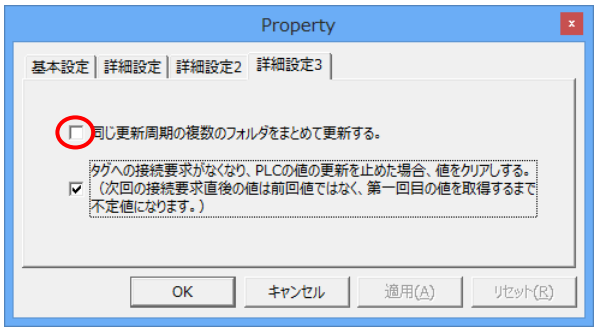



「指定した周期で値を更新する」のチェックをはずすと、自動更新の要求があっても、更新のための通信を行いません。また、ユニットのシステムプロパティ「U01!Update」で自動更新を行うかどうかを外部から切り替えることができます（詳しくは本書中「システムプロパティ」の章を参照下さい）。

---




更新のための通信はフォルダ毎に行われます。あまり、多くのフォルダに分割してしまうと本来 1 回の通信で行える通信があったとしても、強制的に通信が分散されてしまうため、通信が遅くなってしまいます。例えば、フォルダが 2 個あり、それぞれに D0 と D1 のタグがあった場合、従来 D0 と D1 を 1 回のパケットで通信ができるところを、フォルダが分かれているため、D0 と D1 との通信が 2 回のパケットに分かれてしまいます。そういった場合、ユニットの下記のプロパティ「同じ更新周期の複数のフォルダをまとめて更新する」にチェックを入れると、複数のフォルダ内に定義した複数のタグを、自動的に一番最適な通信方法（最小通信パケット数になる）を自動的に算出し通信するようになります。



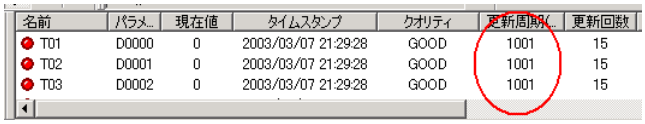

例えば、全てのフォルダを同じ更新周期（例えば 1 秒）にすると、1 秒ごとに全てのタグに対して通信が行われます。これは、値の更新という意味では最も早い方法になりますが、反面、値を更新している最中、書き込みなどの他の通信要求が待たされるというデメリットもあります。まずは、すべてのフォルダを同じ更新周期にしてもらい、書き込みをしてみて問題がなければそのままで OK です。もし、書き込みが待たされるような感覚がありましたら、1000ms の更新周期のものと 1001ms の更新周期のフォルダを半々づつに定義すると、書き込み要求が入り込めるタイミングが増えるため、待たされる時間がへります。また、1000ms と 1001ms などのフォルダでグループ分けする場合、できるだけ同じ種類のデバイス（D、M など）を同じ更新周期でまとまるように、グループわけするとより効率よく通信されます。

---

## 更新周期の計測手順

更新周期はフォルダごとに設定することができます。しかしながら、タグの点数が多くなると通信の負荷が高くなり、実際に指定された周期では通信が行えない場合があります。また、更新周期の設定に「0」を指定して最速とした場合など、実際にはどの程度のパフォーマンスで通信が行われているのかを確認したい場合があります。

簡易的には、タグリストの「更新周期（ms)」から知ることができます。

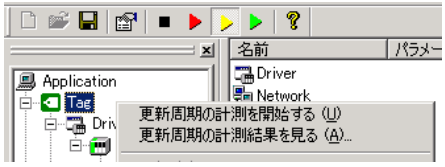


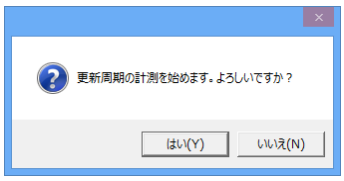

正確に計測を行いたい場合は、下記の手順でパフォーマンスを計測することができます。

1．オンライン後、実際の通信を開始する
タグに実際の通信要求をかけてください。また、最速の通信周期を計測したい場合はフォルダの更新周期の設定を「0」にして下さい。

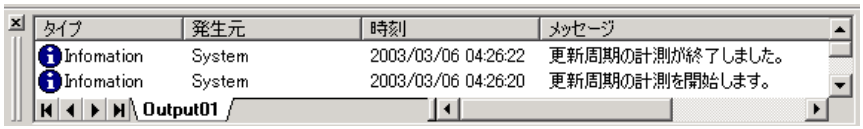



2．左ツリーから「Tag」を右クリックし、「更新周期の計測を開始する」を選択する




3．アウトプットビューで「更新周期の計測が終了しました」というメッセージがでるまで待つ




4．左ツリーから「Tag」を右クリックし、「更新周期の計測結果を見る」を選択する

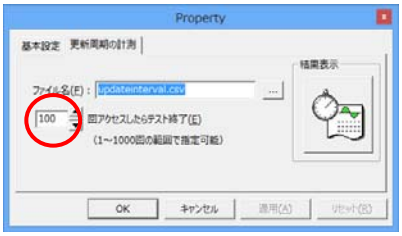



これで、更新周期の計測結果を見ることができます。

---

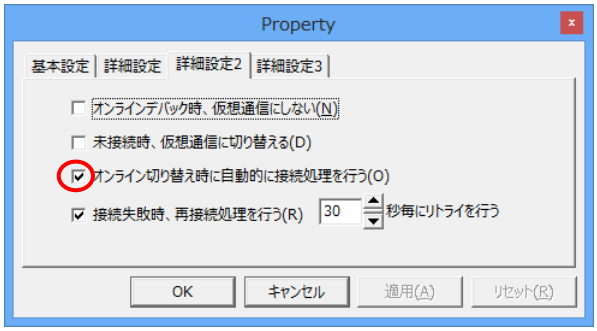
ヒント

更新周期の計測の回数はデフォルトで１００回になっています。この回数を変更したい場合は「Tag」のプロパティの「更新周期の計測」タブで回数を設定してください。

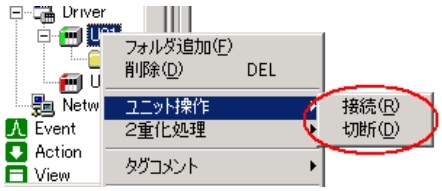


## 接続・切断について

PLC との接続はオフラインからオンラインに変更した時に行われます。アプリケーションから任意のタイミングで接続を行いたい場合は、「オンライン切り替え時に自動的に接続処理を行う」のチェックを外してください。また、ユニットのシステムプロパティ「U01!Connect」で外部から接続・切断の切り替えが行えます。詳しくは本書中「システムプロパティ」の章を参照下さい。

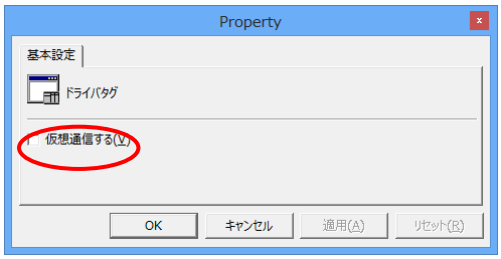


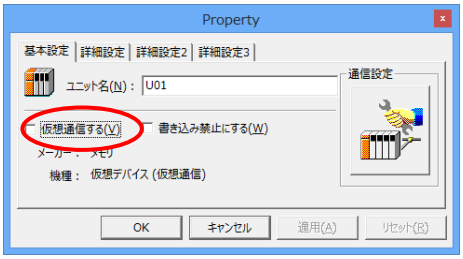

手動での接続の切り替えは各ユニットのメニューから行えます。

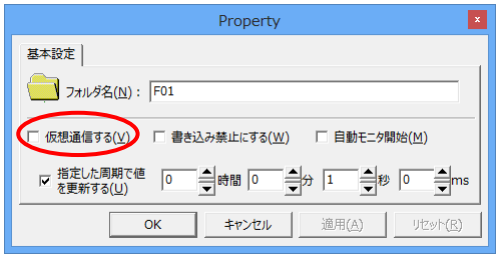


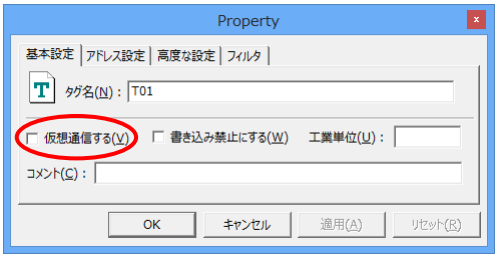
## 仮想通信について

仮想通信にすると、実際に PLC との通信を行わず、常に内部キャシュ上の値の読み書きになります。PLC が手元にない環境でのデバッグなどに使用することができます。仮想通信は（赤）：オンラインデバックにすることにより、すべての PLC が仮想通信になります。全体ではなく、細かいレベルで仮想通信を切り替えたい場合には、ルート、ユニット、フォルダ、タグ毎にも設定することができます。また、システムプロパティ「Virtual」で外部から仮想通信のモードを切り替えることができます。詳しくは本書中「システムプロパティ」の章を参照下さい。




---

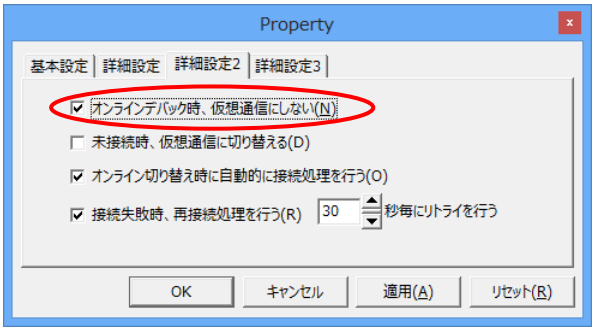



オンラインデバッグを使用する場合は注意点があります。

- 仮想通信の場合、タグや PLC の設定に依存せずに、自由な値が書き込めるようになります。例えば、タグのタイプが整数にもかかわらず、少数を含む値を書き込めたり、PLC 上のアドレスの範囲を超えるような値を書き込めたりするようになります。

- タグフィルタを設定していてもフィルタ機能が無効化されます。

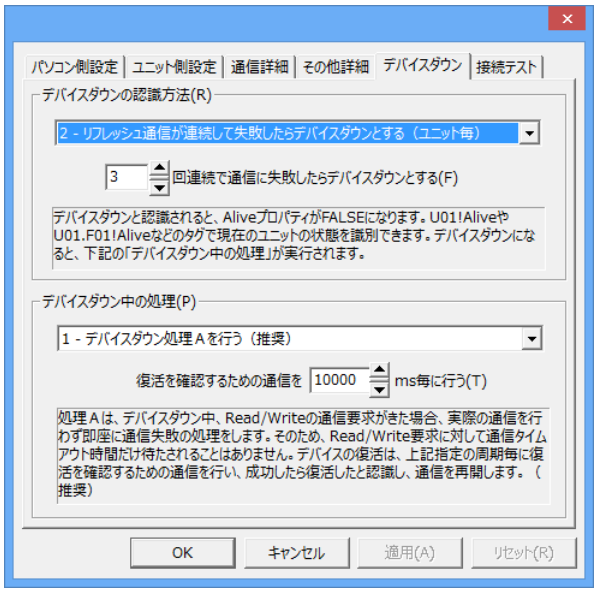
この注意点は、仮想デバイスに対しても有効です。オンラインデバッグ時に、仮想デバイスに設定したタグフィルタを有効にしたい場合は、ユニットのプロパティを開き、詳細設定 2 の「オンラインデバッグ時、仮想通信にしない」にチェックをいれることで、オンラインデバッグ時であっても、仮想通信が行われないようになり、タグフィルタが使用できるようになります。




---

## デバイスダウンについて

デバイスダウンとは PLC の生存状態を認識するための設定です。PLC との通信が連続して失敗することにより、デバイスダウンと認識されます。デバイスダウン中はユニットのシステムプロパティ「U01!Alive」が FALSE になります。デバイスダウン状態になると、「デバイスダウン中の処理」に従い、PLC が復活するまでデバイスダウン処理を行います。




---

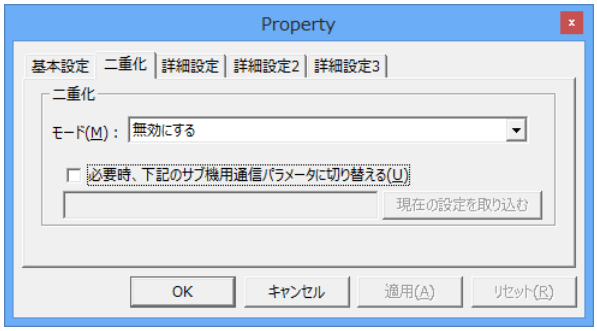
デバイスダウンの認識方法には下記の 2 種類あります。

・リフレッシュ通信が連続して失敗したらデバイスダウンとする（フォルダ毎）
・リフレッシュ通信が連続して失敗したらデバイスダウンとする（ユニット毎）

デフォルトでは「ユニット毎」が選択されています。フォルダ毎のネットワーク設定が異なり、それぞれのフォルダ毎にデバイスダウンを認識したい場合は、「フォルダ毎」を選択します。例えば、RS422 によるマルチドロップ接続時など、フォルダごとに生存状態が異なる場合がありますが、そのとき「フォルダ毎」を選択すると、フォルダ毎にデバイスダウンが認識され、フォルダのシステムプロパティ「U01.F01!Alive」で個別の生存状態を識別できます。「フォルダ毎」を選択するときの注意点としては、フォルダ毎に連続通信エラーをカウントするため、デバイスダウン認識までの時間が遅くなります。

---

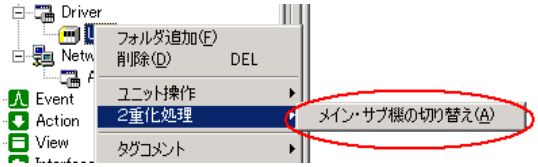
## ２重化について




２重化モードとは、メイン側、サブ側（待機側）の２台の PLC が接続されている環境において、メイン側に障害が発生した場合にサブ側のＰＬＣに接続先を切り替えるための設定です。この設定はユニットごとに行えます。モードには以下を選択できます。

■無効にする
切り替えは行いません。

■手動で行う
下記のメニューから手動で切り替えることができます。




■自動で行う（切り替え後はサブで通信しつづけます）
メイン側の PLC との通信が連続して失敗することによってデバイスダウンと認識された場合、通信対象をサブ側ＰＬＣに切り替えます。このモードでは、切り替え後にサブ側でも障害が発生した場合に、メイン側への復帰を行いません。尚、自動モード中であっても手動で切り替えることもできます。

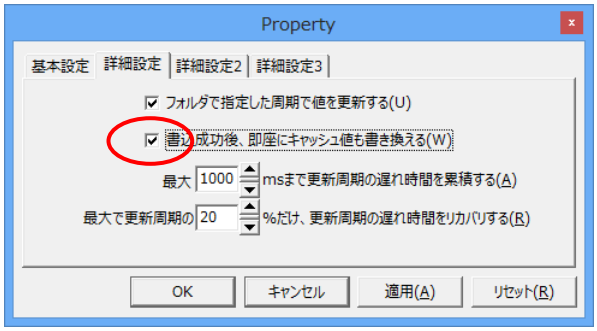
■自動で行う（サブで再ダウンした場合、メインへ戻します）
メイン側の PLC との通信が連続して失敗することによってデバイスダウンと認識された場合、通信対象をサブ側ＰＬＣに切り替えます。このモードでは、切り替え後にサブ側でも障害が発生した場合に、メイン機に再び切り替えます。つまり、障害が発生するたびにメインとサブを交互に切り替えます。尚、自動モード中であっても手動で切り替えることもできます。

また、サブとメイン機の切り替えはシステムプロパティ「UseSub」で切り替えることができます。詳しくは本書中「システムプロパティ」の章を参照下さい。

## ２重化における通信パラメータの切り替えについて

システムプロパティ「DriverMain」「DriverSub」により、ドライバのパラメータを外部からダイナミックに切り替えることができます。もし、２重化を使用しない場合は「DriverMain」のみ使用します。詳しくは本書中「システムプロパティ」の章を参照下さい。

## 書き込み後のキャッシュ更新について




PLC に対して書き込みが成功した場合、即座に内部キャッシュの値を書き換えるかどうかを設定できます。

■チェックされている場合（デフォルト）
＜OPC＞
書き込み成功後、即座に DataChange イベントが発生し、値の変化通知が上位に通知されます。また、キャッシュ値の読み込みでも、書き込み後の値が渡されます。

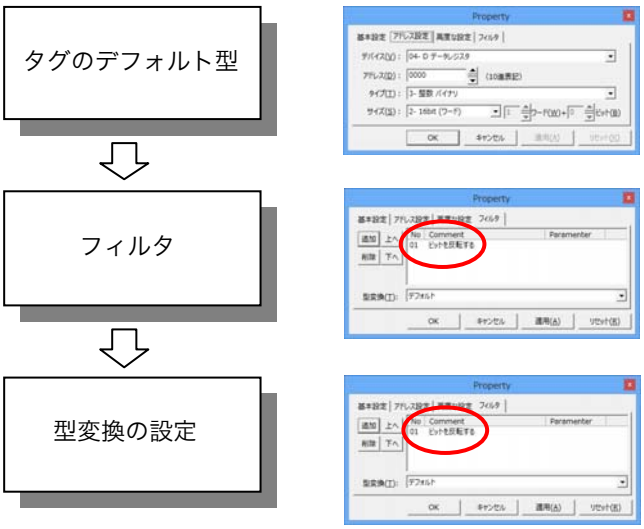
＜IPLink クライアント ActiveX＞
書き込み成功後、ValueChange 等のイベントが発生します。また、ReadVal(DataSource = 1)での値の取得でも、書き込み後の値が渡されます。

■チェックされていない場合
＜OPC＞
自動更新、もしくは Read 要求によりキャッシュがリフレッシュされるまで、DataChange イベントは発生しません。また、キャッシュ値を読み込んだ場合も、キャッシュがリフレッシュされるまでは以前の値が返されます。

＜IPLink クライアント ActiveX＞
次の PLC との読込通信が行われるまで、ValueChange 等のイベントは発生しません。ReadVal(DataSource = 1)での値も、次の PLC との読込通信によりキャッシュがリフレッシュされるまでは以前の値が返されます。

## データ型と型変換について

タグはビットタグ、数値タグ、文字列タグ、及びそれらの配列があります。最終的なデータ型は「型変換の設定」で決定されます。




■タグのデフォルト型

型変換の設定を「デフォルト型」にすると、標準 PLC では下記のように「アドレス設定」タブの「タイプ」により最終的なタグの型が決定します。また、「高度な設定」タブの「配列にする」をチェックすると、タグを配列にすることができます。

| タイプ | デフォルト型 | タグを配列にした場合 |
|---|---|---|
| ビット | ビット型 | ビット型の配列 |
| 整数 BCD | 数値型 | 数値型の配列 |
| 整数 BCD (符号付き) | 数値型 | 数値型の配列 |
| 整数 バイナリ | 数値型 | 数値型の配列 |
| 整数 バイナリ (符号付き) | 数値型 | 数値型の配列 |
| 単精度浮動小数型(IEEE) | 数値型 | 数値型の配列 |
| 倍精度浮動小数型(IEEE) | 数値型 | 数値型の配列 |
| アスキー | 文字列型 | 文字列型の配列 |

※「タグを配列にした場合」というのは、「配列にする」にチェックが入っている場合になります。

■フィルタ

タグのデフォルト型は、フィルタの指定によって変換される場合があります。たとえば、デフォルト型が数値のタグに対し、「ビット反転」フィルタをかけると、数値型のタグはビット型に変換（0 のとき FALSE、それ以外のとき TRUE）されます。フィルタ機能の詳細は、本書中「フィルタ」の記述を参照してください。

■型変換の設定

通常はデフォルトのままにします。何らかの理由で、データ型を変更したい場合のみ型変換を指定します。特に、フィルタ機能で「テーブルファイルを参照して数値や文字列に変換する」を使用してコメントなどを扱いたいときには、文字列型にしてください。



＜OPC＞
OPC ではアイテム（タグ）毎に、リクエストタイプ（データ型のタイプ）をクライアントアプリケーション（上位側）から指示されます。リクエストタイプに特に指定がない場合（デフォルト）であった場合は、タグの最終的な型で値が渡されます（ビット→VT_BOOL/ 数値→VT_R8/ 文字列→VT_BSTR）。リクエストタイプで型の指示があった場合は、リクエストに従って適切に変換を行い上位側に値を渡します。



＜InTouch＞
InTouch で 7 桁以上の整数を扱う場合は、必ず「InTouch で 7 桁以上の整数を使用する」のフィルタを設定してください。



＜アクション＞
各アクションでは、タグの最終型がどのような型であっても対応できるように内部的な処理が行われています。例えば、本来数値を要求する場面でタグの型が文字列だった場合、文字列を数値として解釈する様な内部処理が行われます。

## 配列について

標準 PLC では「高度な設定」タブの「配列にする」をチェックすると、タグが配列として扱われます。配列タグとして設定を行うと、連続したエリアをまとめて扱うことができます。例えば、PLC のアドレス「D00001」の数値タグで、配列数を 1000 と指定した場合、「D00001」から「D01000」までの 1000 個の配列として扱えます。



＜OPC＞
OPC の配列型（VT_ARRAY）に対応しています。接続する OPC クライアントが配列のデータ型に対応しているか否かを確認し、対応しているものであれば、連続したエリアに対して 1 つのタグで一括して通信を行うことができます。



＜DDE＞
Excel で VBA を用いて通信する場合、最大 32 個の配列として扱えます。詳しくは「DDE インターフェース詳細マニュアル」を参照ください。



＜IPLink クライアント ActiveX＞
IPLink クライアントでは ReadVal/ WriteVal などで配列を扱うことができます。詳しくは「IPLink インターフェース詳細マニュアル」を参照下さい。



＜FA-Panel＞
GetVal/ ReadVal/ WriteVal/ ReadRefresh/ WriteRefresh などのメソッドで配列を扱うことができます。配列を使用すると、スクリプトから大量のデータを高速に読み書きできます。詳しくは「ユーザーズマニュアル」を参照ください。

## デバイスの解釈

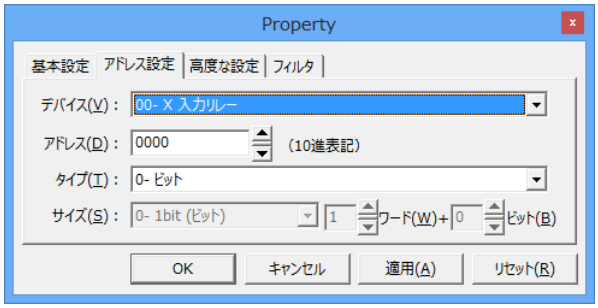
標準 PLC のデバイスには、ビットデバイス・バイトデバイス・ワードデバイスなど、いくつかの種類があります。例えば三菱電機製の PLC では「X」「Y」「M」などはビットデバイスであり、1 デバイスのサイズは 1 ビットです。また「D」「W」「R」はワードデバイスであり、1 デバイスのサイズは 16 ビットとなります。つまり「X」などのビットデバイス 16 個分が、「D」などのワードデバイス 1 つのサイズに相当します。

さて、三菱電機製の PLC では「X」は入力にあたります。入力カードの接点につながった機器が On の場合、値は 1 になり、Off の場合は 0 になります。通常はこれら 1 つ 1 つにそれぞれ別々の機器がつながっているので、ユーザーが知りたい情報は On か Off かということになります。
ところが、この入力カードにサムロータリースイッチなどの BCD 入力がつながっていた場合はどうなるでしょう。通常、BCD は 4 つの接点を使用して、4 ビットで 0 から 9 までを表現します。つまり、X0 から X3 までの情報で、1 桁の数値を示すことになります。当然 4 桁の数値をあらわすためには、4×4=16 ビット必要になります。こういった場合、X0 から X15 までのまとまった単位で初めて 1 つの情報として意味を持ち、それぞれをばらばらに解釈しても意味がありません。

また、「D」などのワードデバイスは通常 16 ビット単位で 1 つの数値をあらわしています。しかしながら、ワードデバイスを構成する各ビットにさまざまな情報のフラグを格納する場合などは、ビット単位で扱わなければならない場合もあります。

本パッケージはこれらのデバイスの種類に関わらず、さまざまな解釈ができるように柔軟に設計されています。また、それらの考え方は非常にシンプルな形にまとめられています。

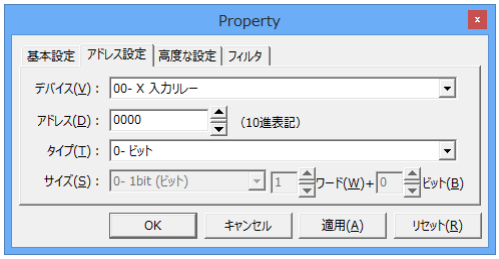
まずは、適当なタグを追加して、タグのプロパティを開いてみてください。



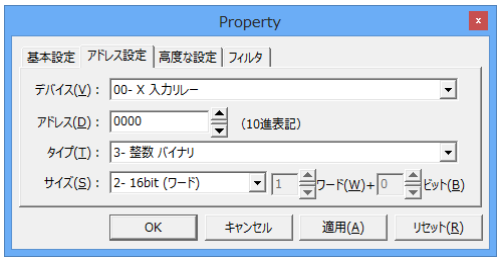

ここで、「タイプ」のコンボボックスを開いてみると、ビット・整数 BCD・整数バイナリ・アスキーなどが選択できることがわかります。ここで「整数バイナリ」を選択すると、サイズには、1 ビット・8 ビット・16 ビット・32 ビット・自由指定などが選択できることがわかります。さらに「自由指定」を選択した場合には、3 ビットや 12 ビットなど自由にビット数を指定できることもわかります。

デバイスをどのように解釈するかは、これらの設定を組み合わせることによって行います。以下は、設定の例です。

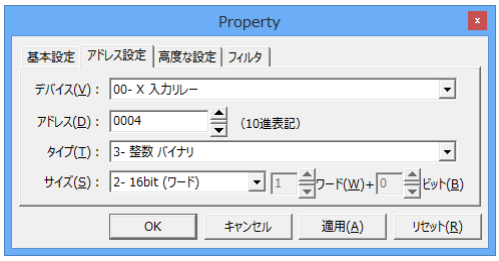
●ビットデバイス「X0000」をビットとして解釈する




●ビットデバイス「X0000～X0015」を 16 ビット整数として解釈する



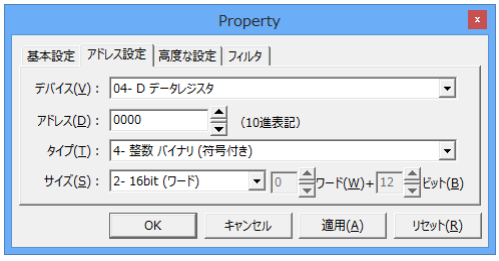

●ビットデバイス「X0004～X0035」を 16 ビット符号付き整数として解釈する



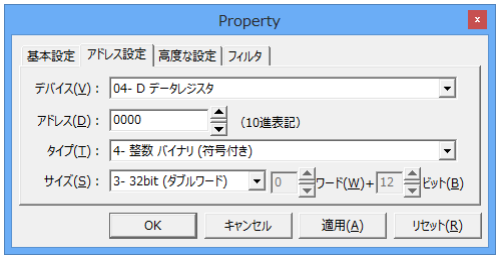

●ビットデバイス「X0004～X0015」の 12 ビットを 3 桁 BCD として解釈する



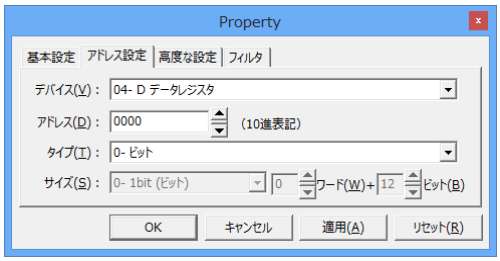

●ワードデバイス「D0000」を 16 ビット符号付き整数として解釈する



●ワードデバイス「D0000～D0001」を 32 ビット符号付き整数として解釈する




●ワードデバイス「D0000」の 16 ビット中、0 ビット目をビットとして解釈する




ワードデバイス「D0000」の 16 ビット中、3 ビット目をビットとして解釈する。




ワードデバイス「D0000」の 16 ビット中、4 ビット目から 7 ビット分を整数として解釈する。







ワードデバイスの場合、アドレスの末尾に「@」+1～15 までの整数を指定することによって、何ビット目かを示すことができます。同じようにバイトデバイスは「@」+1～7 までの整数を指定することができます。

上記で示した例のとおり、デバイスの解釈は以下のルールに従って行われます。

■アドレス
通信を行いたいデバイスの位置を指定します。ワードデバイスのような 16 ビット単位で表現されるアドレス体系でも、アドレスの末尾に「@1」などと指定することによって、ビット単位での指定も可能です。

■サイズ
タグのサイズを決定します。「アドレス」で指定したデバイスの位置から、何ビット分を対象とするかを指定します。

■タイプ
ビット列をどう解釈するかを決定します。同じビット列でも、バイナリとして取得するのか、BCD で取得するのか、あるいはアスキーで取得するのかによって値が異なります。

## フィルタ（型変換／工学値変換など）

フィルタはリアルタイムに生値を変換する機能です。現在 15 種類のフィルタが用意されています。

フィルタはタグごとに定義され、フィルタ番号順に実行されます。フィルタを実行すると値が型変換される場合があり、最終的に実行されたフィルタによる変換結果が、フィルタとしての最終型になります。フィルタによる変換の後、最終的には「型変換」の設定で指定した型に変換されます。

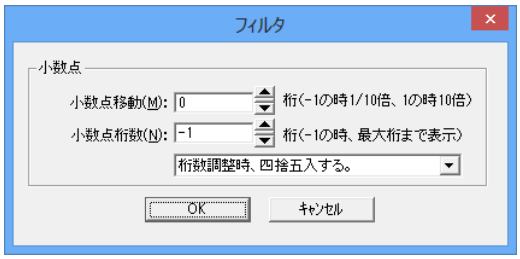
例）

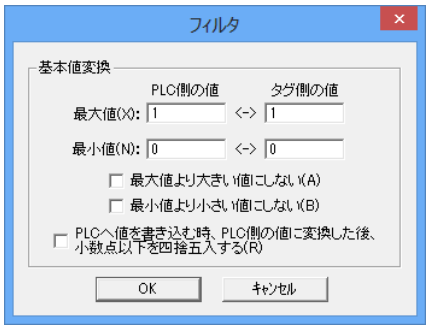


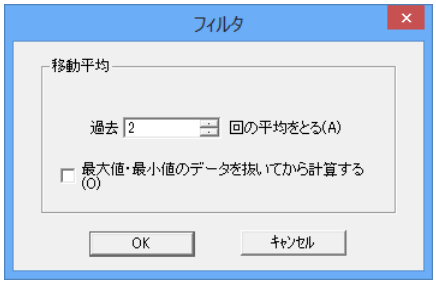

この例では「01 傾きとオフセットを変換する」、「02 移動平均する」の順に実行されます。「型変換」の設定が「デフォルト」であるため、移動平均フィルタの型、つまり数値タグになります。

フィルタには以下のものがあります。

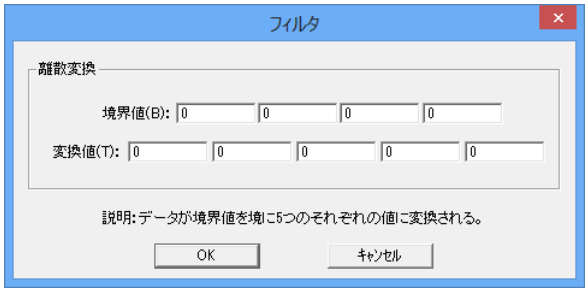
| ビットを反転する | |
|---|---|
| 実行後の型 | ビット型 |
| 解説 | TRUE を FALSE、FALSE を TRUE に変換します。値の書き込み時も逆変換が行われます。 |
| 設定画面 | なし |

| 小数点の移動と切り捨てを行う | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 小数点を移動させます。 |
| 設定画面 |  |

| 傾きとオフセットを変換する（工学値変換） | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 工学値変換を行います。単純に小数点を移動させたい場合、「小数点の移動と切り捨てを行う」を使用して下さい（計算誤差がでる場合があります）。変換後に最大値を超えないようにするには、「最大値より大きい値にしない」をチェックします。最小値についても同様です。 |
| 設定画面 | <br>【例】<br>\| \| PLC 側の値 \| タグ側の値 \|<br>\|---\|---\|---\|<br>\| 最大値 \| 4096 \| 20 \|<br>\| 最小値 \| 0 \| 0 \|<br>上記の設定の場合、PLC から値を読み込んだときには、タグの値は PLC の値の 20/4096 になります。反対に、アプリケーションによってタグの値が変更された場合には、その値を 4096/20 倍した値が PLC に書き込まれます。 |

| 移動平均 | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 過去 x 回の平均を求めます。 |
| 設定画面 |  |

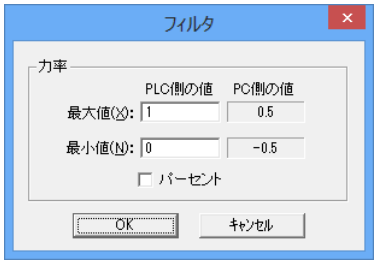
| 差分値（前回値―今回値）を求める | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 現在値から前回に取得したタグの値との差を計算して、その値を返します。 |
| 設定画面 | なし |

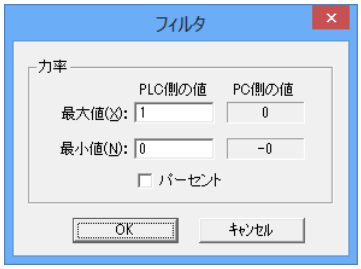
| 値の範囲から整数に変換する | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | タグの値に境界値を設定して、その値にしたがって他の値に変換します。 |
| 設定画面 | <br>例えば、デバイスの値が 20 未満の時はタグの値を 0、20 以上 40 未満のときは 1、40 以上 60 未満のときは 2…などのような変換を行うことができます。 |

| ON の回数をカウントアップする | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | ビットが ON になった回数をカウントアップします。デフォルトがビットタグの場合、型変換で「数値型」に設定して下さい。初回の値が ON の場合はカウントアップしません。値がいったん OFF になり、次に ON になった以降をカウントアップします。<br>値のリセットは「タグパス!ResetFilter = フィルタ ID」を実行することで行えます。OPC/DDE/IPLink などでは「タグパス!ResetFilter」というアイテムに対し、フィルタ ID を書き込んでください。「ResetFilter」についての詳細は本書中「システムプロパティ」の章を参照して下さい。 |
| 設定画面 | なし |

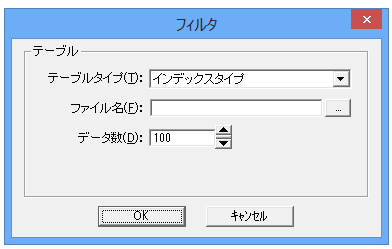
| 最大値を求める | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 最大値を求めます。タグのシステムプロパティ「ResetFilter」で、最大値を途中でリセットできます。 |
| 設定画面 | なし |

| 最小値を求める | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 最小値を求めます。タグのシステムプロパティ「ResetFilter」で、最小値を途中でリセットできます。 |
| 設定画面 | なし |

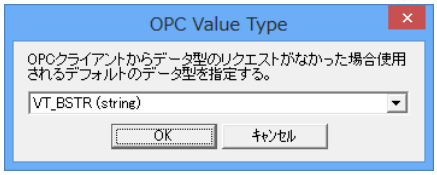
| 力率に変換する（－０．５～０．５） | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 力率に変換します。変換後の値は-0.5～1～0.5 になります。%表示（-50～100～50）にしたい場合は、「小数点の移動と切り捨てを行う」フィルタと組み合わせて下さい。 |
| 設定画面 |  |

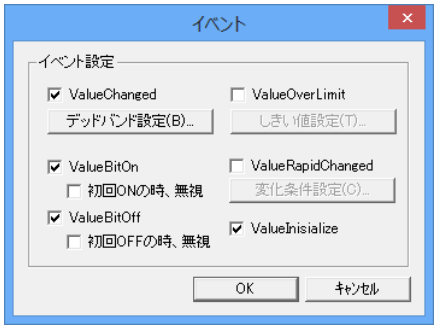
| 力率に変換する（－０～０） | |
|---|---|
| 実行後の型 | 数値型 |
| 解説 | 力率に変換します。変換後の値は-0～1～0 になります。%表示（-0～100～0）にしたい場合は、「小数点の移動と切り捨てを行う」フィルタと組み合わせて下さい。<br>例）PLC の最大値 1000、最小値 0 の場合 |
| 設定画面 |  |

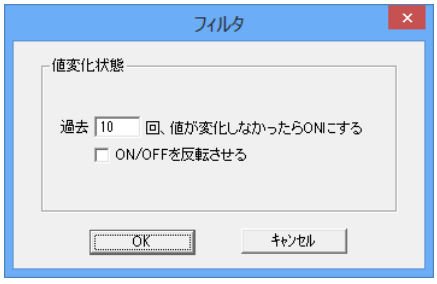
| PLC の値 | 力率変換値 |
|---|---|
| 1000 | +0.00 |
| 750 | +0.50 |
| 500 | 1.00 |
| 250 | -0.50 |
| 0 | -0.00 |

| テーブルファイルを参照して数値や文字列に変換する | |
|---|---|
| 実行後の型 | 文字列型 |
| 解説 | テーブルファイル（テキストファイル）から値に従って、文字を取り出します。文字列型に変換されるため、数値に変換したい場合は「型変換」で「数値型」を指定して下さい。テーブルファイルのタイプは下記の２種類に対応しています。<br>サンプルは以下に格納されています。<br>sample¥common¥tag¥TagFilter¥TBL¥sample01<br><br>■インデックスタイプ<br>まず、下記のようなテキストファイルを予め用意しておきます。<br>圧力 A が上限値を超えました<br>温度 A が上限値を超えました<br>圧力 B が上限値を超えました<br>温度 B が上限値を超えました<br>回転数が異常です<br>読み込んだ PLC デバイスの値を n+1 行目として、このファイルの行の内容を対応付けることができます。この例では、PLC の値が「0」の時、タグ値は「圧力 A が上限値を超えました」になり、PLC の値が「4」の時、タグ値は「回転数が異常です」になります。PLC の値が 5 以上の時は空文字になります（※PLC の値がビット型の場合、「FALSE」を 0、「TRUE」を 1 とします）。<br><br>■検索タイプ<br>まず、下記のようなテキストファイルを予め用意します。<br>101,圧力 A が上限値を超えました<br>102,温度 A が上限値を超えました<br>201,圧力 B が上限値を超えました<br>202,温度 B が上限値を超えました<br>300,回転数が異常です<br>読み込んだ PLC の値をテキスト検索のインデックスとして解釈し、テキストから該当する行を取得します。この例では、PLC の値が「101」の時、タグ値は「圧力 A が上限値を超えました」になり、PLC の値が「300」の時、タグ値は「回転数が異常です」になります。PLC の値が見当たらなければ、空文字になります<br>※PLC の値がビット型の場合、「FALSE」を 0、「TRUE」を 1 とします。<br>※通常、検索を行う値が入っているインデックス列は 0 列目、変換する値を 1 列目に記述します。それ以外のフォーマットを使用したい場合は、「インデックス列」「値列」を指定します。たとえば、アラームサーバーアクションで使用されるアラームマスターファイルを使用したいときは、「インデックス列」「値列」を設定することにより利用できます。 |
| 設定画面 |  |

| InTouch で７桁以上の整数を扱う | |
|---|---|
| 実行後の型 | 変換なし |
| 解説 | InTouch で７桁以上の整数を扱う場合はこのフィルタを指定して下さい。 |

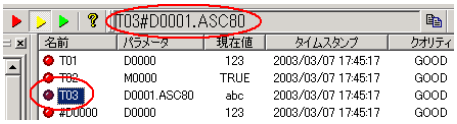
| OPC のデフォルトのデータ型を指定する | |
|---|---|
| 実行後の型 | 変化なし |
| 解説 | OPC クライアントでデータ型のリクエストがなかった場合、デフォルトでは数値タグは VT_R8、ビットタグは VT_BOOL、文字タグは VT_BSTR 型でクライアントに値を返します。これらのデータ型を他の型でクライアントに渡したいとき、このフィルタを使用します。 |
| 設定画面 |  |

| イベント指定（FA-Engine のみ有効） | |
|---|---|
| 解説 | FA-Engine を使用する場合のみ有効です。各イベントの設定を行うことができます。詳細は FA-Engine のユーザズマニュアルをご参照ください。 |
| 設定画面 |  |

| 値変化の状態を取得する | |
|---|---|
| 実行後の型 | ブール型 |
| 解説 | 指定回数値が変化しなかった場合、値を ON します。<br>ON/OFF を反転させることも可能です。 |
| 設定画面 |  |

## パラメータ文字列

パラメータ文字列とは、デバイス種別やアドレス、ビット位置、タイプなどを表す文字列です。接続する PLC に対してどのようなパラメータ文字列が指定されるのかを知りたい場合は、ユニット、フォルダ、タグ等を仮に設定し、ツールバーの表示から情報を得て下さい。下記の例では「T03#D0001.ASC80」となっています。




パラメータ文字列は主に以下の場面で使用されます。

■アクティブタグの設定
アクティブタグを設定する際には「#パラメータ文字列」を下記のように指定します。
例）

```
U01.F01.#D0000
U01.F01.#X0000.BIN16
U01.F01.#D0.ASC80
```

■アプリケーション設定ファイル（XML フォーマット）
設定ファイル内では、各オブジェクトを表現する文字列中にて「名前#パラメータ文字列」が使用されています。

■タグ設定ファイル（CSV フォーマット）
ファイル内でユニット、フォルダ、タグを表現する文字列中にて「名前#パラメータ文字列」が使用されています。

---




標準 PLC の場合、パラメータ文字列の書式は下記のとおりです（フィルタ設定を除く）。

**#**[デバイス種別][アドレス]**@**[ビット位置]**.**[タイプ][サイズ]

例 1）#X0　「X0」をビットとしてリンク
例 2）#D0　「D0」を 16 ビットのバイナリとしてリンク
例 3）#X0.BIN16　「X0」～「X15」を 16 ビットのバイナリとしてリンク
例 4）#X0.BCD8　「X0」～「X7」を 2 桁の BCD としてリンク
例 5）#D0@4.BCD12　「D0」の 4～15 ビットを 3 桁の BCD としてリンク
例 6）#D0.BIN-4　「D0」の 0～3 ビットを符号付バイナリとしてリンク
例 7）#D0.ASC80　「D0」～「D4」を 10 文字のアスキーとしてリンク
※2 バイト文字の場合、5 文字になります。

■[デバイス種別]（必須）
アクセスするデバイス名デバイスの名前（「X」「Y」「D」「M」など）を指定します。

■[アドレス]　（必須）
アクセスするデバイスのアドレスを指定します。

＜次頁につづく＞

<前頁のつづき>

■@[ビット位置]（省略可能）

アドレスの何ビット目から参照するのかをビット単位で指定することができます。その場合、ビット数はアドレスの下位から数え、最下位ビットは「0」になります。ビット位置を省略した場合は、最下位ビットから参照されます。

■[タイプ]（省略可能）

タグの型を、以下のキーワードで指定します。

| 型 | キーワード |
|---|---|
| ビット | BIT |
| 整数 BCD | BCD |
| 整数 BCD（符号付き） | BCD- |
| 整数 バイナリ | BIN |
| 整数 バイナリ（符号付き） | BIN- |
| 単精度浮動小数型(IEEE) | FLT |
| 倍精度浮動小数型(IEEE) | DBL |
| アスキー | ASC |

[タイプ]は省略可能です。省略した場合、指定したデバイスの単位がビットなら「BIT」が、バイトもしくはワードなら「BIN」が自動選択されます。但し、 [サイズ]を指定する場合には省略できません。

■[サイズ]（省略可能）

タグのサイズをビット単位で指定します。指定可能なサイズは、[タイプ]で指定された型により、下表のような制約があります。[タイプ]に「ビット」を指定した場合には、[サイズ]は指定できません。

| タイプ | 指定可能サイズ |
|---|---|
| ビット | 指定できません |
| 整数 BCD | 4/8/12/16/20/24/28/32 |
| 整数 BCD（符号付き） | 4/8/12/16/20/24/28/32 |
| 整数 バイナリ | 1～32 |
| 整数 バイナリ（符号付き） | 8/16/32 |
| 単精度浮動小数型(IEEE) | 指定できません |
| 倍精度浮動小数型(IEEE) | 指定できません |
| アスキー | 8 以上、8 の倍数 |

[サイズ]は省略可能で、省略した場合には指定された[タイプ]と、指定したデバイスの単位（各機種の「デバイス一覧」の表を参照）によって、下表のサイズが指定されたことになります。

| タイプ | ビットデバイス | バイトデバイス | ワードデバイス |
|---|---|---|---|
| 整数 BCD | 16 | 8 | 16 |
| 整数 BCD（符号付き） | 16 | 8 | 16 |
| 整数 バイナリ | 1 | 8 | 16 |
| 整数 バイナリ（符号付き） | 16 | 8 | 16 |
| アスキー | 16 | 8 | 16 |

## エイリアスタグについて

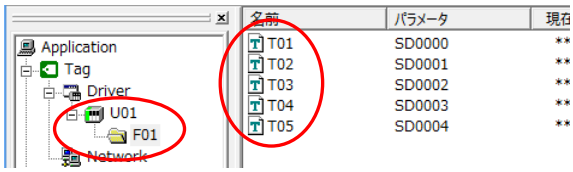
エイリアスタグとは、タグの名称を別の名称として参照できるようにする機能です。通常のタグとは別の名称で、タグの管理を行いたい場合や、外部からのアクセスを行いたいといった場合に利用することができる機能です。

この機能のサンプルは下記に格納されています。

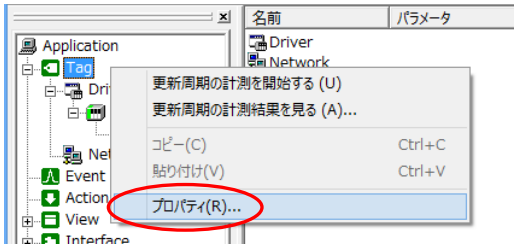
設定ファイル：¥¥sample¥common¥tag¥AliasTag¥sample01¥sample01.txt
エイリアスタグ定義ファイル：¥¥sample¥common¥tag¥AliasTag¥sample01¥AliasTag.csv

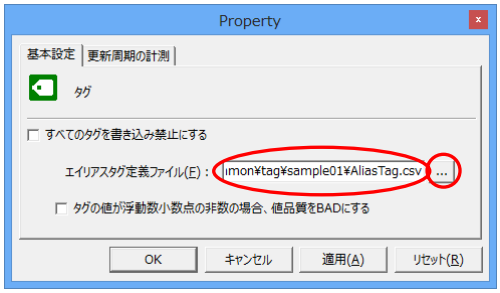
エイリアスタグの設定手順は以下のとおりです。

1．タグを定義する




2．エイリアスタグ定義ファイルを指定する
ワークスペースの Tag を右クリックして、プロパティを開き、定義ファイルを選択します。




エイリアス定義ファイルのフォーマットは、

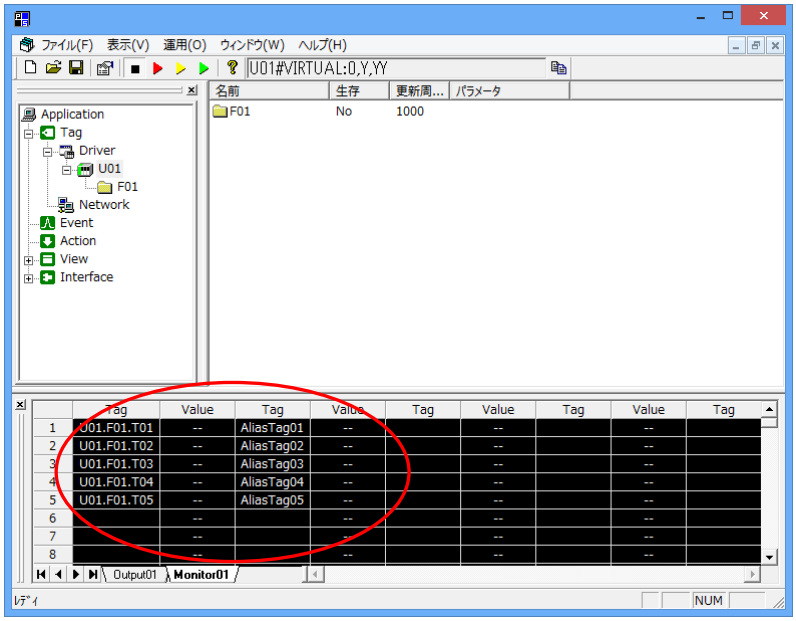
```
[エイリアスタグ名１],[対象タグのフルパス１]
[エイリアスタグ名２],[対象タグのフルパス２]
[エイリアスタグ名３],[対象タグのフルパス３]
[エイリアスタグ名４],[対象タグのフルパス４]
・・・
```

といった形式になります。

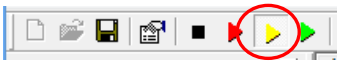
例）

```
AliasTag01,U01.F01.T01
AliasTag02,U01.F01.T02
AliasTag03,U01.F01.T03
AliasTag04,U01.F01.T04
AliasTag05,U01.F01.T05
```

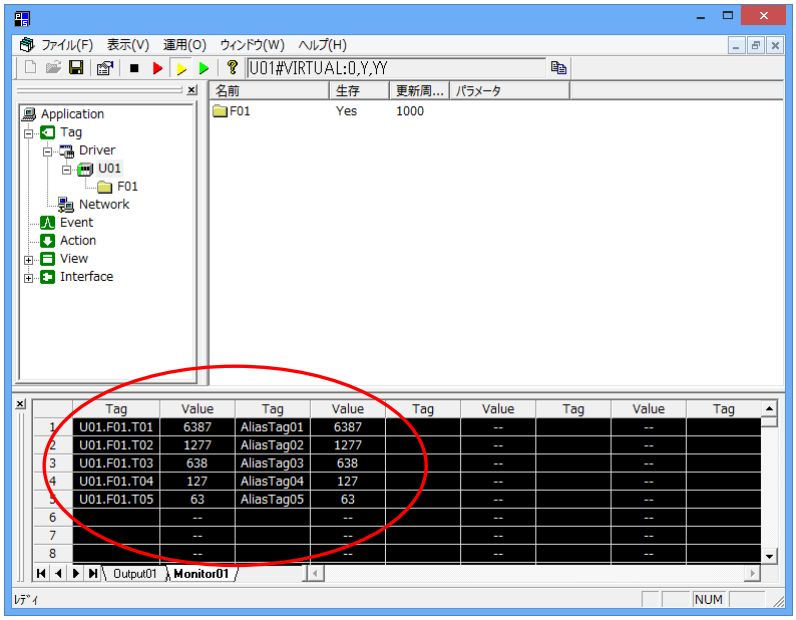
3. タグモニタを開き、通常のタグとエイリアスタグを設定する




4. オンラインにする



5. タグの値の変化と合わせて、エイリアスタグの値が更新されることを確認する




---

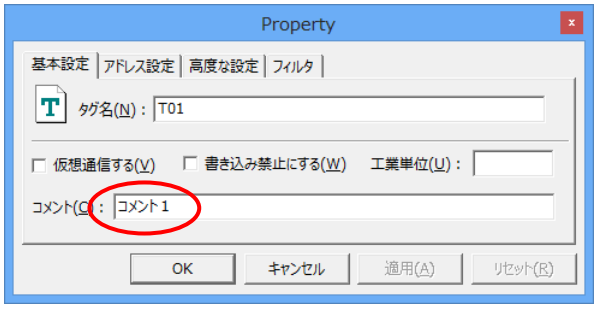
ヒント

エイリアスタグを設定すると、通常のタグとエイリアスタグどちらの名称でも、外部（OPC や Panel Browser 等）から参照可能になります。
エイリアスタグの値を書き換えた場合、通常のタグの値も連動して書き換わるため、参照のみではなく、値の更新も可能です。

---

## タグコメントについて

タグコメントとはタグごとにコメントを設定することができる機能です。

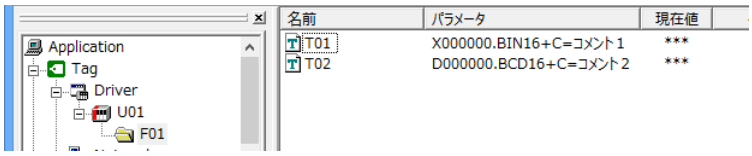



このコメントは外部ファイルをインポートすることで一括設定することが可能です。また、一括クリアすることも可能です。

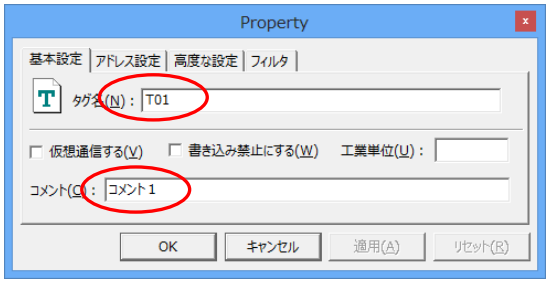
この機能のサンプルは下記に格納されています。

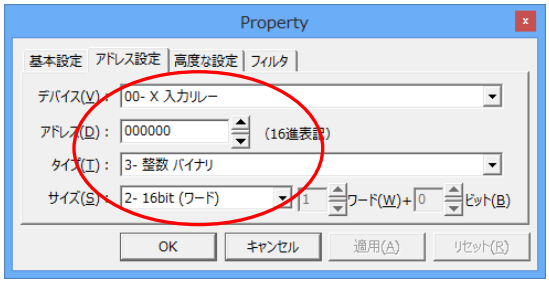
設定ファイル：¥¥sample¥common¥tag¥TagComment¥sample01¥sample01.txt
インポートファイル：¥¥sample¥common¥tag¥ TagComment¥sample01¥ TagComment01～03.csv
タグコメントの一括設定の手順は以下のとおりです。

1. タグを下記の内容で定義する
   ここでは、追加するユニットは「三菱（公開）」-「Ethernet」-「MELSEC-Q/QnU/QnUDE/L アクセス局 1」を使用するします。

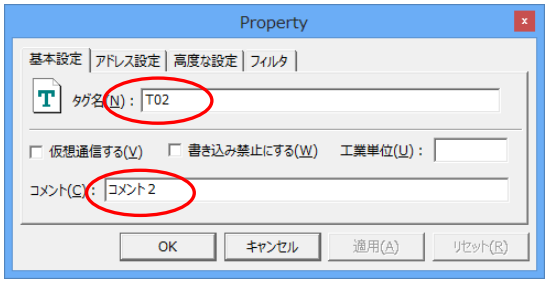


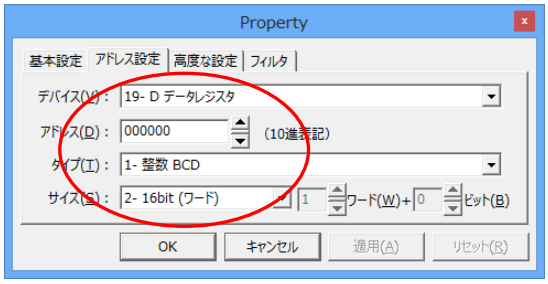

U01.F01.T01



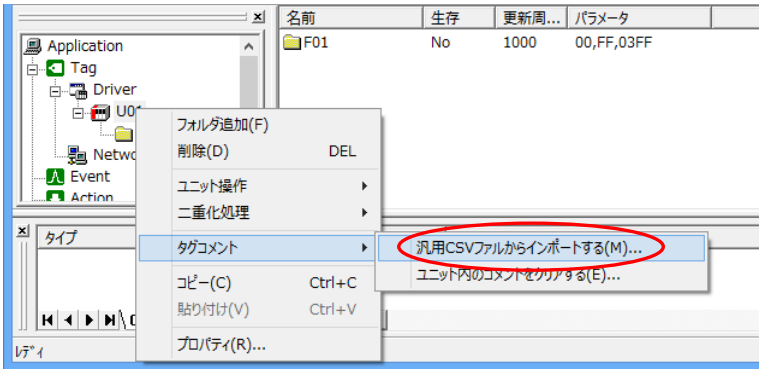




U01.F01.T02

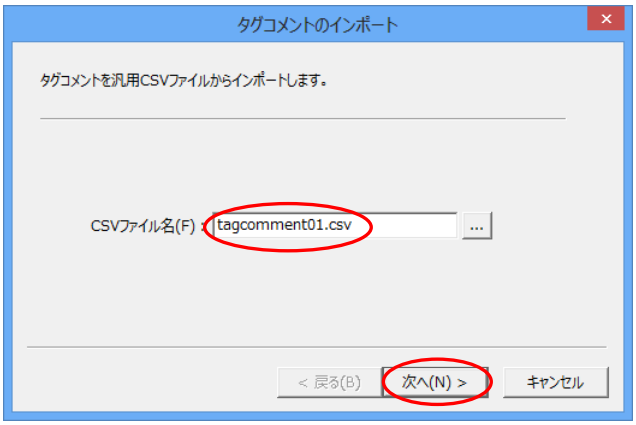


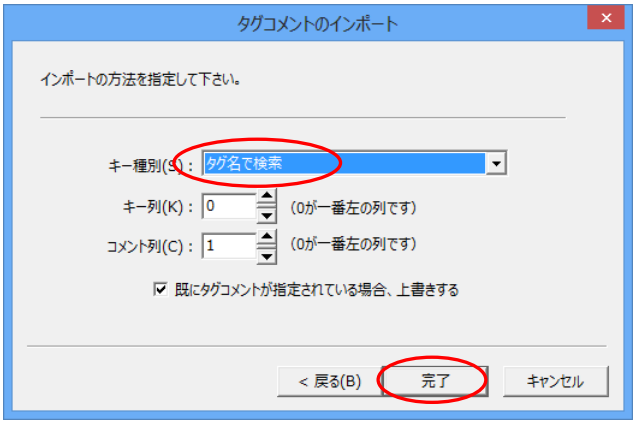
2. ユニットを右クリックして、表示されるメニューの以下のものを選択する



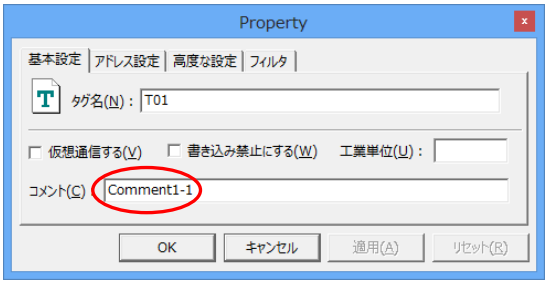

3. インポートファイルに tagcomment01.csv を選択して、次へボタンを押す

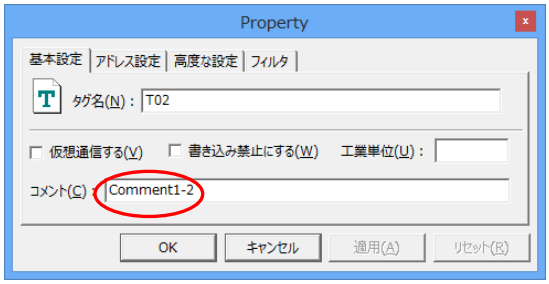



4. タグ名で検索を選び、完了ボタンを押す




5. コメントが上書きされ、インポートファイルの内容が反映されていることを、タグのプロパティを表示して確認する



インポートファイルのフォーマットは以下のとおりです。

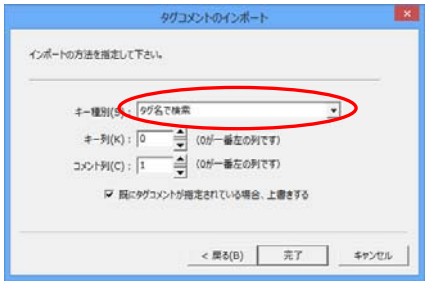
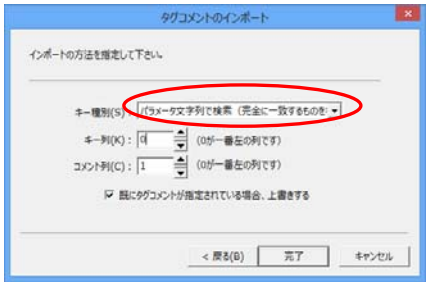
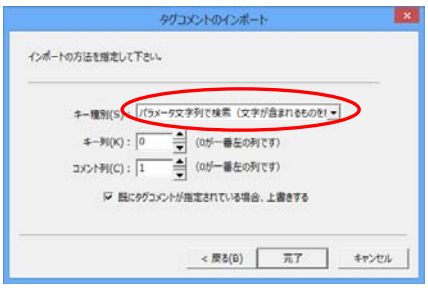
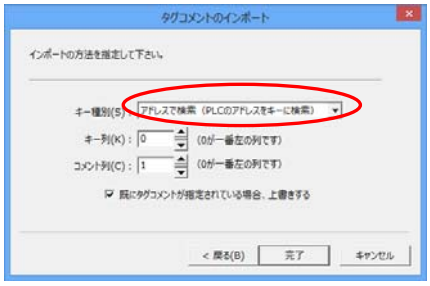
```
[キーワード 1],[コメント 1]
[キーワード 2],[コメント 2]
・・・
```

といった形式になります。

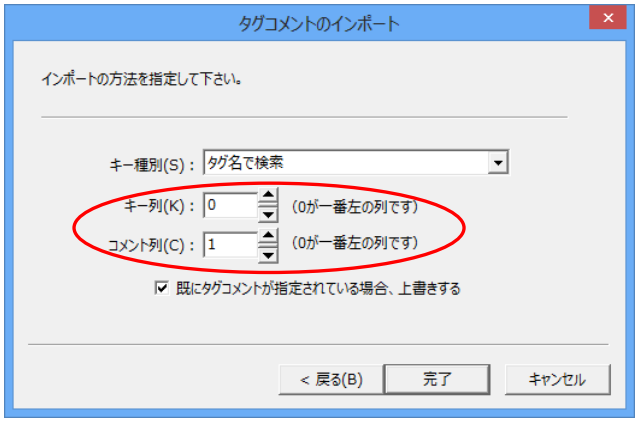
例）

```
T01,Comment1-1
T02,Comment1-2
```

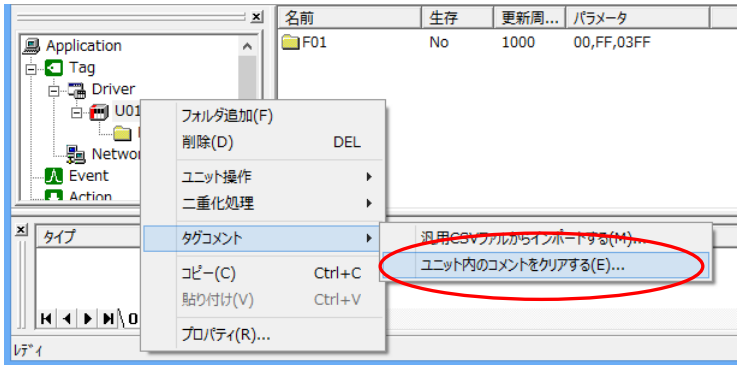
また、キーワードは以下のものを設定することができ、インポート時に、設定したキーの種別の選択を行います。

| | |
|---|---|
| タグ名で検索 | キーワードにタグ名を設定することで、該当のタグが存在していた場合、そのタグにコメントが設定されます。<br><br><br>インポートファイルの設定例）<br>`T01,Comment1-1`<br>`T02,Comment1-2`<br>インポートファイルのサンプル：tagcomment01.csv を参考にしてください。 |
| パラメータ文字列で検索 | キーワードにタグのパラメータを設定することで、該当のパラメータのタグが存在していた場合、そのタグにコメントが設定されます。キーワードのパラメータが「完全一致する」ものか「文字が含まれる」ものかどちらの検索を行うかを選ぶことができます。<br><br><br><br><br>インポートファイルの設定例）<br>`X000000.BIN16,Comment2-2`<br>`D000000.BCD16,Comment2-2`<br>インポートファイルのサンプル：tagcomment02.csv を参考にしてください。 |
| アドレスで検索 | キーワードにタグのアドレスを設定することで、該当のアドレスのタグが存在していた場合、そのタグにコメントが設定されます。<br><br><br>インポートファイルの設定例）<br>`X000000,comment3-1`<br>`D000000,comment3-2`<br>インポートファイルのサンプル：tagcomment03.csv を参考にしてください。 |

インポートファイルのフォーマットは自由に変更することが可能です。
フォーマットを変更した場合は、インポート時の設定を変更したファイルフォーマットに合わせて設定してください。




タグコメントを一括クリアしたい場合意は、ユニットを右クリックして、表示されるメニューから「タグコメント」-「ユニット内のコメントをクリアする」を選択します。



# ４章　システムプロパティ

## 概略

システムプロパティとは各オブジェクト（ユニット、フォルダ、タグなど）が標準で必ず持つプロパティです。システムプロパティは下記の書式で通常のタグと同じようにアクセスすることができます。

　　オブジェクトパス!プロパティ名

## システムプロパティの使用例

例えば下記の「Alive」プロパティは PLC の生存状態を表します。

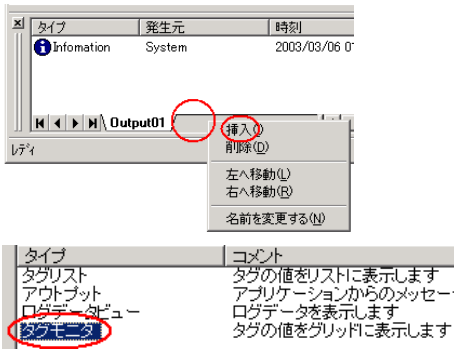
U01!Alive

以下は、Alive プロパティの値をタグモニタ上で確認する手順です。

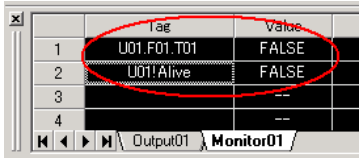
１．ユニット、フォルダ、タグを適当に作成する




２．画面下部のアウトプットビューの下記位置で右クリックし、「挿入」を選択し、「タグモニタ」を追加する




３．オンラインにした後、「U01.F01.T01」及び「U01!Alive」をセルに入力する




４．PLC の電源を ON/OFF することにより、U01!Alive の値が TRUE/FALSE に変化することを確認する

## システムプロパティ一覧

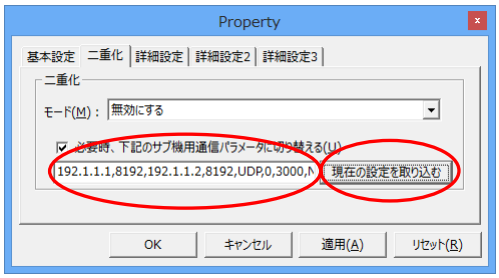
| | |
|---|---|
| 4－1 ユニット | |
| Alive | ユニットの生存状態。 |
| Virtual | 仮想的に通信にする。 |
| Update | 自動更新を行う。 |
| DriverMain | ユニットの通信パラメータ（メイン機）。 |
| DriverSub | ユニットの通信パラメータ（サブ機）。 |
| UseSub | サブ機の通信パラメータを使用する。 |
| Connect | ユニットの通信を接続/切断する。 |
| 4－2 フォルダ | |
| Alive | ユニットの生存状態。 |
| Virtual | 仮想的に通信にする。 |
| Update | 自動更新を行う。 |
| UpdateCount | 自動更新した回数。 |
| UpdateInterval | 更新周期。 |
| DriverMain | ユニットの通信パラメータ（メイン機）。 |
| DriverSub | ユニットの通信パラメータ（サブ機）。 |
| UseSub | サブ機の通信パラメータを使用する。 |
| OffsetBit | ビット単位でフォルダ内のタグのアドレスを移動する。 |
| OffsetTag | タグ単位でフォルダ内のタグのアドレスを移動する。 |
| 4－3 タグ | |
| ResetFilter | フィルタをリセットする。 |
| Comment | タグに設定されたコメントを取得する。 |
| EU | タグに設定された工業単位を取得する。 |

# ４－１　ユニット

| U01!Alive | |
|---|---|
| | ユニットの生存状態。 |
| 取得 | ビット型（TRUE - 生存 OK　FALSE – 生存 NG） |
| 解説 | デバイスダウンの設定により、連続して通信が失敗すると、デバイスダンと識別されます。デバイスダウン中は FALSE になります。 |

| U01!Virtual | |
|---|---|
| | 仮想的に通信する。 |
| 取得・設定 | ビット型（TRUE – 仮想通信中　FALSE – 仮想通信中でない） |
| 解説 | ユニット単位で、仮想通信の状態を切り替えます。 |

| U01!Update | |
|---|---|
| | 自動更新を行う。 |
| 取得・設定 | ビット型（TRUE – 自動更新中　FALSE –　自動更新中でない） |
| 解説 | ユニット単位で、自動更新の状態を切り替えます。このプロパティはすべてのフォルダに影響します。 |

| U01!DriverMain | |
|---|---|
| | ユニットの通信パラメータ（メイン機）。 |
| 取得・設定 | 文字列型 |
| 解説 | メイン機の通信パラメータです。メイン機との通信パラメータはこのプロパティの設定値が用いられます。通信パラメータの文字列はユニットの「二重化」タブで、「現在の設定を取り込む」で取得できます。通信中に設定を切り替えることも可能です。 |

| U01!DriverSub | |
|---|---|
| | ユニットの通信パラメータ（サブ機）。 |
| 取得・設定 | 文字列型 |
| 解説 | サブ機の通信パラメータです。サブ機との通信パラメータはこのプロパティの設定値が用いられます。このプロパティは２重化処理を行うときに使用します。DriverMain 同様、通信中に設定を切り替えることも可能です。 |

| U01!UseSub | |
|---|---|
| | サブ機の通信パラメータを使用する。 |
| 取得・設定 | ビット型（TRUE – サブ機使用中　FALSE –　サブ機使用中でない） |
| 解説 | このプロパティを操作することによって、２重化における通信先 PLC のメイン／サブを切り替えることができます。TRUE を設定するとサブ機側の通信パラメータに切り替わります。また、FALSE を設定するとメイン機側の通信パラメータに切り替わります。 |

| U01!Connect | |
|---|---|
| | ユニットの接続を接続/切断する。 |
| 取得・設定 | ビット型（TRUE – 接続　FALSE –　切断） |
| 解説 | このプロパティを操作することによって、アプリケーションから任意のタイミングで接続/切断を行うことができます。 |

## ４－２　フォルダ

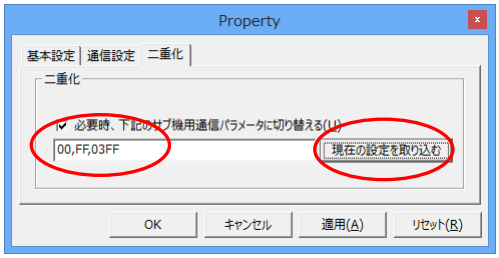
| U01.F01!Alive | |
|---|---|
| | フォルダの生存状態。 |
| 取得 | ビット型（TRUE - 生存 OK　FALSE – 生存 NG） |
| 解説 | デバイスダウンの設定に基づいて、連続して通信が失敗するとデバイスダンを検知します。デバイスダウン中は FALSE になります。 |

| U01.F01!Virtual | |
|---|---|
| | 仮想的に通信する。 |
| 取得・設定 | ビット型（TRUE – 仮想通信中　FALSE – 仮想通信中でない） |
| 解説 | フォルダ単位で、仮想通信の状態を切り替えます。 |

| U01.F01!Update | |
|---|---|
| | 自動更新を行う。 |
| 取得・設定 | ビット型（TRUE – 自動更新中　FALSE –　自動更新中でない） |
| 解説 | フォルダ単位で、自動更新の状態を切り替えます。 |

| U01.F01!UpdateCount | |
|---|---|
| | 自動更新した回数。 |
| 取得 | 数値型 |
| 解説 | 自動更新を行った回数を取得します。 |

| U01.F01!UpdateInterval | |
|---|---|
| | 更新周期。 |
| 取得・設定 | 数値型 |
| 解説 | 更新周期（ms 単位）を取得、設定します。 |

| U01.F01!DriverMain | |
|---|---|
| | フォルダの通信パラメータ（メイン機）。 |
| 取得・設定 | 文字列型 |
| 解説 | フォルダの通信パラメータ（メイン機側）です。メイン機との通信パラメータはこのプロパティの設定値が用いられます。通信パラメータの文字列はフォルダの「二重化」タブで、「現在の設定を取り込む」で取得できます。通信中に設定を切り替えることも可能です。<br><br> |

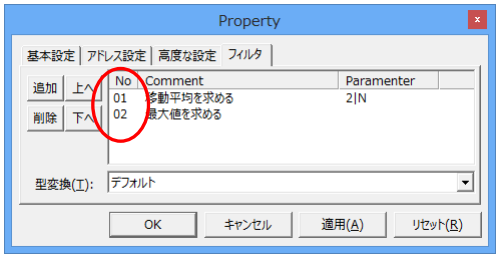
| U01.F01!DriverSub | |
|---|---|
| | フォルダの通信パラメータ（サブ機） |
| 取得・設定 | 文字列型 |
| 解説 | フォルダの通信パラメータ（サブ機側）です。サブ機との通信パラメータはこのプロパティの設定値が用いられます。このプロパティは２重化処理を行うときに使用します。DriverMain 同様、通信中に設定を切り替えることも可能です。 |

| U01.F01!UseSub | |
|---|---|
| | サブ機の通信パラメータを使用する。 |
| 取得・設定 | ビット型（TRUE – サブ機使用中　FALSE –　サブ機使用中でない） |
| 解説 | このプロパティを操作することによって、２重化における通信先 PLC のメイン／サブをフォルダ単位で切り替えることができます。TRUE を設定するとサブ機側の通信パラメータに切り替わります。また、FALSE を設定するとメイン機側の通信パラメータに切り替わります。 |

| U01.F01!OffsetBIt | |
|---|---|
| | フォルダ内のタグのアドレスをビット単位でオフセット指定する。 |
| 取得・設定 | 数値型 |
| 解説 | フォルダ内の全タグのアドレスをビット単位でオフセットします。例えば、アドレスが D0001（ワードレジスタ）のタグは「16」と設定すると１ワード分ずれ、D0002 になります。アドレスが M0001（ビットレジスタ）の場合は M0017 になります。 |

| U01.F01!OffsetTag | |
|---|---|
| | フォルダ内のタグのアドレスをタグ単位でオフセット指定する。 |
| 取得・設定 | 数値型 |
| 解説 | フォルダ内の全タグのアドレスをタグのサイズを単位としてオフセットします。例えば、タグが D0001（ワードレジスタ）のタグは、D0002 になります。アドレスが M0001（ビットレジスタ）の場合は M0002 になります。 |

# 4－3　タグ

| U01.F01.T01!ResetFilter | |
|---|---|
| | フィルタによる値の保持をリセットする（「最大値」、「最小値」、「ON の回数をカウントアップする」、「差分値」のフィルタに対して、値をリセットする）。 |
| 設定 | 数値型（フィルタ ID） |
| 解説 | 対象となるフィルタ ID を指定して、そのフィルタによって保持されている値をリセットします（最大値、最小値、ON 回数のカウント、差分値）。フィルタ ID は下記ダイアログで左に表示されている番号です。0 を指定するとすべてのフィルタの値をリセットします。<br> |

| U01.F01.T01!Comment | |
|---|---|
| | タグに設定されたコメントを取得することができます。 |
| 取得 | 文字列型 |
| 解説 | このプロパティを使用することでタグに設定されたコメントを取得することができます。ただし、取得は可能ですが設定することはできません。そのため、コメントを変更する場合は、タグのプロパティで直接変更する必要があります。 |

| U01.F01.T01!EU | |
|---|---|
| | タグに設定された工業単位を取得することができます。 |
| 取得 | 文字列型 |
| 解説 | このプロパティを使用することでタグに設定された工業単位を取得することができます。ただし、取得は可能ですが設定することはできません。そのため、工業単位を変更する場合は、タグのプロパティで直接変更する必要があります。 |

「タグマニュアル」Copyright © 2010 Roboticsware, Inc. All rights reserved.
2010 年 6 月 1 日 発行　著作・発行 株式会社ロボティクスウェア
〒331-0811 埼玉県さいたま市北区吉野町 1-11-1　TEL:048-782-9861 FAX:048-782-9862

■本書は著作権上の保護を受けています。本書の一部または全部について、株式会社ロボティクスウェアの
文章での許諾を得ずに、いかなる方法においても無断で複写・複製することは禁じられています。
■記載されている会社名、製品名、ブランド名は、対応する法人または個人の（登録）商標です。
■このソフトウェアの仕様及び本書に記載されている事項は、将来予告無しに変更することがあります。