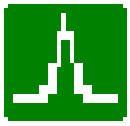
Roboticsware



Automation シリーズ

イベントマニュアル

RWMN-5-21-930

ようこそ **4**

本マニュアルについて 4
本マニュアルで紹介するアクション 4
仮想デバイスとは 5

１章　イベントの概要 **6**

イベントの位置付けと目的 6
定周期イベント 7
定刻イベント 8
タグイベント 9

２章　はじめてのイベント **10**

概要 10
ステップ１　タグの準備 11
ステップ２　アクションの準備 12
ステップ３　ビューの準備 13
ステップ４　定周期イベント 14
ステップ５　定刻イベント 16
ステップ６　タグイベント 18

３章　イベントの機能解説 **21**

アクションの実行順序とイベントの並列処理 21
オンライン中のイベントの有効・無効の切り替え 22
フォルダでイベントをまとめる 24
イベント属性 24
タグイベントの比較条件 25

４章　イベントの実践 **29**

４－１　ビットが ON の間のみイベントを有効にする（SAMPLE01） 29
概要 29
設定手順 29
解説 30
４－２　イベントを一定期間有効にする（SAMPLE02） 31
概要 31
設定手順 31
解説 32
４－３　イベントを条件により切り替える（SAMPLE03） 33
概要 33
設定手順 33
解説 34
４－４　イベントを設定回数だけ実行する（SAMPLE04） 35
概要 35
設定手順 35
解説 36

５章　システムプロパティとイベント **37**

概略 37
システムプロパティの使用例 37

システムプロパティ一覧 38
５－１　フォルダ 39
５－２　イベント 40

ようこそ

本マニュアルについて

本書「イベントマニュアル」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。

本マニュアルで紹介するアクション

本パッケージのアクションはいくつかの機能で構成されていますが、本マニュアルではアクションを代表して「スクリプト Ver1 アクション」を用いて解説するものとします。イベントからアクションを呼び出す方法を理解すれば、他のアクションについても同様の手順で設定を行うことができます。「スクリプト Ver1 アクション」の詳細については「アクションマニュアル」を参照して下さい。

仮想デバイスとは

本パッケージは現在 100 機種相当と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

SM0000 10ms ごとに On/Off
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。

1章　イベントの概要

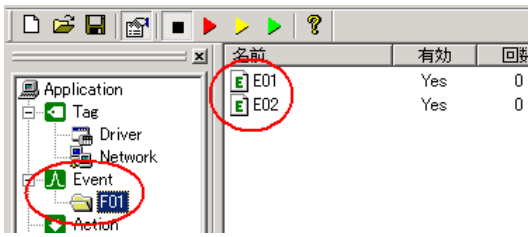
イベントの位置付けと目的

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「イベント」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。



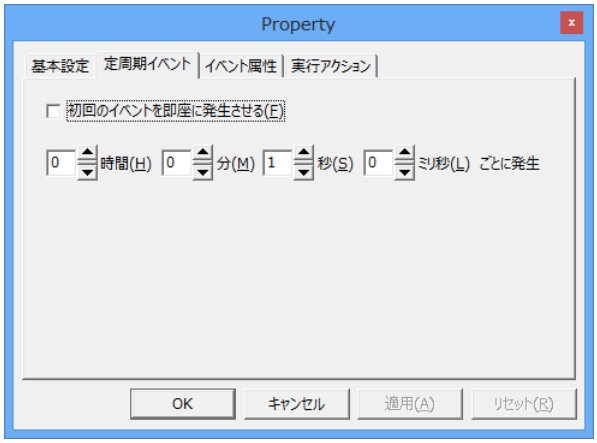
イベントの追加は画面左側ツリーの「イベント」アイコンから行います。



イベントはアクションを実行するトリガです。ほとんどのアクションはイベントから呼び出して実行します。
イベントには下記の３種類があります。

定周期イベント	一定周期間隔でアクションを実行します。
定刻イベント	指定した時刻にアクションを実行します。
タグイベント	タグ値を評価し、条件が成立した時にアクションを実行します。値変化、値一致、不一致、大なり、小なりなどの条件を組み合わせます。

定周期イベント

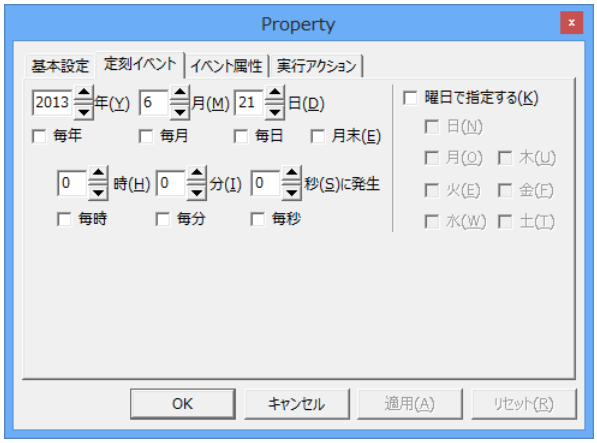


定周期イベントは、一定間隔で発生するイベントです。時間からミリ秒単位までの周期を指定できます。すべて「0」と設定すると、可能な限り速い周期でイベントを発生させます。

オンライン直後に即座に初回のイベントを発生させたい場合には「初回のイベントを即座に発生させる」をチェックします。

例）１分と設定した場合
「初回のイベントを即座に発生させる」ON 時：オンライン直後、即座に発生します。
「初回のイベントを即座に発生させる」OFF 時：オンラインから１分経過した時に発生します。

定刻イベント



設定した時刻になったときに発生します。



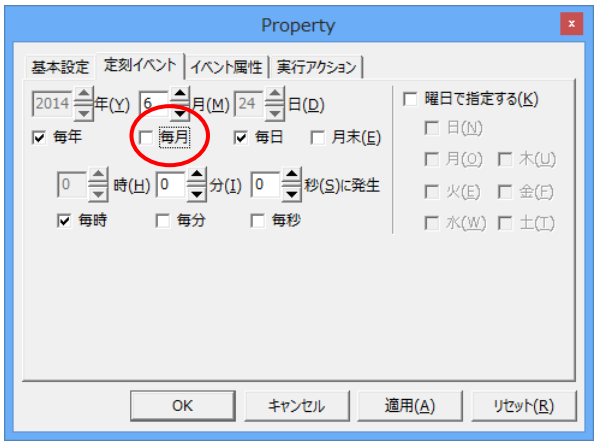
定刻イベントの「毎」設定は順番に設定を行う必要があるため注意が必要です。

例）

・順に設定がされている場合は OK です。



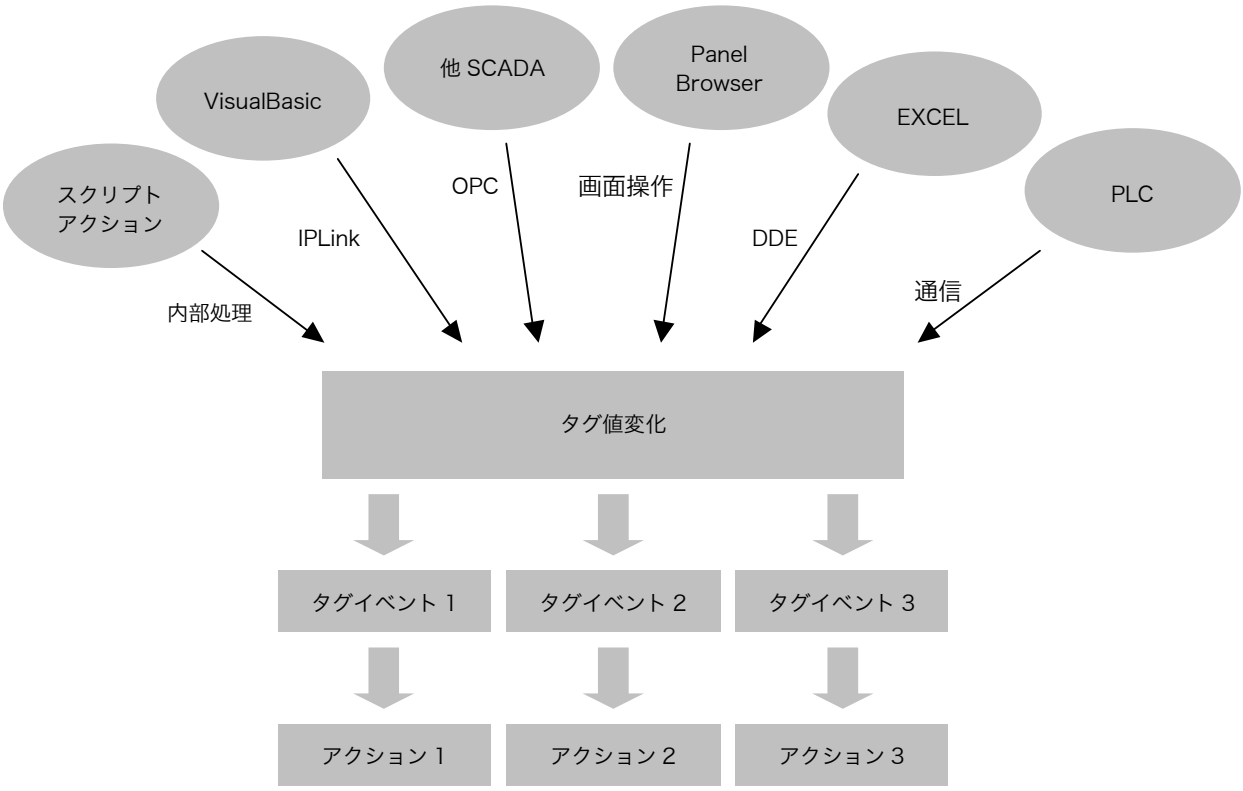
・順に設定がされていない場合は NG です。



※「毎日」、「毎時」にチェックは入っているが「毎月」の設定が行われていないため NG

タグイベント

タグイベントは、タグの値を評価して、条件の成立時に発生するイベントです。タグイベントは、様々な個所からアクションを実行するためのトリガとして利用することができます。



以下はタグイベントの利用例です。

■PLC のラダーからアクションを実行
ラダーからイベントのトリガとしたい PLC デバイスの値を ON/OFF させ、そのデバイスとリンクしたタグ値の評価によってイベントを発生させることで、PLC からアクションを実行させることができます。例えば、PLC のラダーロジックからレポート印刷やメール送信などのトリガをかけることができます。また、システムプロパティを用いることで、PLC と通信不能になった時に自動的にメールを送信するなどの応用も可能です。

■SCADA/VisualBasic/Excel などからアクションを実行
OPC/DDE/IPLink などのインターフェースを経由して、外部からタグの値を更新することによってアクションのトリガを発生させることができます。

■Panel Browser の画面操作からアクションを実行
Panel Browser の画面からタグの値を更新し、タグ値の変化によるアクションのトリガをかけることができます。

■スクリプト Ver1 アクションで実行
計算結果によりアクションを実行させたい場合はスクリプトアクションを用います。

2章　はじめてのイベント

概要

ここでは、実際にイベントを設定する手順について、以下の 5 ステップに分けて解説します。

ステップ1	タグを登録します。
ステップ2	イベントから呼び出すアクションを作成します。ここでは例として、「スクリプト Ver1」アクションを作成します。
ステップ3	ビューを作成します。ここではタグ値の確認用に「タグモニタ」を追加します。
ステップ4	定周期イベントを設定します。
ステップ5	定刻イベントを設定します。
ステップ6	タグイベントを設定します。

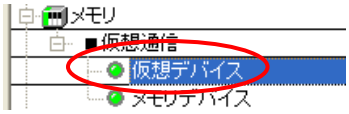
ステップ１　タグの準備

まず、準備としてタグの登録を行います。ここでは、仮想デバイスのタグを作成します。

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」を選択し、「OK」を押す



3．「U01」を右クリックし「フォルダ追加」を選択



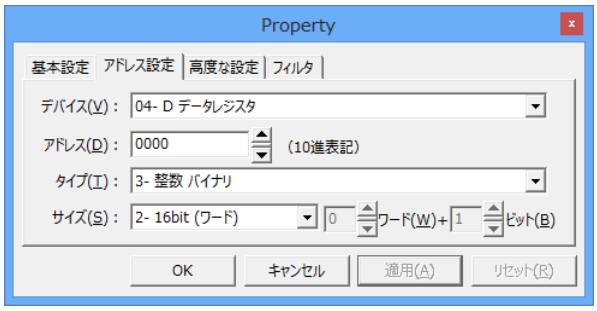
4．「F01」を右クリックし「タグ追加」を選択



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



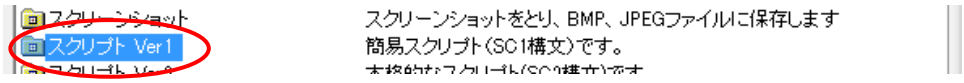
ここでは、D0000（ワードレジスタ）を数値タグ（整数バイナリ）として設定しています。

上記設定によって、数値タグ「U01.F01.T01」が設定されました。

ステップ２　アクションの準備

続いて、イベントから呼び出すアクションを作成します。ここでは例として「スクリプト Ver1」アクションを作成します。

1．アクションを追加する



画面左側のツリーから「Action」を右クリックし、「追加」「アクション」を選択します。選択画面が表示されるので、「スクリプト Ver1」アクションを追加してください。すると、「A01」というアクションが作成され、ツリーに追加されます。

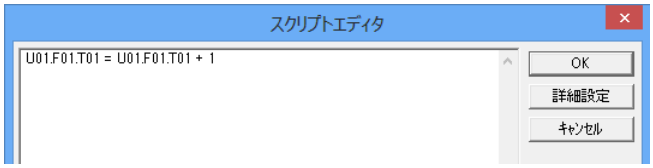
2．「A01」を右クリックし「プロパティ」を選択



3．スクリプトを設定する



スクリプトタブの右のボタンを押すと、スクリプトエディタが表示されます。



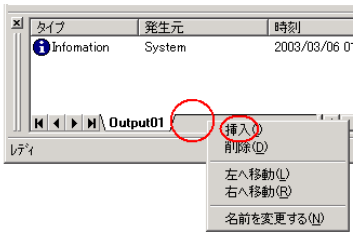
スクリプトエディタから、「U01.F01.T01 = U01.F01.T01 + 1」と入力してください。

ヒント

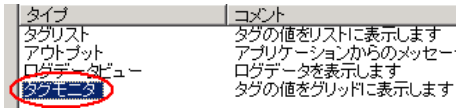
このアクションの設定により、スクリプト Ver1 アクションを実行するたびに「U01.F01.T01」がインクリメントされる動作となります。

ステップ３　ビューの準備

1．タグモニタを追加する



画面下部のアウトプットビューの上記位置で右クリックし、「挿入」を選択してください。選択画面が表示されるので「タグモニタ」を追加してください。

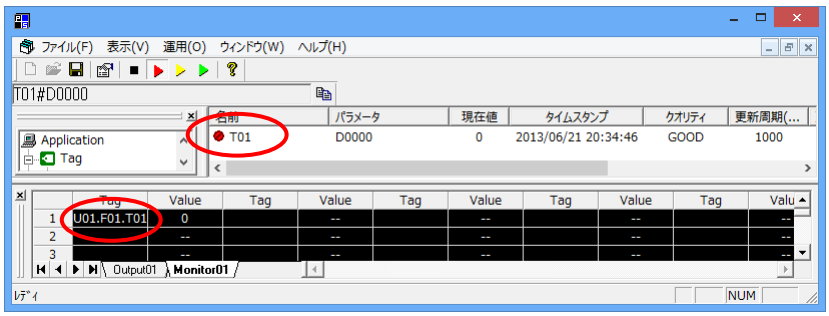


2．オンライン（モニタ）にする

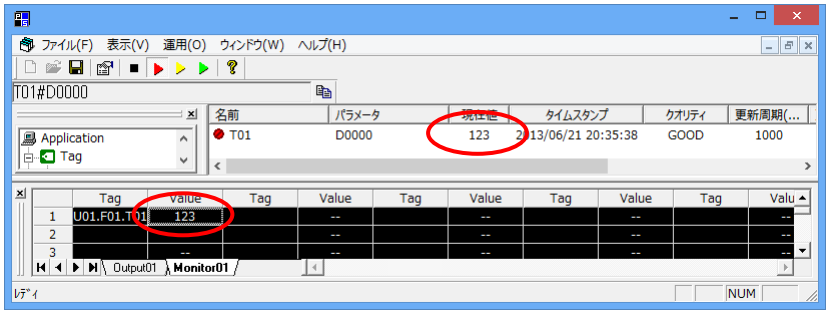


3．タグモニタからタグ値を確認する

タグモニタのタブを選択し、１行目のセルにタグパス「U01.F01.T01」を入力してください。



T01 のランプ表示が赤色になり、セルの Value にタグの値が表示されます。初期の状態では、Value の列には「0」が表示されています。ここで、セルから適当な数値を書き込んでください。すると、「U01.F01.T01」タグに値が書き込まれます（タグリストの現在値が変化することを確認してください）。



4．オフラインにする

確認ができたらオフラインにして下さい。



ステップ４　定周期イベント

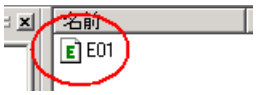
ここでは、定周期イベントを設定し、あらかじめ作成済みの「スクリプト Ver1」アクション（A01）の呼び出しを行います。

1．左側のツリーの「Event」を右クリックし「追加」「定周期イベント」を選択

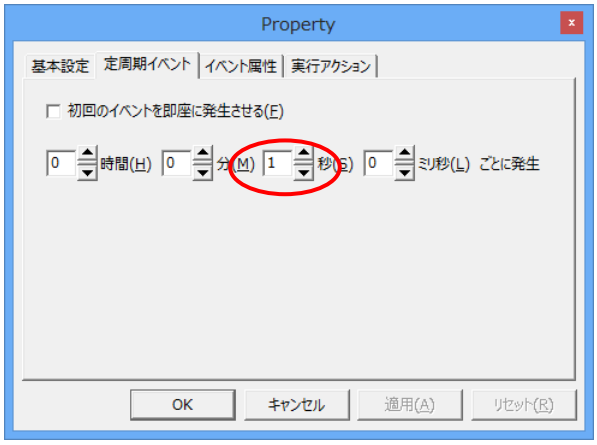


「E01」というイベントが追加されます。

2．左側のリストから「E01」を右クリックし、「プロパティ」を選択

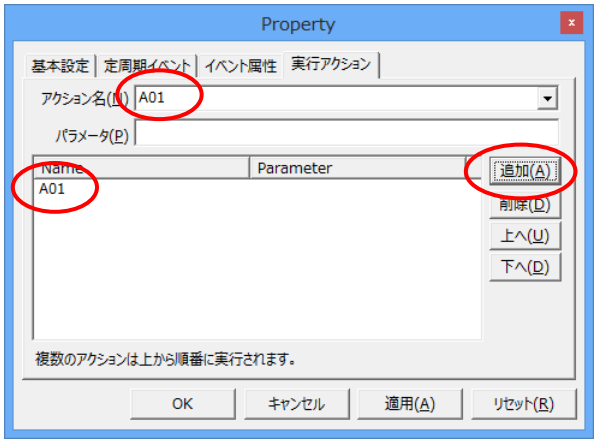


3．定周期イベントタブで、１秒が設定されていることを確認



「１秒」はデフォルト設定です。

4．実行アクションタブで「A01」を追加する



この設定により、E01 のイベントが発生する度に、A01 アクションが呼び出されます。

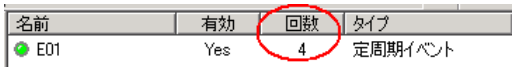
5．オンライン（モニタ）にする

イベントの動作確認を行います。アイコンをクリックし、オンライン状態にしてください。



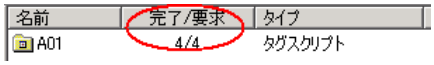
6．下記の３箇所を確認する

■左ツリーの「Event」を選択



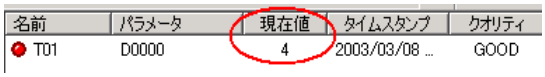
イベント「E01」のランプが緑色になり、回数が１秒ごとに増えていきます。これは、１秒周期でイベントが発生していることを意味します。

■左ツリーの「Action」を選択



アクション「A01」の「完了/要求」の回数が１秒ごとに増えていきます。これは、１秒ごとに発生したイベントによりアクションが実行されていることを意味します。

■左ツリーの「U01/F01」を選択



タグ「T01」の値が１秒ごとに変化します。これは、アクション「A01」のスクリプトによって「U01.F01.T01 = U01.F01.T01 + 1」が実行され、その度に現在値がインクリメントされていることを意味します。

7．オフラインにする

確認が終了したら、オフラインに戻して下さい。





定周期イベントタブで「0」ms に設定すると、できるだけ速い周期でイベントを発生させます。速度は最速で処理されますが、その反面、ＣＰＵ負荷が高くなりすぎてしまう場合があります。そのような場合には、10ms～100ms 程度の周期に設定するなどの調整を行い、CPU 負荷と処理実行周期のバランスをとって下さい。

ステップ５　定刻イベント

ここでは、定刻イベントを設定し、あらかじめ作成済みの「スクリプト Ver1」アクション（A01）の呼び出しを行います。尚、ステップ３までの作業が終了していることを前提に解説を進めます。

1．ステップ４で作成した定周期イベントを削除する
画面左側のツリーから、ステップ４の定周期イベント「E01」を削除して下さい。

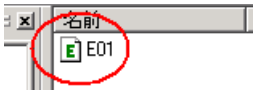
2．定刻イベントの追加

画面左側のツリーから「Event」を右クリックし、「追加」「定刻イベント」を選択してください。



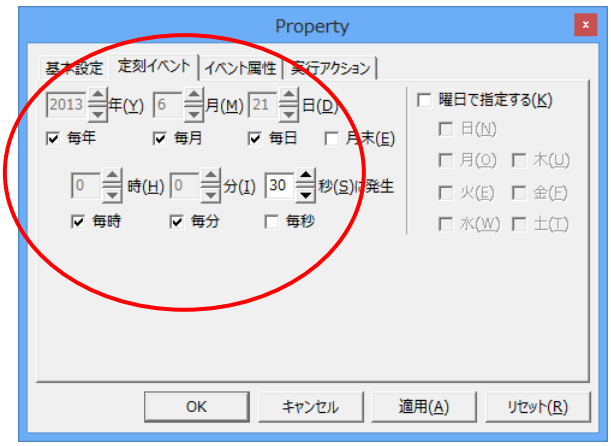
新たに「E01」が作成されます。

3．左側のリストから「E01」を右クリックし、「プロパティ」を選択



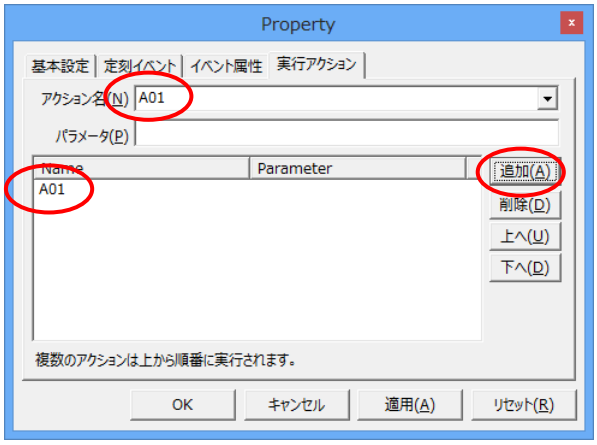
プロパティダイアログが表示されます。

4．定刻イベントタブで下記のように設定する



・毎年/毎月/毎日/毎時/毎分をチェック
・30 秒に発生

5．実行アクションタブで「A01」を追加する



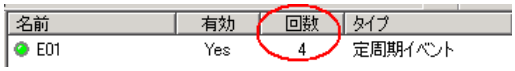
この設定により、E01 のイベントが発生する度に、A01 アクションが呼び出されます。

6．オンライン（モニタ）にする

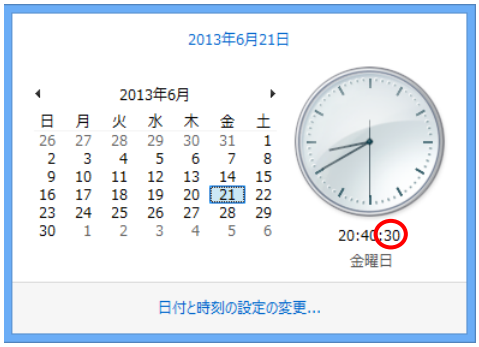


7．下記の３箇所を確認する

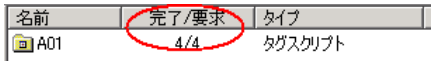
■左ツリーの「Event」を選択



イベント「E01」のランプが緑色になり、時刻が３０秒になるごとに回数が増えていきます。これは、時刻が毎分 30 秒になる度に発生したイベントによりアクションが実行されていることを意味します。

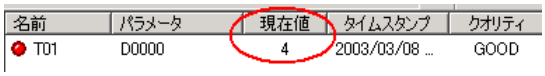


■左ツリーの「Action」を選択



イベント「E01」によりアクションが実行され、アクション「A01」の「完了/要求」の回数が増えていきます。

■左ツリーの「U01/F01」を選択



アクション「A01」によりスクリプトが実行され、そのたびに現在値がインクリメントされます。

8．オフラインにする

確認が終了したら、オフラインに戻して下さい。



ステップ６　タグイベント

ここでは、タグイベントを設定し、あらかじめ作成済みの「スクリプト Ver1」アクション（A01）の呼び出しを行います。尚、ステップ５までの作業が終了していることを前提に解説を進めます。

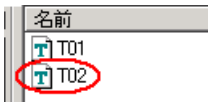
1．ステップ５で作成した定周期イベントを削除する
画面左側のツリーから、ステップ５の定刻イベント「E01」を削除して下さい。

2．新たにタグ「T02」を追加する



「F01」を右クリックし「タグ追加」を選択します。

3．画面左側のリストから「T02」を右クリックし、「プロパティ」を選択



4．T02 のプロパティを設定する



「アドレス設定」タブから、「04 - M 内部リレー」を選択して「OK」を押します。

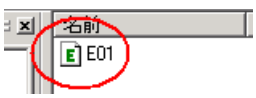
この設定により、T02 タグはビットタグとして設定されました（M0000（ビットレジスタ））。

5．画面左側のツリーの「Event」を右クリックし「追加」「タグイベント」を選択



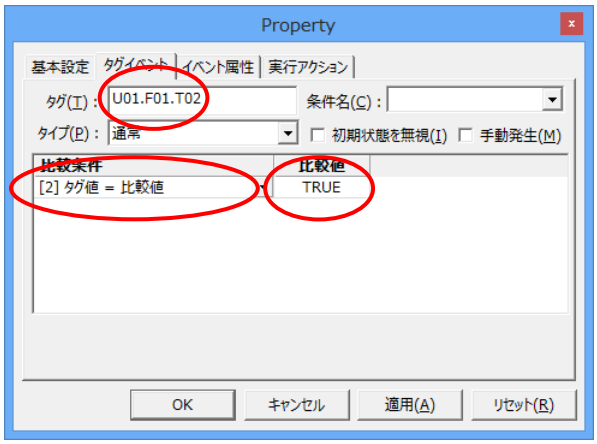
新たに E01 が追加されます。

6．左側のリストから「E01」を右クリックし、「プロパティ」を選択



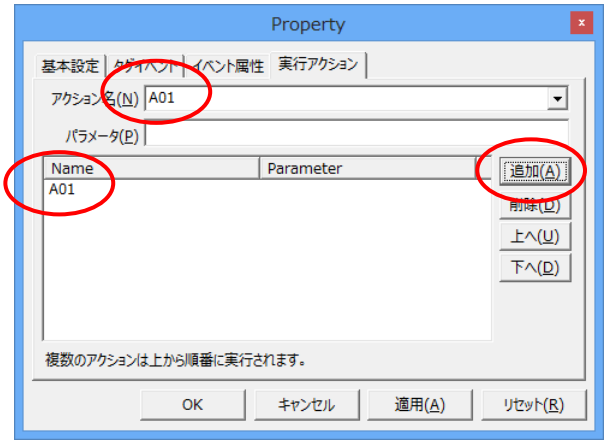
プロパティダイアログが表示されます。

7．タグイベントタブで下記のように設定する



上記の設定により、「U01.F01.T02」が TRUE（ON）になった時、イベントが発生します。

8．実行アクションタブで「A01」を追加する

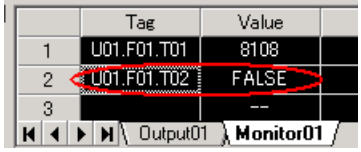


この設定により、E01 のイベントが発生する度に、A01 アクションが呼び出されます。

9．オンライン（モニタ）にする

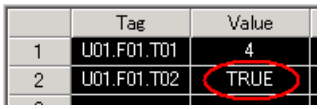


１０．タグモニタに「U01.F01.T02」と入力する



Value 列に「FALSE」が表示されます。

１１．「U01.F01.T02」タグの値を操作し、イベントを確認する



	Tag	Value
1	U01.F01.T01	4
2	U01.F01.T02	TRUE

「Value」列のセルに「TRUE」「FALSE」と繰り返し書き込みます。

下記の３箇所を確認してください。

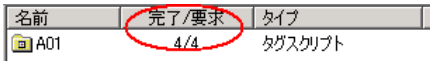
■左ツリーの「Event」を選択



名前	有効	回数	タイプ
E01	Yes	4	定周期イベント

イベント「E01」のランプが緑色になり、「U01.F01.T02」が TRUE になるごとに回数が増えていきます（タグ値の変化によりイベントが実行されている）。

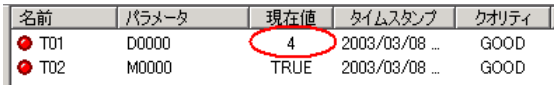
■左ツリーの「Action」を選択



名前	完了/要求	タイプ
A01	4/4	タグスクリプト

イベント「E01」によりアクションが実行され、アクション「A01」の「完了/要求」の回数が増えていきます。

■左ツリーの「U01/F01」を選択



名前	パラメータ	現在値	タイムスタンプ	クオリティ
T01	D0000	4	2003/03/08 ...	GOOD
T02	M0000	TRUE	2003/03/08 ...	GOOD

アクション「A01」により、スクリプトが実行され、そのたびに現在値がインクリメントされます。

１２．オフラインにする

確認が終了したら、オフラインに戻して下さい。

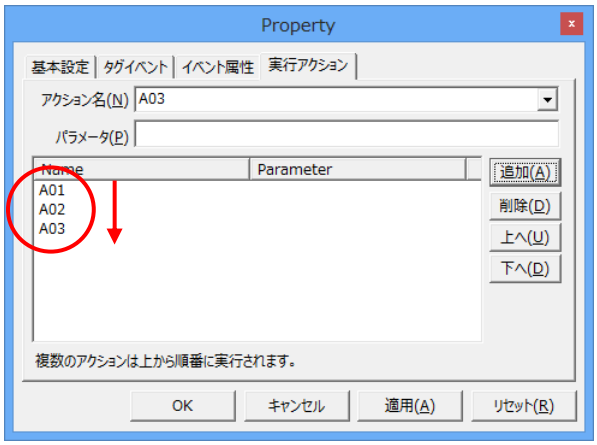


3章　イベントの機能解説

アクションの実行順序とイベントの並列処理

■アクションは上から順番に実行される

１つのイベントに対して、アクションは複数定義できます。複数のアクションを定義した場合、イベントが発生する度に上から順番にアクションが実行されます。



■各イベントは独立してアクションを実行する

各イベントの処理は独立しており、複数のイベントが同時に発生した場合でも、各アクションは他のイベントに影響されることなく並列的に実行されます。

例えば、２つのデータベースに対して１秒周期でログを取りたい場合、１つの定周期イベントに２つのロガーアクションを定義するよりも、２つの定周期イベントを作成して、それぞれにロガーアクションを定義した方が高速に処理されます。

例１）E01 内に２つのアクションを定義

Name	Parameter
A01	
A02	

イベントが発生すると、「A01」の処理が終了してから「A02」の処理が実行されます。

例２）E01/E02 にそれぞれ異なるアクションを定義

「E01」

Name	Parameter
A01	

「E02」

Name	Parameter
A02	

イベントはそれぞれ独立して発生し、A01 の処理と A02 の処理はお互いに影響を及ぼしません（但し、処理の内容が重複する場合、例えば同じ PLC に対して通信を行ったりする場合などは影響があります）。

例３）E01/E02 にそれぞれ同じアクションを定義

「E01」

Name	Parameter
A01	

「E02」

Name	Parameter
A01	

イベントはそれぞれ独立して発生しますが、既に他のイベントでアクションが実行中の場合、後続のイベントは先のイベントの処理が終わるまで待機します。

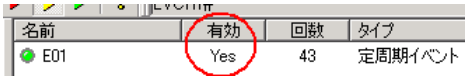
オンライン中のイベントの有効・無効の切り替え

システムがオンライン中であっても、システムプロパティ「E01!Enable」の操作によって、イベントの有効・無効状態を切り替えることができます。これにより、一定期間や一定条件でイベントを発生させることができます。詳しくは「システムプロパティとイベント」の章を参照して下さい。

■有効にする
//Event/E01!Enable = 1

■無効にする
//Event/E01!Enable = 0

イベントの現在の状態はリストの「有効」で確認できます。

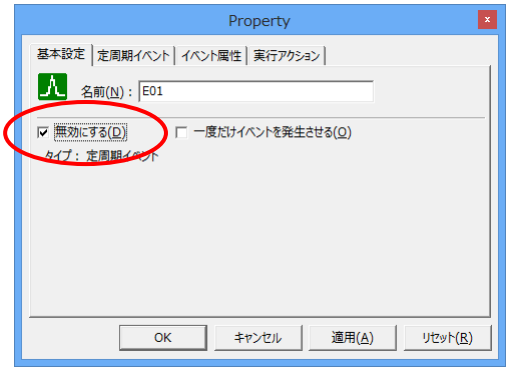


注意

タグイベントが複数登録されている場合、タグイベントは順次処理されます。
例えば、1 つ目のタグイベントで処理に時間がかかっている場合、2 つ目のタグイベントは 1 つ目のタグイベントの処理完了を待っている状態になります。そのため、実行タイミングが遅くなる場合がありえます。時間がかかる重たい処理などを行う場合は、注意してください。

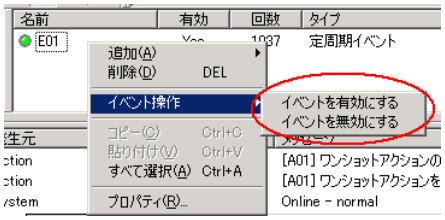
ヒント

イベントの初期状態を無効にするには、下記の「無効にする」をチェックします。

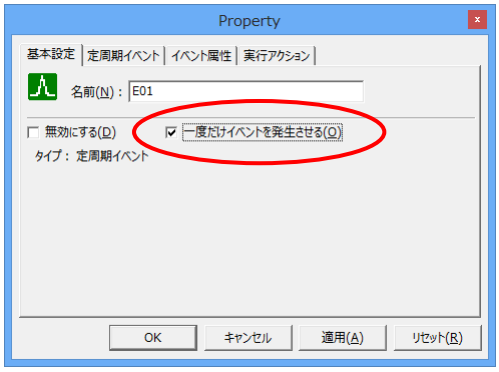


ヒント

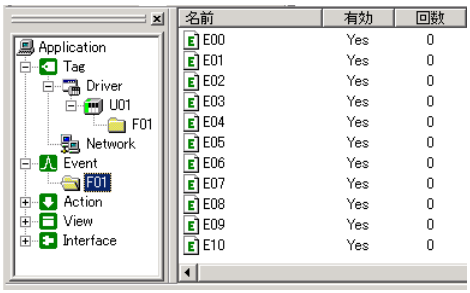
イベントの有効・無効状態は画面から手動で切り替えることもできます。イベント（「E01」など）をマウスで選択して右クリックして下記メニューを表示することで、有効・無効を切り替えることができます。

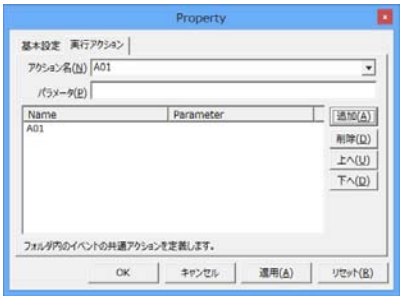


「一度だけイベントを発生させる」をチェックすると１度のみイベントが発生し、イベントが無効状態になります。再びイベントを有効にすると、再度１度イベントが発生します。



フォルダでイベントをまとめる





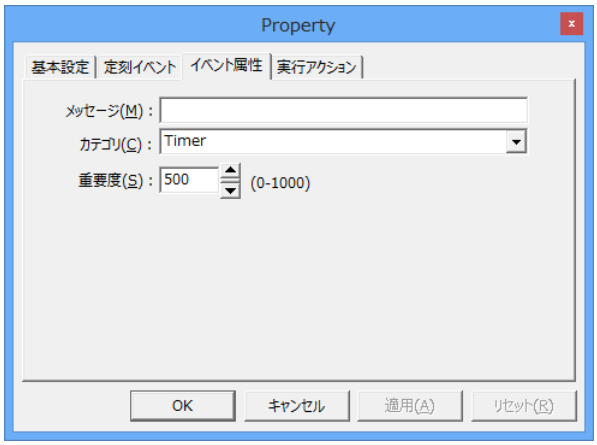
イベントアイコンの直下には、フォルダを作成することができます。

フォルダを作成する事によって、複数のイベントをまとめて管理することができるようになります。これにより、イベントの有効無効の切り替えや、イベントの強制発生を、フォルダ単位に一括で行うことができます。

また、フォルダのプロパティの「実行アクション」タブでアクションを定義すると、そのフォルダに含まれるイベントのうちどれか１つでもイベントが成立した時に共有的にアクションを実行することができます。もしも、フォルダとイベントの両方にアクションの実行が定義されていた場合には、まずイベントによってアクションが実行され、その後で親フォルダのアクションが実行されます。

イベント属性

イベント属性は「ロガーアクション」のみで利用される設定です。ログデータのフィールドに、メッセージ、カテゴリ、重要度などを含めて出力したい場合に設定します。



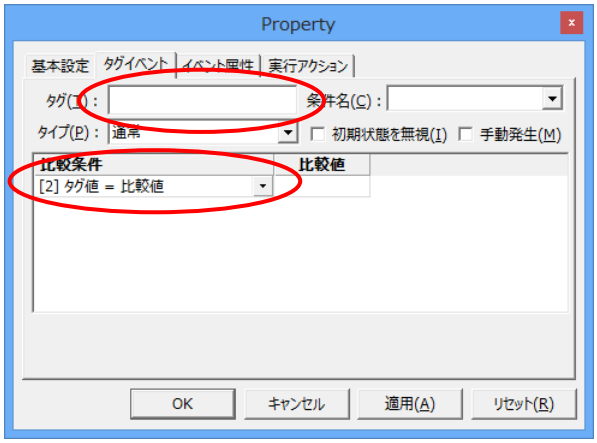
詳しくは「ロガーアクション詳細マニュアル」を参照ください。

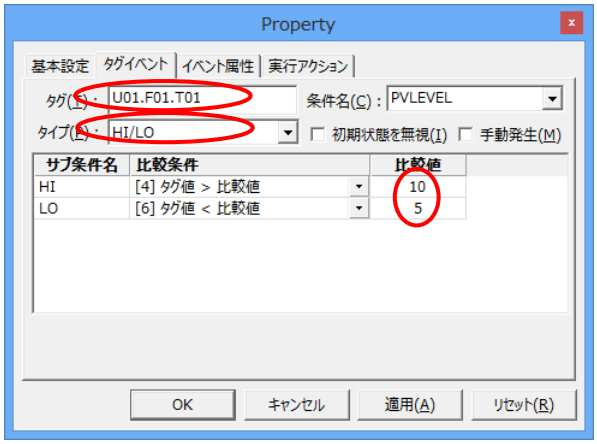
タグイベントの比較条件

タグイベントは様々な値の条件をもとにイベントを発生させることができます。

■単一条件の場合

タグの値が変化した、ビットが ON した、値が 100 以上になった等、１つの条件でイベントを発生させる場合は、下記の２箇所を設定します。まず、「タグ」の部分に、イベントを判定する元となるタグをタグパスで指定します（「U01.F01.T01」など）。そして、どのような条件によってイベントを発生させるのかを、「比較条件」に指定します。「タグの値が変化した」を選択した場合、タグが変化する度にイベントが発生するため、比較値を定義する必要はありません。その他の条件（「タグ値=比較値」など）を選択した場合には、比較値を設定して下さい。尚、ビットタグの場合、比較値には「TRUE/FALSE」もしくは「1/0」を設定することができます。





タグ名に「U01!Alive」などのシステムプロパティも設定できます。「Alive」プロパティは、PLC との通信が出来なくなったときにイベントを発生させたい場合に便利です。

■複数条件の場合

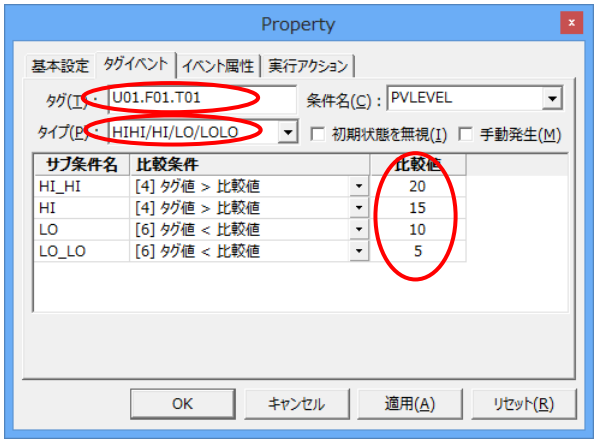
HI/LO 条件

HI/LO 条件でイベントを発生させる場合は、上の３箇所を設定します。必ず、LO 側より HI 側の比較値の方が大きな値となるように設定して下さい。

例）HI「10」LO「5」のとき

変化前の値	変化後の値	イベント
8	11	発生する
11	15	発生しない
15	11	発生しない
11	8	発生しない
8	6	発生しない
6	3	発生する
3	1	発生しない
1	11	発生する
11	1	発生する

HI_HI/HI/LO/LO_LO 条件



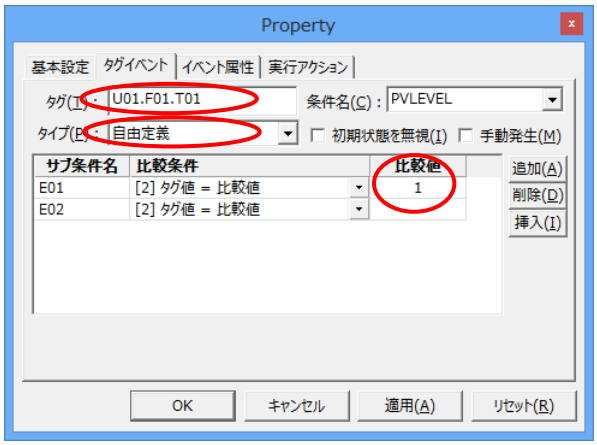
HI_HI/HI/LO/LO_LO 条件でイベントを発生させる場合は、上の３箇所を設定します。必ず、LO_LO から HI_HI になるにつれ比較値を大きい値にして下さい。

尚、右の例では、４→６及び 21→19 の変化でイベントが発生しています。これは、LOLO（または HIHI）からの復帰という意味のイベント発生ではなく、LOLO の後に LO が発生した（HIHI の後に HI が発生した）ことを表すイベントです。従って、LO から正常値への復帰（または HI からの復帰）については、イベントは発生しません。

例）HIHI「20」HI「15」LO「10」LOLO「5」のとき

変化前の値	変化後の値	イベント
1	4	発生しない
4	6	発生する
6	9	発生しない
9	11	発生しない
11	14	発生しない
14	16	発生する
16	19	発生しない
19	21	発生する
21	24	発生しない
24	21	発生しない
21	19	発生する
19	16	発生しない
16	14	発生しない
14	11	発生しない
11	9	発生する
9	6	発生しない
6	4	発生する
4	1	発生しない
1	21	発生する
21	1	発生する
1	16	発生する
16	9	発生する

自由定義



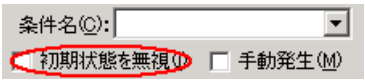
「=」「<>」と「>」「<」「>=」「<=」を混在させる場合は、「=」「<>」を最初の方の行に記述して下さい。また、「>」「<」「>=」「<=」に関しては、最初の方の行の比較値が一番大きく、最後の方の行が一番小さい値となるように定義して下さい（つまり、「HIHI/HII/LO/LOLO 条件」のように「>」「>=」が先になり、「<」「<=」は後になります）。

例）EQ1「タグ値=1」EQ2「タグ値=3」

変化前の値	変化後の値	イベント
0	1	発生する
1	2	発生しない
2	3	発生する
3	4	発生しない
3	1	発生する

タグの値がある範囲（例えば 50 以上 100 以内）になったときにイベントを発生させたい等、複数の条件を組み合わせてイベントの発生を定義したい場合には、スクリプト Ver1 アクションと組み合わせてイベントを定義するようにします。詳細は次頁の記述を参照して下さい。

■初期状態を無視



「初期状態を無視」をチェックすると、オンライン直後の値が条件を満たしていてもイベントを発生させません。いったん条件からはずれて、再び条件を満たしたときにイベントを発生させます。例えば、条件を「タグ値=比較値」で値「1」としたとき、オンライン直後の PLC の値が 1 になっていたとしてもイベントは発生しません。いったん 1 以外の値になった後、再び 1 になったときに発生します。

「初期状態を無視」はイベントのシステムプロパティ「Enable」を「1」にした直後の動作にも適用されます。

■手動発生



「手動発生」をチェックすると、値が条件を満たしていてもイベントを発生させません。ただし、イベントが発生した時の値は保持されます。この設定は、スクリプト「E01!Fire」でイベントの発生のタイミングを手動で制御したいときにチェックします。

「手動発生」は主にロガーアクションのイベント型ログで使用されます。それ以外の用途で使用される場面は少ないかもしれません。ロガーアクションでイベント手動発生によるロギングを行う方法は、「ロガーアクション詳細マニュアル」の、「イベント型ログフォーマットで定周期にログをとる」の記述を参照下さい。

■スクリプト Ver1 アクションとの組み合わせ

タグイベントでは定義しきれないような複雑な条件でイベントを発生させたい場合には、スクリプト Ver1 アクションとの組み合わせによって実現します。具体的には、判定結果を格納する判定用のタグを専用で 1 つ用意しておき、条件の判定をスクリプト Ver1 によって行い、判定結果を判定用タグに代入します。そのタグが TRUE の時にタグイベントを発生させるようにします。

例）「U01.F01.T02」が 50 以上 100 以下の範囲に入ったら

```
U01.F01.T01  =  U01.F01.T02 > 50 And U01.F01.T02 < 100
```

例）「U01.F01.T02」と「U01.F01.T03」の平均値が 100 を超えたら

```
U01.F01.T01  = (U01.F01.T02 + U01.F01.T03 ) / 2 > 100
```

この場合、条件比較を行うスクリプト Ver1 アクションの呼び出しは、それ用に作成した定周期イベント又は定刻イベントで定期的に計算を行うようにします。又は、計算の右辺で使用されているタグを元にタグイベントを作成し、各タグの値が変化した時にイベントが発生するようにしておき、評価に用いるタグが変化する都度スクリプト Ver1 が実行されるようにします。



通常、上記のような判定用タグは、仮想デバイスもしくはメモリデバイスで作成します。メモリデバイスの数値タグでは小数点を扱う事もできます。

４章　イベントの実践

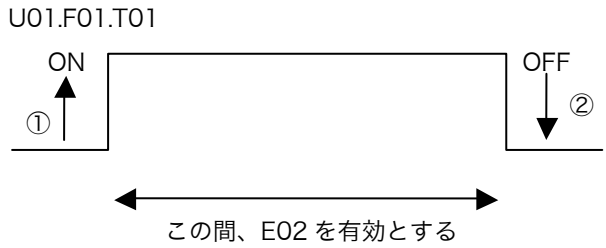
４－１　ビットが ON の間のみイベントを有効にする（Sample01）

概要

ここでは、ビットが ON の間のみイベントを有効にする設定方法について解説します。

このサンプルは以下に格納されています。
設定ファイル：　¥¥sample¥common¥event¥sample01¥event_sample01.txt

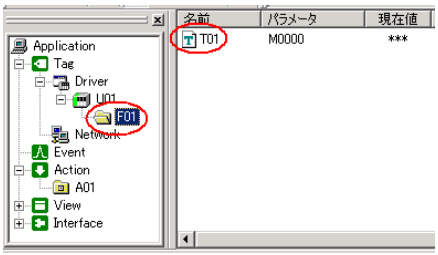
ここでは例として、「U01.F01.T01」が ON の間、イベント「E02」を有効にします。



設定手順

タグ設定

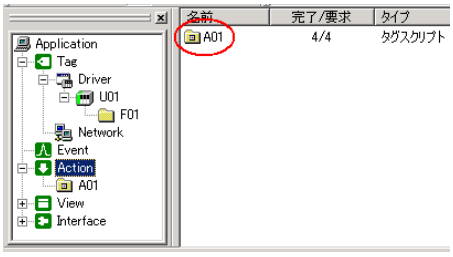
イベントを有効・無効にするトリガとなるビットタグを設定します。
ここでは、「U01.F01.T01」を設定します。



アクション設定

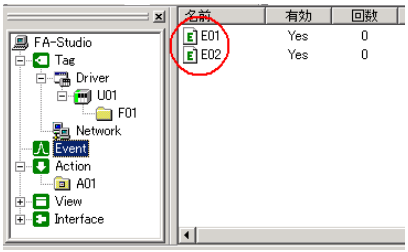
スクリプト Ver1 アクション「A01」を追加します。
スクリプトでは下記の処理を記述します。

```
//Event/E02!Enable = U01.F01.T01
```



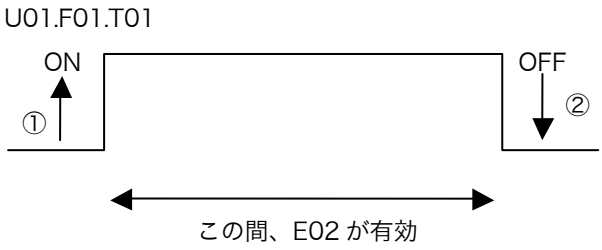
イベント設定

タグイベント「E01」及び定周期イベント「E02」を追加します。



E01：
・タグ「U01.F01.T01」
・比較条件「タグ値が変化した」
・実行アクション「A01」
E02:
今回切り替えのターゲットとなるイベントです。
初期状態を無効にするため、「無効にする」をチェックします。

解説



このサンプルを実行すると、以下のような動作となります。

① ビット ON 時
1．U01.F01.T01 が TRUE になるとタグイベント「E01」が発生し、スクリプト Ver1 アクション「A01」が実行される
2．「A01」では「//Event/E02!Enable = U01.F01.T01」の式が実行され、イベント「E02」が有効になる

② ビット OFF 時
1．U01.F01.T01 が FALSE になるとタグイベント「E01」が発生し、スクリプト Ver1 アクション「A01」が実行される
2．「A01」では「//Event/E02!Enable = U01.F01.T01」の式が実行され、イベント「E02」が無効になる

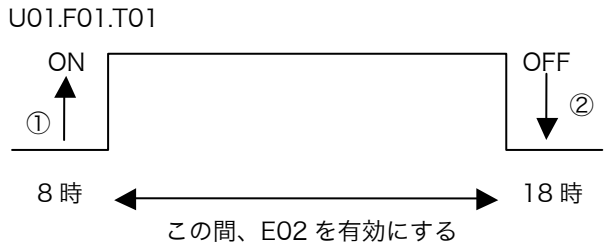
４－２　イベントを一定期間有効にする（Sample02）

概要

ここでは、イベントを一定の期間だけ有効にしたい場合の設定方法について解説します。例として、月曜から金曜日の、各 8 時 0 分 0 秒から 17 時 59 分 59 秒までの間のみイベントを有効にします。

このサンプルは以下に格納されています。
設定ファイル：　¥¥sample¥common¥event¥sample02¥event_sample02.txt

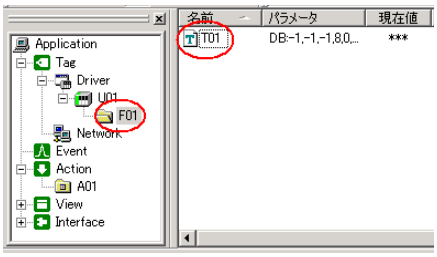
前述で解説した「Sample01」では、U01.F01.T01 が ON の間、イベント「E02」を有効にしました。この例では Sample01 の考え方を応用して設定を行います。



設定手順

タグ設定

イベントを有効・無効にするトリガとなるメモリデバイスタグ（時刻ビット）「U01.F01.T01」を設定します。



メモリデバイスの時刻ビットタグの設定を、8 時 0 分 0 秒から 17 時 59 分 59 秒の月曜から金曜日までの期間 ON になるように設定します。



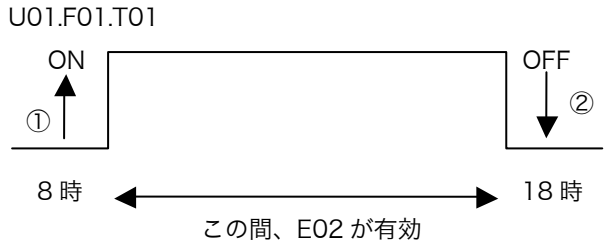
アクション設定

Sample01 と同じです。

イベント設定

Sample01 と同じです。

解説



メモリデバイスの時刻ビットタグは、きめられた時刻や期間のみ ON させることができるビットタグです。今回の設定では、月曜から金曜日までの期間で、８時から１８時までの間のみ ON させています。

① ビット ON 時
1．U01.F01.T01 が TRUE になるとタグイベント「E01」が発生し、スクリプト Ver1 アクション「A01」が実行される
2．「A01」では「//Event/E02!Enable = U01.F01.T01」の式が実行され、イベント「E02」が有効になる

② ビット OFF 時
1．U01.F01.T01 が FALSE になるとタグイベント「E01」が発生し、スクリプト Ver1 アクション「A01」が実行される
2．「A01」では「//Event/E02!Enable = U01.F01.T01」の式が実行され、イベント「E02」が無効になる

メモリデバイスに関する詳細は、「接続マニュアル」を参照してください。

４－３　イベントを条件により切り替える（Sample03）

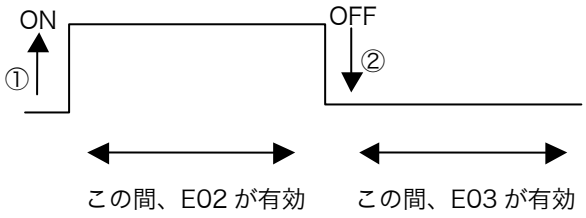
概要

ここでは、イベントそのものを条件に応じて切り替えたい場合の設定方法について解説します。例として、２つの定周期イベント（1 秒周期、１０秒周期）を用意し、「U01.F01.T01」の ON・OFF によりそれぞれのイベントを有効・無効に切り替えを行います。

このサンプルは以下に格納されています。
設定ファイル：　¥¥sample¥common¥event¥sample03¥event_sample03.txt

「U01.F01.T01」が ON の間、イベント「E02」を有効にし、OFF の間、イベント「E03」を有効にします。

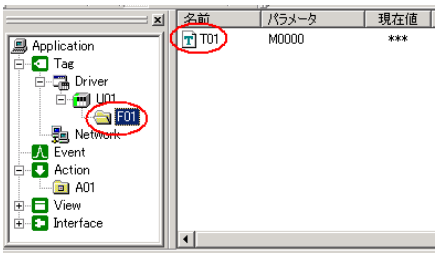
U01.F01.T01



設定手順

タグ設定

イベントを有効・無効にするトリガとなるビットタグを設定します。
ここでは、「U01.F01.T01」を設定します。

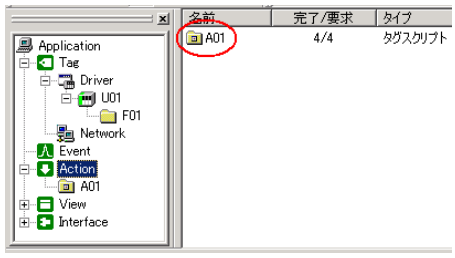


アクション設定

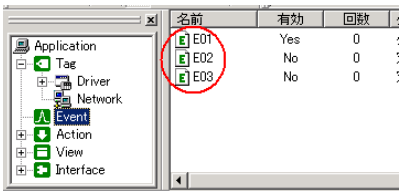
スクリプトアクション「A01」を追加し、スクリプトを下記のように設定する。

//Event/E02!Enable = U01.F01.T01
//Event/E03!Enable = Not U01.F01.T01

つまり、「U01.F01.T01」が TRUE 時 E02 を有効にし、FALSE の時 E03 を有効にします。



イベント設定



タグイベント「E01」及び定周期イベント「E02」「E03」を追加します。

E01：

・タグ「U01.F01.T01」

・比較条件「タグ値が変化した」

・実行アクション「A01」

E02:

・周期を１秒に設定

・初期状態を無効にするために「無効にする」をチェック

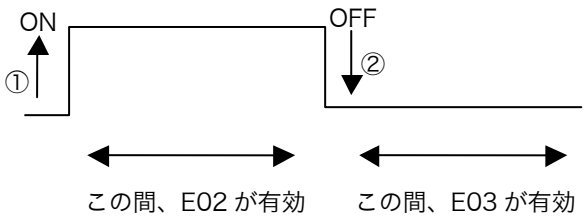
E03:

・周期を１０秒に設定

・初期状態を無効にするため、「無効にする」をチェック

解説

U01.F01.T01



① ビット ON 時

1．U01.F01.T01 が TRUE になるとタグイベント「E01」が発生し、スクリプト Ver1 アクション「A01」が実行される

2．「A01」では「//Event/E02!Enable = U01.F01.T01」の式が実行され、イベント「E02」が有効になる

3．同時に「//Event/E03!Enable = Not U01.F01.T01」の式が実行され、イベント「E03」が無効になる

② ビット OFF 時

1．U01.F01.T01 が FALSE になるとタグイベント「E01」が発生し、スクリプト Ver1 アクション「A01」が実行される

2．「A01」では「//Event/E02!Enable = U01.F01.T01」の式が実行され、イベント「E02」が無効になる

3．同時に「//Event/E03!Enable = Not U01.F01.T01」の式が実行され、イベント「E03」が有効になる

※スクリプト中の「Not」とは、TRUE/FALSE を反転します。

４－４　イベントを設定回数だけ実行する（Sample04）

概要

ここでは、イベントをある回数だけ発生させたい場合の設定方法について解説します。例として、１０回イベントが発生したらイベントを無効にします。

このサンプルは以下に格納されています。
設定ファイル：　¥¥sample¥common¥event¥sample04¥event_sample04.txt

「U01.F01.T01」が ON になると「E02」を有効にし、「A02」アクションを１０回実行した後に「E02」を無効にします。

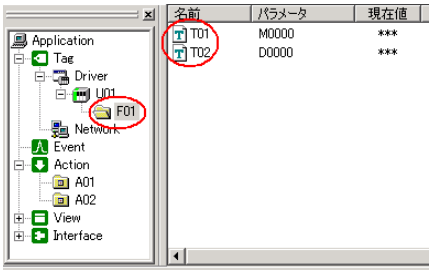


E02 が有効になった後、１０回イベントを発生させて無効にする。

設定手順

タグ設定

イベントを有効にするトリガとなるビットタグ「U01.F01.T01」を設定します。また、カウンタ用の数値タグ「U01.F01.T02」を設定します。

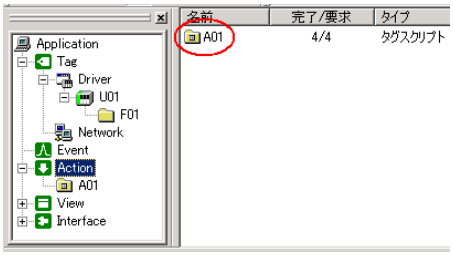


アクション設定

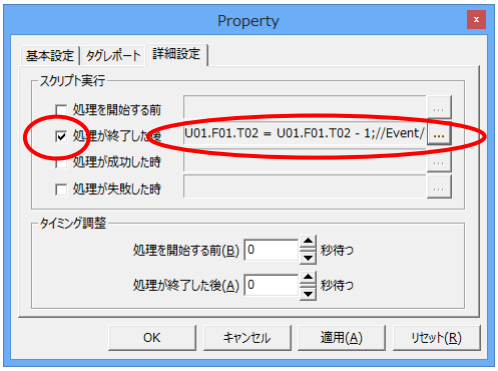
スクリプトアクション「A01」を追加し、スクリプトを下記のように設定する。

U01.F01.T02 = 10
//Event/E02!Enable = 1

ここで U01.F01.T02 に設定した値が実行回数になります。



指定回数分だけ実行したいアクション「A02」を追加し、「処理が終了したとき」のスクリプトを下記のように設定する

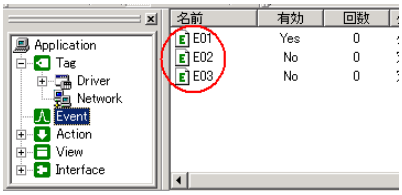


U01.F01.T02 = U01.F01.T02 - 1
//Event/E02!Enable = U01.F01.T02 > 0

※「A02」はどのアクションでも構いませんが、サンプルファイルではスクリプト Ver1 アクションを定義しています。

イベント設定

タグイベント「E01」及び定周期イベント「E02」を追加します。



E01：
・タグ「U01.F01.T01」
・比較条件「タグ値が=比較値」
・比較値「TRUE」
・実行アクション「A01」
E02:
・周期を 1 秒に設定
・初期状態を無効にするため、「無効にする」をチェック

解説



E02 が有効になった後、１０回イベントを発生させて無効にする。

このサンプルを実行すると以下のような動作となります。

① ビット ON 時
1．U01.F01.T01 が TRUE になるとタグイベント「E01」が発生し、スクリプト Ver1 アクション「A01」が実行される
2．「A01」では「U01.F01.T02 = 10」の式が実行され、10 が設定される
3．同時に「//Event/E02!Enable = 1」の式が実行され、イベント「E02」が有効になる

② 定周期イベント「E02」が有効中
1．「E02」により、ターゲットアクション「A02」が 1 秒ごとに実行される
2．処理が終了後、「U01.F01.T02 = U01.F01.T02 – 1」の式が実行され、タグ値は毎回 1 ずつ減算される
3．同時に「//Event/E02!Enable = U01.F01.T02 > 0」の式が実行される。「U01.F01.T02」が 0 より大きい時は特に変化無し。「U01.F01.T02」が 0 の時は、「E02」が無効になる

5章　システムプロパティとイベント

概略

システムプロパティとは各オブジェクト（イベント、フォルダ）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できます。

//Event/オブジェクトパス!プロパティ名

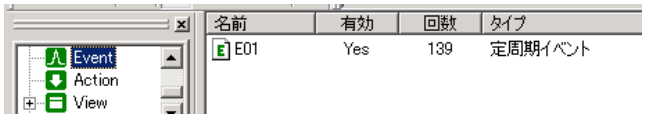
オブジェクトパスはフォルダとイベントを「.」ピリオドでつなげた文字列です。

例　//Event/E01（ルート直下に「E01」がある場合）
例　//Event/F01.E01（フォルダ「F01」内にイベント「E01」がある場合）
例　//Event/F01.F01.E01（フォルダ「F01」内にフォルダ「F01」があり、その中にイベント「E01」がある場合）

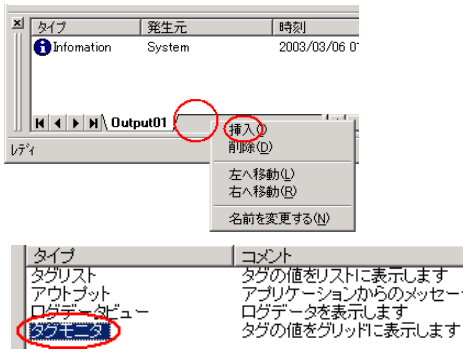
システムプロパティの使用例

例えば「Enable」プロパティはイベントの有効・無効を示します。

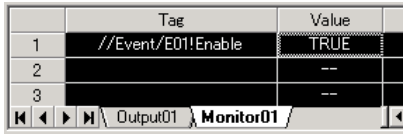
1．定周期イベント「E01」を作成する



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、「//Event/E01!Enable」をセルに入力する



4．//Event/E01!Enable の値を TRUE/FALSE と切り替えて、イベントの有効・無効が切り替わる事を確認する



システムプロパティ一覧

5－1 フォルダ	
Enable	フォルダ内の全イベントの有効・無効状態
Fire	フォルダ内の全イベントを一括で強制発生させる
5－2 イベント	
Enable	イベントの有効・無効状態
Fire	イベントを強制的に発生させる

5－1　フォルダ

F01!Enable	
	フォルダ内の全イベントの有効・無効状態
取得・設定	ビット型（True – 有効　False – 無効）

解説	有効・無効状態の切り替えを行います。フォルダ内の全てのイベントを一括で操作できます。
	
	例）
	スクリプト Ver1 アクションで、有効に切り替えるとき
	`//Event/F01!Enable = 1`
	
	同じく、無効に切り替えるとき
	`//Event/F01!Enable = 0`
	
	同じく、タグの値により有効・無効を切り替えるとき
	`//Event/F01!Enable = U01.F01.T01`

F01!Fire	
	フォルダ内の全イベントを強制発生させる
取得・設定	数値型

解説	フォルダ内の全てのイベントを強制的に発生させます。
	
	■設定時は下記の規則に従います。
	設定値が 0 もしくは FALSE の時、なにもしません。
	設定値が 0 以外　もしくは　TRUE の時強制発生します。
	
	例）
	スクリプト Ver1 アクションで強制発生させるとき
	`//Event/F01!Fire = 1`
	
	同じく、タグの値により強制発生させるとき
	`//Event/F01!Fire = U01.F01.T01`
	
	■取得時は強制的に発生させた回数を返します。10000 より大きくなると 0 に戻ります。

5－2　イベント

E01!Enable	
	イベントの有効・無効状態
取得・設定	ビット型（True – 有効　False – 無効）

解説	有効・無効状態を切り替えます。
	
	例）
	スクリプト Ver1 アクションで、有効に切り替えるとき
	`//Event/E01!Enable = 1`
	
	同じく、無効に切り替えるとき
	`//Event/E01!Enable = 0`
	
	同じく、タグの値により有効・無効を切り替えるとき
	`//Event/E01!Enable = U01.F01.T01`

E01!Fire	
	イベントを強制発生させる
取得・設定	数値型

解説	イベントを強制発生させます。
	
	■設定時は下記の規則に従います。
	設定値が 0 もしくは FALSE の時、なにもしません。
	設定値が 0 以外　もしくは　TRUE の時強制発生します。
	
	例）
	スクリプト Ver1 アクションで、強制発生させるとき
	`//Event/E01!Fire = 1`
	
	同じく、タグの値により強制発生させるとき
	`//Event/E01!Fire = U01.F01.T01`
	
	■取得時は強制的に発生させた回数を返します。10000 より大きくなると 0 に戻ります。

「イベントマニュアル」Copyright © 2010 Roboticsware, Inc. All rights reserved.
2010 年 6 月 1 日　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。