



New Generation GUI

FA-Panel6

for Graphic operation

— トレーニングテキスト —

FC01 機能別コース／アラーム

はじめに.....	3
Lesson 1 アラームとは.....	4
1-1 アラームについて.....	4
Lesson 2 標準プロジェクトのアラーム構築.....	13
2-1 標準プロジェクトのアラーム構築.....	13
2-2 アラームの設定について.....	29
Lesson 3 アラームのカスタマイズ.....	33
3-1 アラームにコメントを追加する.....	33
3-2 アラームメッセージにタグの値を含める.....	43
3-3 アラームのしきい値を画面から設定する.....	53
3-4 代表警報信号を作成する（クライアント編）.....	62
3-5 代表警報信号を作成する（サーバ編）.....	70
3-6 復帰条件、発生・復帰待ちタイマーの利用.....	76
3-7 画面操作のイベントを記録する.....	80
3-8 アラームメールを送信する.....	92
3-9 データベースを利用したアラーム構築.....	104
Lesson 4 アラームの詳細手順.....	115
4-1 アラーム機能の手動組み込み.....	115
4-2 独自プロジェクトに警報音を実装する.....	131

はじめに

本トレーニングコースについて

本トレーニングコース「機能別コース／アラーム」は、FA-Panel6 のアラーム監視について学ぶためのトレーニングコースです。

このコースでは、FA-Panel6 が標準機能として提供する画面ライブラリに含まれている「アラームサマリ」「アラーム履歴」を使用し、アラーム監視を行うための各種設定や構築手順について学習します。また、同じく画面ライブラリに含まれている「イベント履歴」「イベントリアルタイム」を使用したイベント履歴機能（運転操作履歴）の構築についても取り上げます。

尚、スピード開発スタイルによる「標準フレームワーク」のウィザード機能によって生成されたプロジェクトについても、本トレーニングで解説する手順によってアラーム機能を構築することができます。

本トレーニングでは以下の内容について学習します。

- Lesson1 アラームとは
- Lesson2 標準プロジェクトのアラーム構築
- Lesson3 アラームのカスタマイズ
- Lesson4 アラームの詳細手順

■その他のトレーニングコースについて

本トレーニングコース以外にも、以下のトレーニングコースが用意されています。ロギング、トレンドグラフ、日報集計などの個別機能の詳細については、それぞれの機能別コースのトレーニングコンテンツで別途学習することができます。

トレーニングコース	概要
TC01 入門コース	FA-Panel の入門編トレーニングです。 (前提条件：なし)
TC02 スピード開発コース	スピード開発コースに関するトレーニングです。 (前提条件：TC01 を修了していること)
TC03 カスタム開発コース	カスタム開発コースに関するトレーニングです。 (前提条件：TC01、TC02 を修了していること)
FC01 機能別コース／アラーム	本トレーニングです。 (前提条件：TC01、TC02 を修了していること)
FC02 機能別コース／ロギング・トレンド	ロギング、トレンドグラフに関するトレーニングです。 (前提条件：TC01、TC02 を修了していること)
FC03 機能別コース／日報	日報に関するトレーニングです。 (前提条件：TC01、TC02、FC02 を修了していること)

Ver6.0 2nd Edition

アラームサマリ／アラーム履歴とは

アラームとは、監視対象の設備で故障が発生した際や、温度や圧力、水量などの数値が平常時と異なる値になった場合などに、監視対象設備で異常が発生したという事をオペレータに対して伝えるための警報監視機能の総称です。

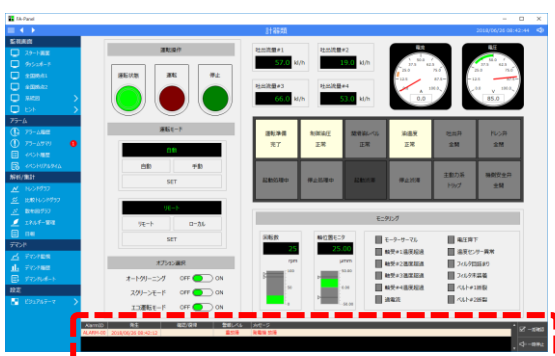
一般的に SCADA システムでは、現在発生しているアラームを画面上に表示する機能、警報発生時に警報音を鳴動させる機能、そして、発生したアラームの履歴を検索して表示するための機能が求められます。

以下に、具体的な例を挙げます。



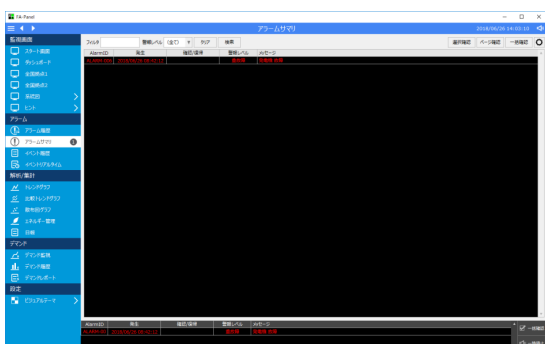
平常時の監視画面では、オペレータは様々なグラフィック画面やトレンドグラフ、レポート画面などを目的に応じて表示して、操作対象の機器の運転操作や、設備の運転状況のチェックなどを行います。

このように、平常時はアラームを意識することはありません。



アラームが発生すると、画面下部に現在発生中の警報が点滅表示されるとともに、警報音が鳴動し、アラームが発生したことを即座に伝えます。

ここで、現在発生中のアラームを表示する仕組みのことを「アラームサマリ」と呼びます。



画面左のメニューからアラームサマリを選択すると、メインエリアにアラームサマリ画面を表示することができます。この画面から、アラームの発生状況の把握や、アラームに対する確認などの各種操作を行うことができます。



画面左のメニューからアラーム履歴を選択すると、メインエリアにアラーム履歴画面が表示されます。この画面から、過去に発生したアラームの履歴情報を調べることができます。

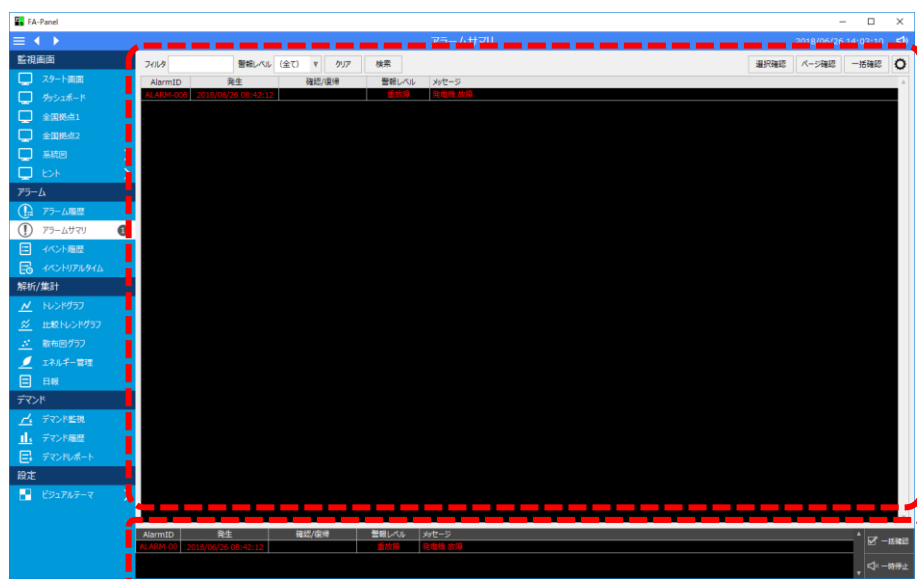
アラームサマリ画面

現在発生中のアラームを一覧形式で表示する機能のことを「アラームサマリ」と呼びます。

アラームサマリでは、以下の情報を知ることができます。

- 何のアラームが、いつ発生したのか？
- 現在発生中のアラームは何か？（まだ発生中か、すでに復帰したか）
- 発生したアラームの重要度（警報レベル）

アラームサマリは、画面下部の共通エリアと、メインエリアのアラームサマリ画面が用意されています。共通エリアは全ての画面上に共通的に表示されるため、アラーム以外の画面を表示している状況でも、オペレータは警報の発生を見逃すことはありません。



アラームが発生すると、画面下部のアラームサマリは以下のような表示になります。

AlarmID	発生	確認/復帰	警報レベル	メッセージ	
ALARM-00	2018/06/26 14:37:13		番故障	52S 不足電圧	<input checked="" type="checkbox"/> 一括確認 <input type="checkbox"/> 一時停止

上記では、「メッセージ」の欄にアラームのメッセージ（警報の内容）を表示するとともに、「発生」の欄にアラームの発生日時を表示しています。又、「警報レベル」には発生したアラームの重要度が表示されています。そして、「確認/復帰」の欄は空白となっており、対象のアラーム表示の行を点滅表示することによって、このアラームが現在も発生中である事がわかります。

尚、アラームには、「発生」と「復帰」という考え方があり、アラームの検出条件が成立した時を「発生」とし、検出条件が成立しなくなった時を「復帰」とみなします。アラームサマリや履歴では、発生、あるいは復帰を検出した日時が記録されます。

■アラームの確認とは

監視対象の設備上で何らかのアラームが発生した事によって設備が停止してしまった場合には、停止の原因となったアラームを特定する必要があります。しかし、アラームの中には、一瞬だけ発生してすぐに復帰してしまうものもあります。そのような種類のアラームの場合、アラームの復帰によりアラームサマリの表示から即座に消えてしまうと、オペレータは何が起きたのかを知ることができません。

そこで、発生したアラームに対し、オペレータが確認操作を行うまではアラームサマリに表示された状態のまま保持しておく機能が必要となります。この機能の事を、「アラームの確認」と呼びます。FA-Panel のアラーム機能では、各アラームに対して「確認あり」なのかどうかを個別に定義する事ができるようになっています。

例えば、「確認あり」として定義されたアラームが未確認の状態で復帰すると、アラームサマリは以下のように、「確認／復帰」の欄にアラームの復帰日時が表示され、表示色は水色で点滅表示された状態になります。

AlarmID	発生	確認/復帰	警報レベル	メッセージ	
ALARM-00	2018/06/26 14:37:13	2018/06/26 14:59:01	重故障	52S 不足電圧	<input checked="" type="checkbox"/> 一括確認
					<input type="checkbox"/> 一時停止

そして、画面下部の「一括確認」ボタンをクリックすると、復帰かつ未確認だったアラームは確認済となり、アラームサマリのリストから消去します。

AlarmID	発生	確認/復帰	警報レベル	メッセージ	
					<input checked="" type="checkbox"/> 一括確認
					<input type="checkbox"/> 一時停止

一方、「確認あり」のアラームが発生中の状態で（まだ復帰していない状態で）、画面下部の「一括確認」ボタンをクリックすると、「確認／復帰」の欄に確認日時が表示されるとともに、表示色は点滅から点灯に切り替わります。

AlarmID	発生	確認/復帰	警報レベル	メッセージ	
ALARM-00	2018/06/26 15:08:01	2018/06/26 15:08:11	重故障	52S 不足電圧	<input checked="" type="checkbox"/> 一括確認
					<input type="checkbox"/> 一時停止

その後、アラームが復帰すると、既に確認済のアラームが復帰したため、アラームサマリのリストから自動的に消去します。

AlarmID	発生	確認/復帰	警報レベル	メッセージ	
					<input checked="" type="checkbox"/> 一括確認
					<input type="checkbox"/> 一時停止

アラーム履歴画面

SCADA システムのアラーム監視に求められる機能の一つに、アラーム履歴機能があります。アラーム履歴とは、アラームの発生、復帰、確認などの過去の履歴情報を一覧形式で表示するための機能です。例えば、複数のアラームが発生した際などに、アラーム履歴から対象のアラームの発生順序を調べることによって、どのアラームが原因となって他のアラームが連鎖的に発生したのかを知る事ができます。あるいは、目的の機器で過去にどのようなアラームが発生したのかを検索して調べることもできます。

FA-Panel のアラーム履歴では、アラームのメッセージ、警報レベルのほか、発生、復帰、確認の各日時を時系列で表示することが可能であり、対象のアラームがいつ発生し、復帰、あるいは確認が行われたのかを知ることができます。また、過去に発生したアラームの履歴情報の中から、フィルタに任意の文字列を指定した絞り込みや、日付による範囲を指定した検索を行うことができます。

例えば、以下のような履歴が表示されている状態で、メッセージに「52F3」を含むアラームのみに絞り込みたい場合、フィルタに「52F3」と入力して検索を行います。

フィルタ	警報レベル	(全て)	クリア	検索
AlarmID	日時	タイプ	警報レベル	メッセージ		
ALARM-005	2018/06/26 17:47:43	確認	中故障	52F3 過電流		
ALARM-004	2018/06/26 17:47:43	確認	中故障	52F2 過電流		
ALARM-003	2018/06/26 17:47:43	確認	中故障	52F1 過電流		
ALARM-007	2018/06/26 17:47:43	確認	軽故障	LBSTリップ		
ALARM-008	2018/06/26 17:47:43	確認	軽故障	LBSTリップ		
ALARM-006	2018/06/26 17:47:43	確認	重故障	発電機 故障		
ALARM-002	2018/06/26 17:47:43	確認	重故障	52S 不足電圧		
ALARM-001	2018/06/26 17:47:43	確認	重故障	52S 過電流		
ALARM-005	2018/06/26 17:47:39	復帰	中故障	52F3 過電流		
ALARM-004	2018/06/26 17:47:39	復帰	中故障	52F2 過電流		
ALARM-003	2018/06/26 17:47:39	復帰	中故障	52F1 過電流		
ALARM-007	2018/06/26 17:47:39	復帰	軽故障	LBSTリップ		
ALARM-008	2018/06/26 17:47:39	復帰	軽故障	LBSTリップ		
ALARM-006	2018/06/26 17:47:39	復帰	重故障	発電機 故障		

すると、リストの内容は、メッセージに「52F3」を含むアラームの履歴のみに絞り込まれました。

フィルタ	警報レベル	(全て)	クリア	検索
52F3						
AlarmID	日時	タイプ	警報レベル	メッセージ		
ALARM-005	2018/06/26 17:47:43	確認	中故障	52F3 過電流		
ALARM-005	2018/06/26 17:47:39	復帰	中故障	52F3 過電流		
ALARM-005	2018/06/26 17:47:32	発生	中故障	52F3 過電流		

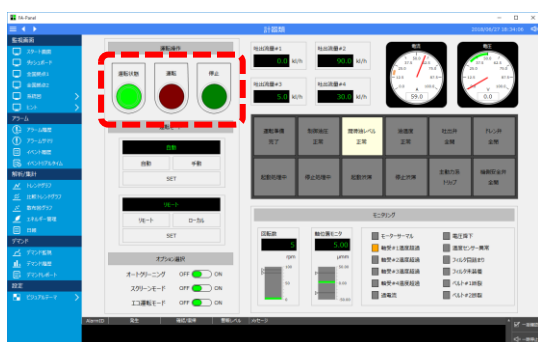
その他、日付による範囲選択や、警報レベルによる絞り込みも可能です。

イベント履歴／イベントリアルタイムとは

アラームの機能の一部に、「イベント」があります。イベントとは、設備の運転操作などの履歴を記録するための機能です。

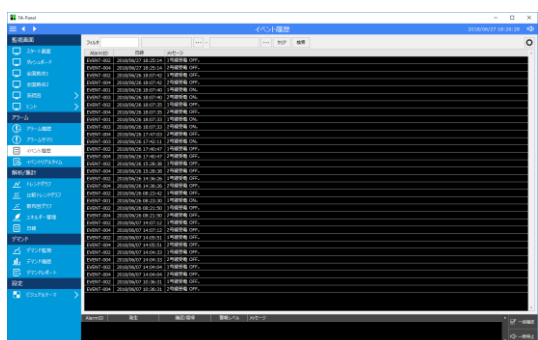
一般的に SCADA システムでは、監視対象の機器の運転／停止などの運転操作を履歴として記録するとともに、過去の履歴を画面に表示するための機能が求められます。FA-Panel には、機器の運転／停止の履歴を表示するための機能として、「イベント履歴」と「イベントリアルタイム」が用意されています。

以下に、具体的な例を挙げます。



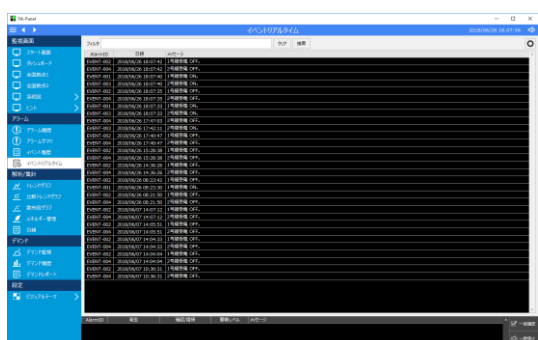
通常、オペレータはグラフィック画面を介して、操作対象の機器の運転操作や、設備の運転状況のチェックなどを行います。

設備の運転／停止などの現在のステータスについては、監視画面上のランプ表示などから知ることができます。一方で、対象の設備が過去にどのように運転されたのかを、履歴として調べたい場合があります。



そのような場面で、イベント履歴を使用します。イベント履歴では、あらかじめ登録されている機器の運転状況を履歴として閲覧することができます。

尚、イベント履歴画面は動的な表示更新が行われなため、最新の情報を表示するには「更新」ボタンをクリックする必要があります。



そこで、イベントリアルタイム画面が用意されています。

イベントリアルタイム画面では、最新のイベント情報で自動的に表示更新される仕組みになっています。

機器の動作の変化を画面上でリアルタイムに見たい場合に、この画面を使用します。

イベント履歴画面

イベント履歴とは、イベントの過去の履歴情報を一覧形式で表示するための機能です。FA-Panel のイベント履歴では、過去に発生したイベントの履歴情報の中から、フィルタに任意の文字列を指定した絞り込みや、日付による範囲を指定した検索を行うことができます。

例えば、以下のような履歴が表示されている状態で、メッセージに「2 号線」を含むイベントのみに絞り込みたい場合、フィルタに「2 号線」と入力して検索を行います。

フィルタ

...

-

...

クリア

検索

AlarmID	日時	メッセージ
EVENT-002	2018/06/26 18:07:42	1号線受電 OFF。
EVENT-004	2018/06/26 18:07:42	2号線受電 OFF。
EVENT-001	2018/06/26 18:07:40	1号線受電 ON。
EVENT-003	2018/06/26 18:07:40	2号線受電 ON。
EVENT-002	2018/06/26 18:07:35	1号線受電 OFF。
EVENT-004	2018/06/26 18:07:35	2号線受電 OFF。
EVENT-001	2018/06/26 18:07:33	1号線受電 ON。
EVENT-003	2018/06/26 18:07:33	2号線受電 ON。
EVENT-004	2018/06/26 17:47:03	2号線受電 OFF。
EVENT-003	2018/06/26 17:42:11	2号線受電 ON。
EVENT-002	2018/06/26 17:40:47	1号線受電 OFF。

すると、リストの内容は、メッセージに「2 号線」を含むイベントの履歴のみに絞り込まれます。

フィルタ

2号線

...

-

...

クリア

検索

AlarmID	日時	メッセージ
EVENT-004	2018/06/26 18:07:42	2号線受電 OFF。
EVENT-003	2018/06/26 18:07:40	2号線受電 ON。
EVENT-004	2018/06/26 18:07:35	2号線受電 OFF。
EVENT-003	2018/06/26 18:07:33	2号線受電 ON。
EVENT-004	2018/06/26 17:47:03	2号線受電 OFF。
EVENT-003	2018/06/26 17:42:11	2号線受電 ON。

イベントリアルタイム画面

機器の運転・停止などのイベントを監視する場合、機器が運転状態になった、あるいは停止した等の状態の変化を、画面上に動的に表示したい場合があります。このようなニーズでは、画面の表示更新ボタンなどをクリックしなくても自動的に最新の情報が表示される必要があります。FA-Panel のイベントリアルタイム表示では、登録されているイベントの発生状況をリアルタイムに表示することができます。

フィルタ

クリア

検索

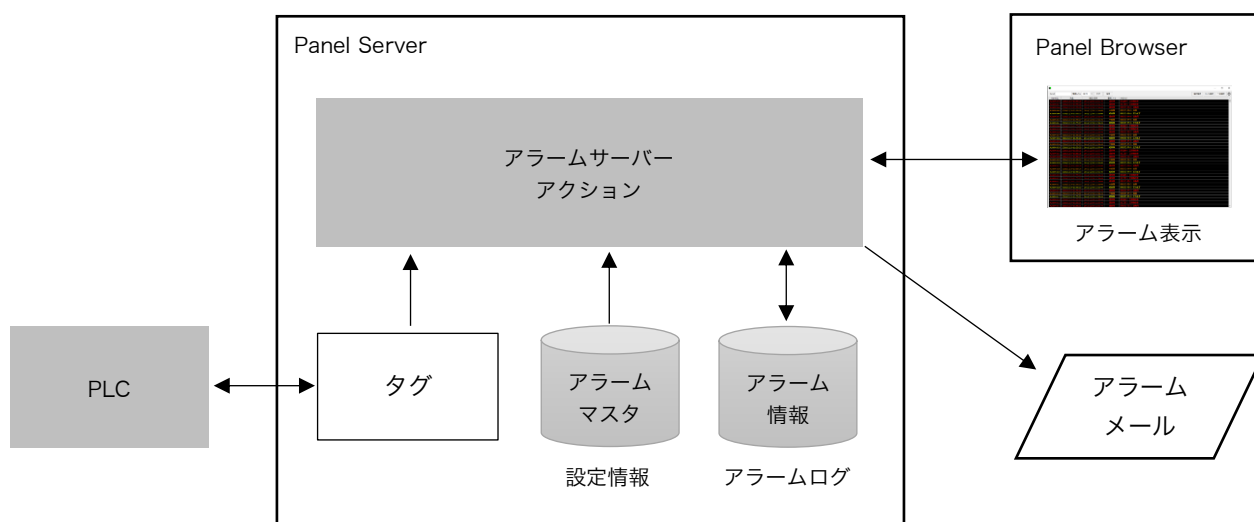
AlarmID	日時	メッセージ
EVENT-005	2018/06/27 18:49:02	主系マシン 運転
EVENT-006	2018/06/27 18:49:00	主系マシン 停止
EVENT-005	2018/06/27 18:48:55	主系マシン 運転
EVENT-006	2018/06/27 18:34:02	主系マシン 停止
EVENT-005	2018/06/27 18:33:48	主系マシン 運転

尚、イベントリアルタイム画面についても、フィルタによる絞り込みに対応しています。

FA-Panel のアラームの機能構成

FA-Panel のアラーム機能は、Panel Server 側の「アラームサーバーアクション」と、Panel Browser 側の画面ライブラリで構成されています。

アラーム情報の管理・監視は Panel Server の「アラームサーバーアクション」によって行います。Panel Server が検出したアラーム情報は Panel Server 配下のディスク上にアラームログとして保存されます。そして、クライアントとなる Panel Browser 側では、アラーム機能用に用意された画面ライブラリである「アラームサマリ」「アラーム履歴」「イベントリアルタイム」「イベント履歴」を使用して画面表示を行います。さらに、アラームサーバーアクションには、アラームの検出をトリガとしてメールを自動送信する機能も用意されています。



FA-Panel でアラーム機能を構築する際には、プロジェクトにアラームの画面ライブラリを組み込み、アラームサーバーアクションの設定を行う必要があります。また、アラームサーバの主な設定作業としては、アラームマスタに対してアラームの発生条件やアラームメッセージなどを登録する必要があります。

アラームサーバーアクションについて

アラーム機能を使用するには、Panel Server にアラームサーバーアクションの登録・設定を行う必要があります。

アラームサーバーアクションはタグの値をリアルタイムに監視し、あらかじめ「アラームマスタ」と呼ばれる CSV 形式の定義ファイルに定義されたアラーム発生条件（タグの値に対するアラームの発生条件）に従ってアラームの発生／復帰などの状態監視を行います。検出されたアラーム情報は Panel Server 側で履歴情報として記録されるとともに、接続されている Panel Browser に対してアラーム状態の通知が行われます。

アラームサーバーアクションには以下の 2 種類があります。

アクション名	概要
アラームサーバー	アラームの履歴情報をテキストファイル（CSV 形式）で保存します。 データベースを使用しない、最もシンプルな構成です。 本トレーニングコースでは、こちらのアラームサーバを主に使用します。
アラームサーバー（ODBC）	アラームの履歴情報を ODBC ドライバ経由でデータベース上のテーブルに保存することができます。 DBMS の種類としては、SQLServer、Oracle、MySQL、PostgreSQL に対応しています（※対応可能なデータベースのバージョン等、最新情報についてはオンラインマニュアルを参照してください）。 尚、アラームサーバー（ODBC）を利用したアラームの構築については、本トレーニング中のカスタマイズ事例として紹介します。

尚、本テキストでは、「アラームサーバー」と「アラームサーバー（ODBC）」を明確に区別したい場面などで、「アラームサーバー（CSV）」と表記する場合があります。



Panel Server には、同時に複数のアラームサーバーアクションを登録して並列的に動作させることもできます。

Lesson 2 標準プロジェクトのアラーム構築

2-1 標準プロジェクトのアラーム構築



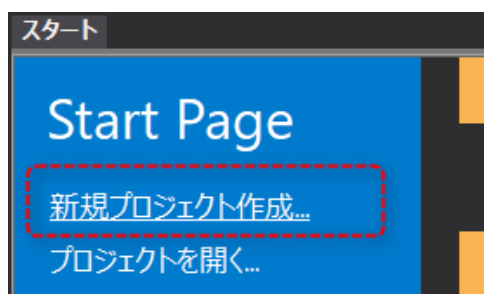
このセクションでは、標準フレームワークのウィザード機能を利用して、アラーム機能を含むプロジェクトを実際に作成してみます。

標準プロジェクトを作成する

それでは、実際に標準プロジェクトを作成してみましょう。

1. スタートページから、「新規プロジェクト作成」を選択する

Panel Editor を起動し、スタートページから「新規プロジェクト作成」をクリックしてください。

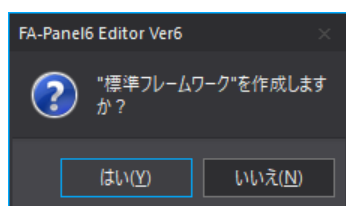


2. 標準フレームワークを選択し「作成」を開く

スタートページに表示された一覧の中から「標準フレームワーク」をクリックしてください。
画面の下部に以下のような説明とボタンが表示されるので、「作成」ボタンをクリックしてください。

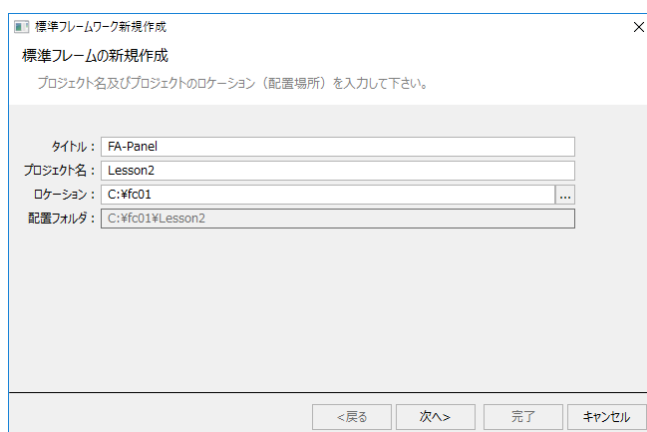


すると、以下のダイアログが表示されるので、「はい」をクリックします。



3. プロジェクトの基本情報を入力する

以下のダイアログが表示されるので、プロジェクトの作成に必要な情報を入力します。



標準フレームワーク新規作成

標準フレームワークの新規作成

プロジェクト名及びプロジェクトのロケーション（配置場所）を入力して下さい。

タイトル: FA-Panel

プロジェクト名: Lesson2

ロケーション: C:\fc01

配置フォルダ: C:\fc01\Lesson2

<戻る 次へ> 完了 キャンセル

■タイトル

プロジェクトのタイトルを指定します。ここでは、「FA-Panel」としておきます。
タイトルは後で変更することができます。

■プロジェクト名

プロジェクトの名前を指定します。ここでは、「Lesson2」としておきます。

■ロケーション

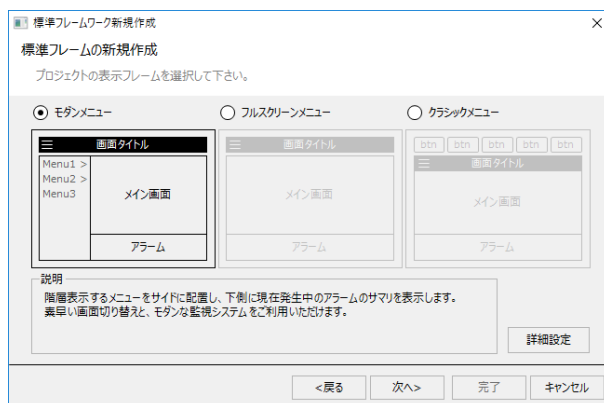
プロジェクトフォルダを作成するフォルダの場所を指定します。実際に存在する任意のフォルダを選択してください。
上記は、「C:\fc01」フォルダを指定した例です。

上記を入力すると、配置フォルダの欄に実際のフォルダのパス名が表示されます。
実際には、ロケーションで指定したフォルダの配下に、プロジェクト名のフォルダが作成されます。

必要な事項を入力したら、「次へ」ボタンをクリックしてください。

4. プロジェクトの表示フレームを選択する

標準フレームワークにはいくつかの表示様式が用意されています。ここではデフォルトの「モダンメニュー」を選択し、「次へ」ボタンをクリックしてください。



標準フレームワーク新規作成

標準フレームワークの新規作成

プロジェクトの表示フレームを選択して下さい。

☒ モダンメニュー ☐ フルスクリーンメニュー ☐ クラシックメニュー

モダンメニュー: 画面タイトル、Menu1 > Menu2 > Menu3、メイン画面、アラーム

フルスクリーンメニュー: 画面タイトル、メイン画面、アラーム

クラシックメニュー: 画面タイトル、btn btn btn btn btn、メイン画面、アラーム

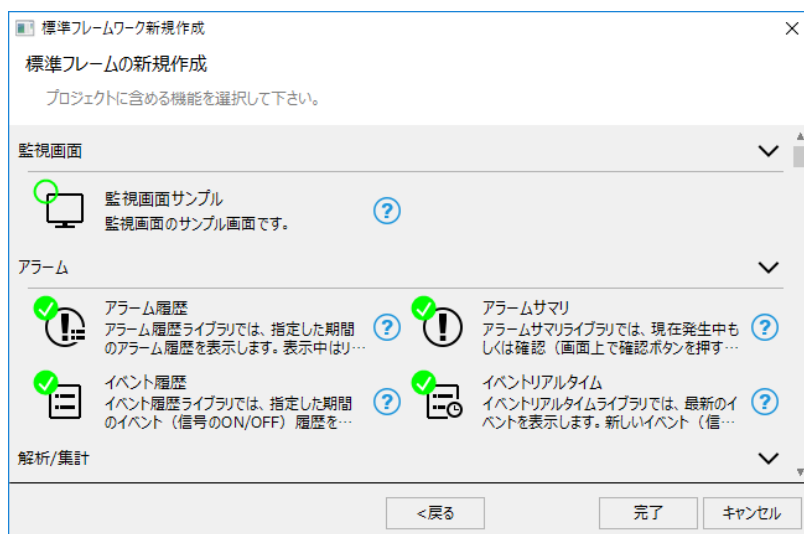
説明: 階層表示するメニューをサイドに配置し、下側に現在発生中のアラームのサマリを表示します。最も早い画面切り替えと、モダンな監視システムをご利用いただけます。

詳細設定

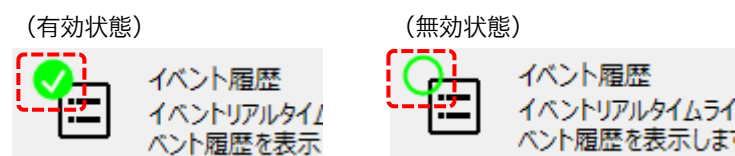
<戻る 次へ> 完了 キャンセル

5. プロジェクトに含める標準機能を選択する

続いて、プロジェクトに含める機能の選択を行います。



各機能に表示されているチェックマークをクリックすることで、プロジェクトに含めるか否かを切りかえることができます。



デフォルトでは、全ての標準機能が有効になっています。

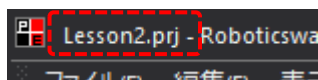
今回は例として、アラーム関連の4機能とビジュアルテーマにのみチェックを入れ、他の機能は無効にしてください。

機能	有効
監視画面サンプル	—
アラーム履歴	○
アラームサマリ	○
イベント履歴	○
イベントリアルタイム	○
トレンドグラフ	—
比較トレンドグラフ	—
散布図グラフ	—
エネルギー管理	—
日報	—
デマンド監視	—
デマンド履歴	—
デマンドレポート	—
ビジュアルテーマ	○

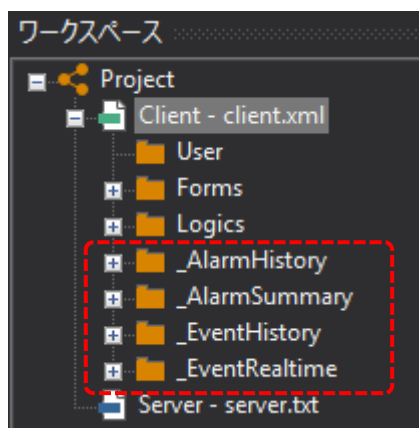
設定が終わったら、「完了」ボタンをクリックしてください。

6. プロジェクトが作成されました

プロジェクトが正しく作成されると、Panel Editor のタイトル部分に、ウィザードで指定したプロジェクト名が表示された状態になります。



また、ワークスペースのツリー表示を見ると、アラーム関連画面のフォルダが含まれていることが確認できます。

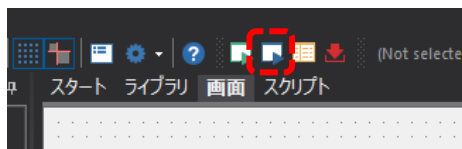


タグの登録

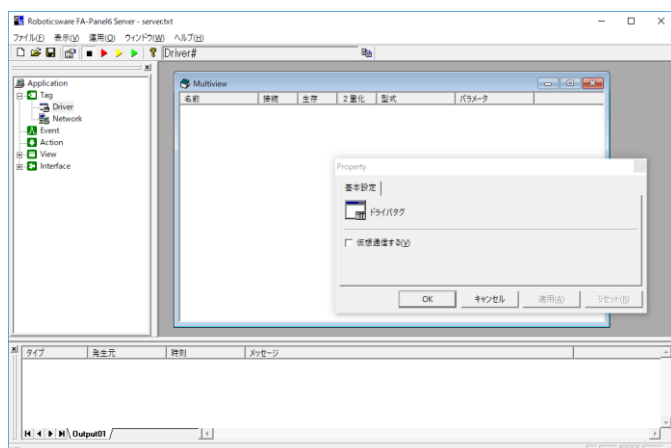
続いて、アラーム監視対象とするタグを登録します。
今回は例として、仮想デバイスユニットを登録します。

1. Panel Server を起動する

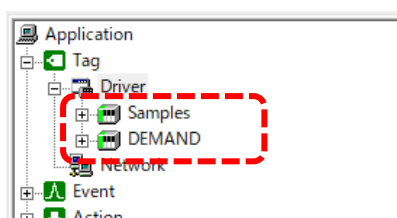
Panel Editor のサーバアイコンをクリックし、Panel Server を起動してください。



Panel Server が起動しました。



Panel Server の左ツリーから、Tag/Driver を開いてください。



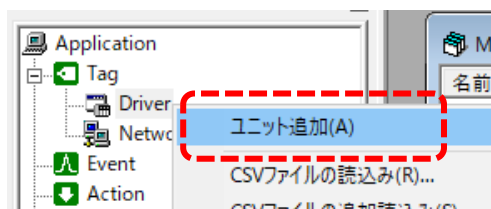
「Samples」ユニットには、標準フレームワークにあらかじめ含まれているサンプルアプリケーション（グラフィック画面、アラーム監視など）が参照するタグが格納されています。

また、「DEMAND」フォルダには、デマンド監視機能が参照するタグが格納されています。

今回はこれらのサンプルタグは利用しませんが、現時点ではそのままにしておいて問題ありません。

2. ユニットを追加する

ツリー選択ビューから、「Tag」の「Driver」を右クリックしてメニューを表示し、表示されたメニューから「ユニット追加」を選択してください。

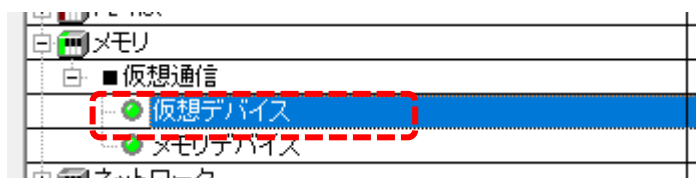


すると、ドライバー一覧画面が表示されます。



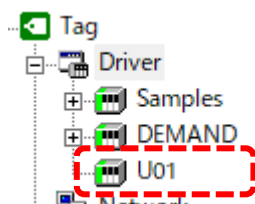
今回は、実際の PLC の代わりとして「仮想デバイス」を追加する事とします。

ツリーから「仮想デバイス」を選択し、OK ボタンをクリックしてください。



メモリ/仮想通信/仮想デバイス

すると、Tag/Driver の配下に「U01」ユニットが追加されました。

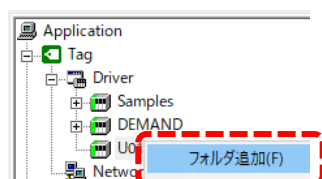


「U01」というユニット名はそのまま開発に使う事ができます。

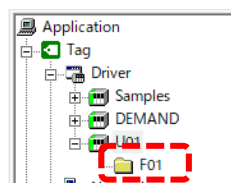
本トレーニングではこの名称のまま作業を進めます。

3. フォルダを追加する

ツリー選択ビューから、「U01」ユニットを右クリックしてメニューを表示し、表示されたメニューから「フォルダ追加」を選択してください。



フォルダ「F01」が追加されました。フォルダ名は変更することもできますが、ここでは F01 のまま使用します。



4. タグを一括登録する

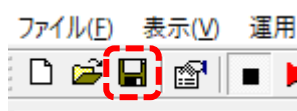
U01.F01 フォルダの下に、以下の条件でタグを登録してください（※タグの連続追加を使用すると便利です）。

登録するタグ：

D0000～D0099 の数値型タグ（100 点）

M0000～M0099 のビット型タグ（100 点）

タグの登録が終わったら、Panel Server の設定を保存しておいてください（※設定の保存は都度行ってください）。



以降の手順では、上記で登録した仮想デバイスのタグをアラームとイベントの監視対象としてアラームマスタに登録し、アラームの構築を行います。

アラームマスタの編集 (Alarm)

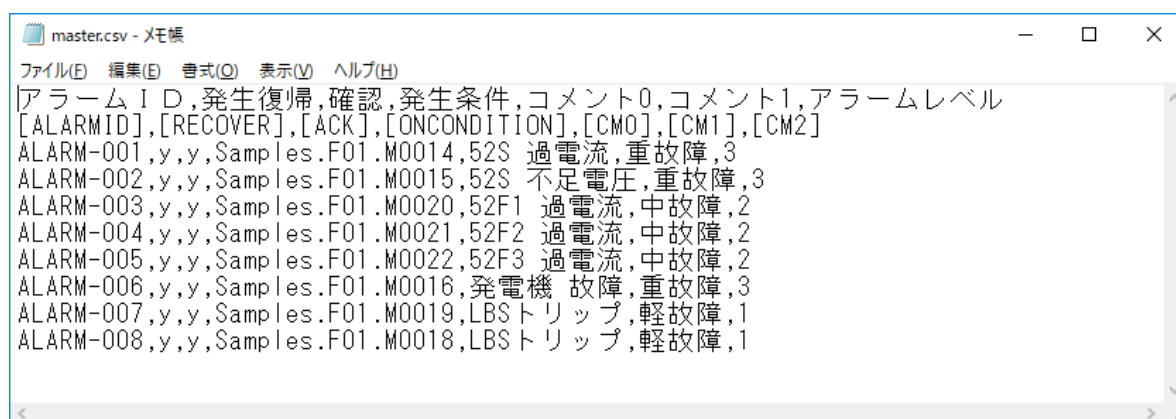
標準フレームワークのサーバ設定ファイルには、アラーム監視を行うためのアラームサーバー (CSV) として、「Alarm」という名称のアクションが登録されています。

アラームサーバーアクションは実行時にアラームマスタを読み込んでアラームの監視を行います。「Alarm」アクション用のアラームマスタは、以下に格納されています。

(プロジェクトフォルダ)¥server¥alarm¥master.csv

アラームマスタは CSV 形式のテキストファイルであり、メモ帳などのテキストエディタや Excel など自由に編集することができます。アラームマスタ「master.csv」を、メモ帳などのテキストエディタで開いてみてください。

すると、以下のような内容が記述されています。



上記はサンプルデータです。実運用の際には不要な行を削除するとともに、アラームマスタの書式のルールに従ってアラームを定義する必要があります。

アラームマスタの記述ルールは以下のとおりです。

- ・ 1 行目：タイトル行です。
- ・ 2 行目：フィールドのタイプを定義します。
- ・ 3 行目以降にアラームの設定を定義します。

標準のアラーム機能では、デフォルトで以下のようなフィールド定義となっています。

フィールド	内容
[ALARMID]	アラーム ID です。任意の文字列を定義します。アラームを識別するための一意の ID を付与してください。
[RECOVER]	発生・復帰の有無を、半角小文字の「y」または「n」で定義します。
[ACK]	確認の有無を、半角小文字の「y」または「n」で定義します。
[ONCONDITION]	アラームの発生条件を定義します。 例えば、「U01.F01.M0000」とタグバスを定義すると、当該タグが True になったときにアラーム発生とみなします（「U01.F01.M0000==TRUE」と表記しても同様の意味となります）。
[CM0]	コメント 0 に、アラームのメッセージを記述します。
[CM1]	コメント 1 は、アラームレベルのラベル文字列を定義します。「軽故障」「中故障」「重故障」のいずれかを定義できます。
[CM2]	コメント 2 は、アラームレベルの数値を定義します。上記のアラームレベルに応じたレベル値を 1 から 3 の数値で定義します。 軽故障：1 中故障：2 重故障：3

ここでは、1 行目、2 行目は変更せずに、3 行目以降のみを編集してください。

- ・ 1 行目、2 行目を残し、3 行目以降を削除する。
- ・ 以下を参考に、3 行目以降を記述する。

アラーム ID,発生復帰,確認,発生条件,コメント 0,コメント 1,アラームレベル
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]
ALARM-001,y,y,U01.F01.M0000,テストアラーム 1,軽故障,1
ALARM-002,y,y,U01.F01.M0001,テストアラーム 2,中故障,2
ALARM-003,y,y,U01.F01.M0002,テストアラーム 3,中故障,2
ALARM-004,y,y,U01.F01.M0003,テストアラーム 4,重故障,3

上記の例では、先ほど登録した仮想デバイスの U01.F01.M0000 から M0003 までのタグをアラームとして使用します。

編集が終わったら、アラームマスタを上書き保存してください。



尚、アラームメッセージなどにカンマを含めて表示したい場合、カンマを含む文字列を半角のダブルクォーテーション「"」で囲むことで対応可能です。

例)

ALARM-001,y,y,U01.F01.M0000,"テストアラーム 1, ABC",軽故障,1



アラームマスタは CSV 形式のファイルのため、Excel による編集も可能です。大量の点数のアラームを定義する場合などは、Excel による編集が便利です。

(アラームマスタを Excel で開いた例)

	A	B	C	D	E	F	G	H
1	アラーム I D	発生復帰	確認	発生条件	コメント0	コメント1	アラームレベル	
2	[ALARMID]	[RECOVER]	[ACK]	[ONCONDITION]	[CM0]	[CM1]	[CM2]	
3	ALARM-001	y	y	U01.F01.M0000	テストアラーム1	軽故障		1
4	ALARM-002	y	y	U01.F01.M0001	テストアラーム2	中故障		2
5	ALARM-003	y	y	U01.F01.M0002	テストアラーム3	中故障		2
6	ALARM-004	y	y	U01.F01.M0003	テストアラーム4	重故障		3
7								
8								

尚、カンマを含む文字列が存在する場合、Excel で CSV を保存すると自動的にダブルクォーテーションで囲んで処理されます。

注意点として、Excel で編集する場合はファイルの保存形式に注意し、かならず CSV ファイルで保存するようにしてください。例えば、タブ区切りなどで上書き保存してしまうと、アラームサーバーアクションが正しく読み込むことができなくなります。

アラームマスタの編集 (Event)

続いて、イベント監視のためのアラームマスタを編集します。

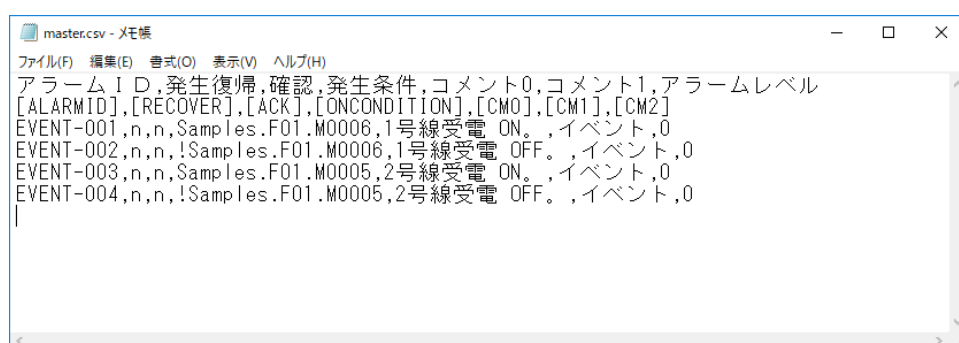
標準フレームワークのサーバ設定ファイルには、イベントの監視を行うためのアラームサーバー (CSV) として、「Event」という名称のアクションが登録されています。

「Event」アクション用のアラームマスタは、以下に格納されています。

(プロジェクトフォルダ)¥server¥event¥master.csv

アラームマスタ「master.csv」を、メモ帳などのテキストエディタで開いてみてください。

すると、以下の内容が記述されています。



上記はサンプルデータです。実運用の際には不要な行を削除するとともに、アラームマスタの書式のルールに従ってイベントの条件を定義する必要があります。

この書式はアラーム機能のアラームマスタと同様、1行目はタイトル、2行目はフィールド定義、3行目以降にイベントの定義となっています。

標準のイベント機能では、デフォルトで以下のようなフィールド定義となっています。

フィールド	内容
[ALARMID]	アラーム ID です。任意の文字列を定義します。アラームを識別するための一意の ID を付与してください。
[RECOVER]	発生・復帰の有無です。イベントの場合、「n」を指定します。
[ACK]	確認の有無です。イベントの場合、「n」を指定します。
[ONCONDITION]	イベントの発生条件を定義します。 例えば、「U01.F01.M0000」とタグを定義すると、当該タグが False から True になったときにイベント発生とみなします（機器の運転開始など）。又、「!U01.F01.M0000」、あるいは「U01.F01.M0000==FALSE」と定義すると、当該タグが True から False になったときにイベントを検出します（機器の運転停止など）。
[CM0]	コメント 0 に、イベントのメッセージを記述します。
[CM1]	コメント 1 は、アラームレベルのラベル文字列を定義します。ここでは「イベント」としてください。
[CM2]	コメント 2 は、アラームレベルの数値を定義します。ここでは「0」固定としてください。

イベント用のアラームマスタを編集してください。

- ・ 1 行目、2 行目を残し、3 行目以降を削除する。
- ・ 以下を参考に、3 行目以降を記述する。

```
アラーム I D,発生復帰,確認,発生条件,コメント 0,コメント 1,アラームレベル  
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]  
EVENT-001,n,n,U01.F01.M0010,M0010 が ON になりました,イベント,0  
EVENT-002,n,n,U01.F01.M0010,M0010 が OFF になりました,イベント,0  
EVENT-003,n,n,U01.F01.M0011,M0011 が ON になりました,イベント,0  
EVENT-004,n,n,U01.F01.M0011,M0011 が OFF になりました,イベント,0  
EVENT-005,n,n,U01.F01.M0012,M0012 が ON になりました,イベント,0  
EVENT-006,n,n,U01.F01.M0012,M0012 が OFF になりました,イベント,0
```

上記の例では、先ほど登録した仮想デバイスの U01.F01.M0010 から M0012 までのタグをイベントとして使用します。

編集が終わったら、アラームマスタを上書き保存してください。



発生条件のうち、「!U01.F01.M0010」などの先頭に記述されている半角の「!」は、論理演算の Not に相当します。

動作確認

それでは、動作を確認してみましょう。

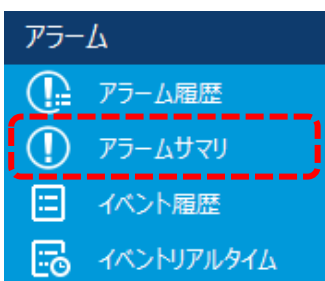
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



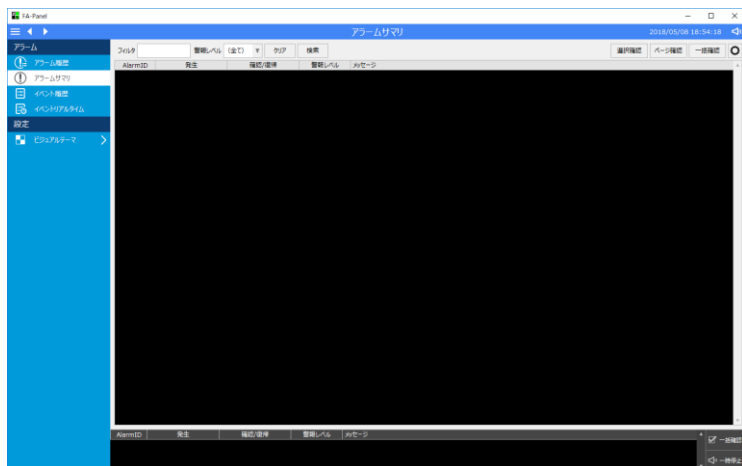
Panel Browser が起動しました。



画面左のメニューを見ると、ウィザードで追加した「アラーム履歴」「アラームサマリ」「イベント履歴」「イベントリアルタイム」が追加されています。

メニューから「アラームサマリ」をクリックします。

アラームサマリ画面が表示されました。



Panel Server から、U01 ユニットの F01 フォルダを選択し、マルチビューに M0000 から M0003 のタグが表示された状態にしてください。

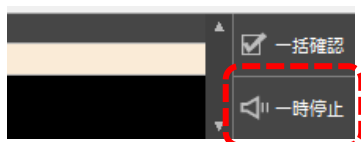
Multiview - U01.F01			
名前	パラメータ	現在値	
● M0000	M0000	FALSE	201
● M0001	M0001	FALSE	201
● M0002	M0002	FALSE	201
● M0003	M0003	FALSE	201

マルチビューから M0000 を選択し、「F5」キーを押して書き込みダイアログを表示し、値を「TRUE」にしてください（※警報音が鳴ります）。

すると以下のように、アラームサマリ画面には 1 行のアラームが新たに表示され、点滅状態となります。

フィルタ		警報レベル (全て)	▼	クリア	検索
AlarmID	発生	確認/復帰	警報レベル	メッセージ	
ALARM-001	2018/05/08 18:54:51		軽故障	テストアラーム1	

警報音を停止するには、画面右下に表示されている「一時停止」ボタンを押してください。



続いて、上記の操作と同じ要領で、M0001 から M0003 のすべてのタグに TRUE を書き込んでください。すると以下のように、新たに 3 行のアラームが追加され、点滅状態となります（※必要に応じて警報音を停止してください）。

フィルタ		警報レベル (全て)	▼	クリア	検索
AlarmID	発生	確認/復帰	警報レベル	メッセージ	
ALARM-004	2018/05/08 18:56:03		重故障	テストアラーム4	
ALARM-003	2018/05/08 18:56:03		中故障	テストアラーム3	
ALARM-002	2018/05/08 18:56:03		中故障	テストアラーム2	
ALARM-001	2018/05/08 18:54:51		軽故障	テストアラーム1	

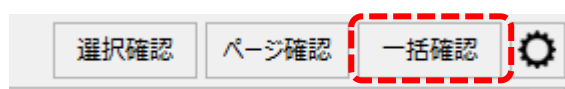
このように、軽故障は黄色、中故障はオレンジ、重故障は赤色で表示されています。

続いて、Panel Server 側で、M0000 を TRUE から FALSE に変更してください。

すると、以下のように、M0000 によって黄色点滅していたアラームが青色点滅となり、確認/復帰の列に復帰した日時が表示されました。

フィルタ		警報レベル (全て)	▼	クリア	検索
AlarmID	発生	確認/復帰	警報レベル	メッセージ	
ALARM-004	2018/05/08 18:56:03		重故障	テストアラーム4	
ALARM-003	2018/05/08 18:56:03		中故障	テストアラーム3	
ALARM-002	2018/05/08 18:56:03		中故障	テストアラーム2	
ALARM-001	2018/05/08 18:54:51	2018/05/08 18:56:41	軽故障	テストアラーム1	

続いて、アラームの確認操作を行います。アラームサマリ画面の右上にある「一括確認」ボタンをクリックしてください（※尚、1行毎に選択しての確認や、ページ単位での確認もできます）。



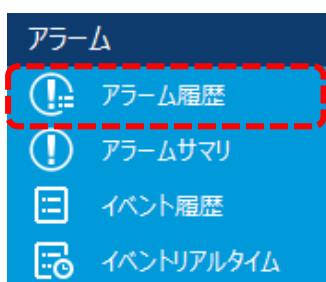
すると、アラームサマリ画面のリストからは、既に復帰した M0000 のアラームが消去され、継続して発生中の M0001、M0002、M0003 のアラームは点滅状態から点灯表示になるとともに、確認／復帰の列には確認を行った日時が表示されます。

フィルタ	警報レベル	(全て)	クリア	検索
AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-004	2018/05/08 18:56:03	2018/05/08 18:57:38	重故障	テストアラーム4
ALARM-003	2018/05/08 18:56:03	2018/05/08 18:57:38	中故障	テストアラーム3
ALARM-002	2018/05/08 18:56:03	2018/05/08 18:57:38	中故障	テストアラーム2

続いて、Panel Server 側で、M0001、M0002、M0003 を TRUE から FALSE に変更してください。

すると、アラームサマリ画面のリストからすべてのアラームの表示が消去されました。

フィルタ	警報レベル	(全て)	クリア	検索
AlarmID	発生	確認/復帰	警報レベル	メッセージ



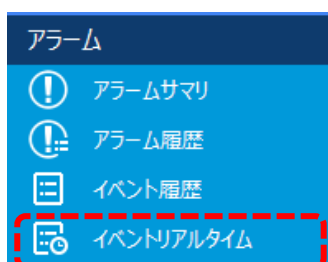
続いて、画面左のメニューから、「アラーム履歴」をクリックします。

すると、アラーム履歴画面が表示されました。

AlarmID	日時	タイプ	警報レベル	メッセージ
ALARM-004	2018/05/08 18:56:26	復帰	重故障	テストアラーム4
ALARM-003	2018/05/08 18:56:26	復帰	中故障	テストアラーム3
ALARM-002	2018/05/08 18:56:26	復帰	中故障	テストアラーム2
ALARM-004	2018/05/08 18:57:38	確認	重故障	テストアラーム4
ALARM-003	2018/05/08 18:57:38	確認	中故障	テストアラーム3
ALARM-002	2018/05/08 18:57:38	確認	中故障	テストアラーム2
ALARM-001	2018/05/08 18:57:38	確認	軽故障	テストアラーム1
ALARM-001	2018/05/08 18:56:41	復帰	軽故障	テストアラーム1
ALARM-004	2018/05/08 18:56:03	発生	重故障	テストアラーム4
ALARM-003	2018/05/08 18:56:03	発生	中故障	テストアラーム3
ALARM-002	2018/05/08 18:56:03	発生	中故障	テストアラーム2
ALARM-001	2018/05/08 18:54:31	発生	軽故障	テストアラーム1

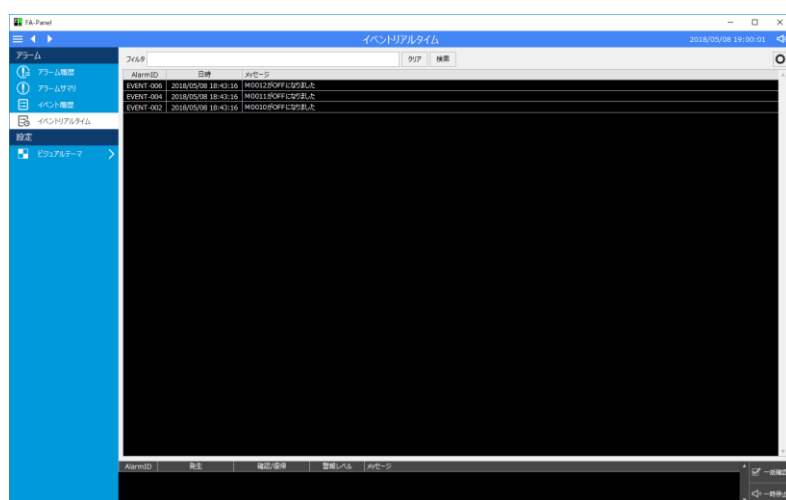
先ほどの一連の操作によるアラーム発生、復帰、確認が、アラーム履歴として記録されていることを確認してください。

引き続き、イベント機能の動作確認を行ってみましょう。



メニューから「イベントリアルタイム」をクリックします。

イベントリアルタイム画面が表示されました。



イベントリアルタイム画面のリスト上には、登録されたイベントのうち信号がOFFのイベントが検出されて表示された状態となっています。

フィルタ		クリア	検索
AlarmID	日時	メッセージ	
EVENT-006	2018/05/08 18:43:16	M0012がOFFになりました	
EVENT-004	2018/05/08 18:43:16	M0011がOFFになりました	
EVENT-002	2018/05/08 18:43:16	M0010がOFFになりました	

Panel Server から、U01 ユニットの F01 フォルダを選択し、マルチビューに M0010 から M0012 のタグが表示された状態にしてください。

Multiview - U01.F01			
名前	パラメータ	現在値	
M0009	M0009	***	
M0010	M0010	FALSE	201
M0011	M0011	FALSE	201
M0012	M0012	FALSE	201
M0013	M0013	***	

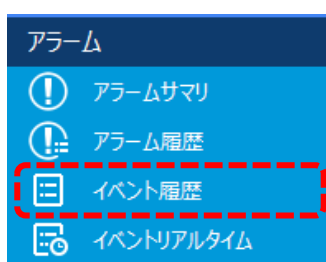
マルチビューから M0010 を選択し、「F5」キーを押して書き込みダイアログを表示し、値を「TRUE」にしてください。

すると以下のように、イベントリアルタイム画面の先頭に1行のイベントが新たに表示されました。このように、イベントリアルタイム画面では、イベントが発生すると動的に表示が更新され、先頭行に新しいイベントが表示されます。

AlarmID	日時	メッセージ
EVENT-001	2018/05/08 19:00:54	M0010がONになりました
EVENT-006	2018/05/08 18:43:16	M0012がOFFになりました
EVENT-004	2018/05/08 18:43:16	M0011がOFFになりました
EVENT-002	2018/05/08 18:43:16	M0010がOFFになりました

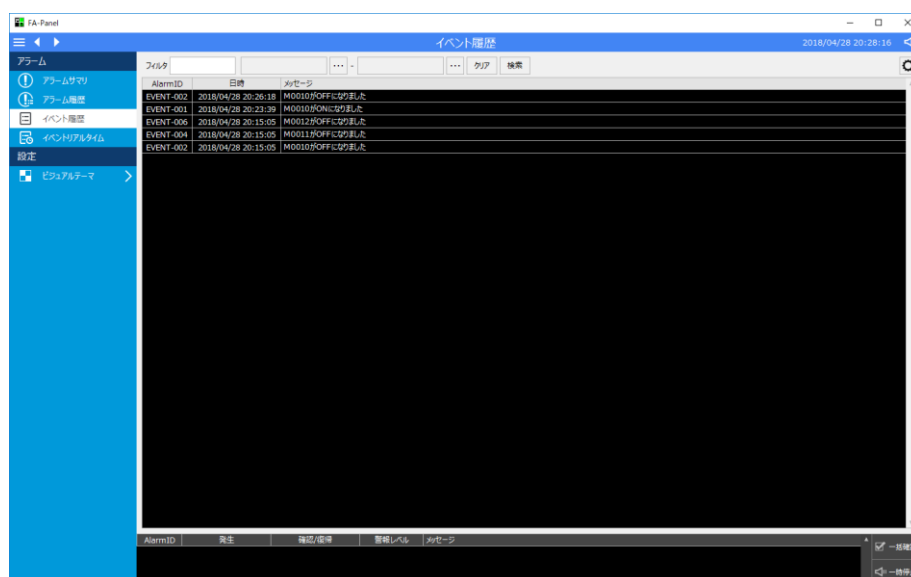
続いて、上記の操作と同じ要領で、M0010をFALSEにしてください。
 すると以下のように、リストの先頭行にOFFのイベントが表示されました。

AlarmID	日時	メッセージ
EVENT-002	2018/05/08 19:01:38	M0010がOFFになりました
EVENT-001	2018/05/08 19:00:54	M0010がONになりました
EVENT-006	2018/05/08 18:43:16	M0012がOFFになりました
EVENT-004	2018/05/08 18:43:16	M0011がOFFになりました
EVENT-002	2018/05/08 18:43:16	M0010がOFFになりました



続いて、画面左のメニューから、「イベント履歴」をクリックします。

すると、イベント履歴画面が表示されました。



イベント履歴が正しく記録されていることを確認してください。

2-2 アラームの設定について

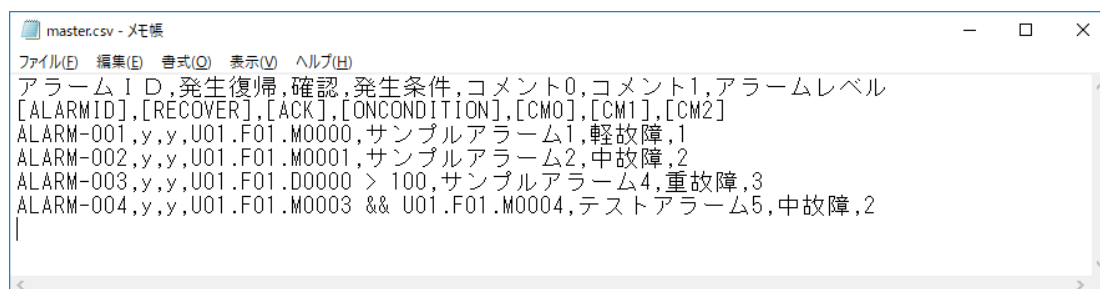


アラームマスタについて

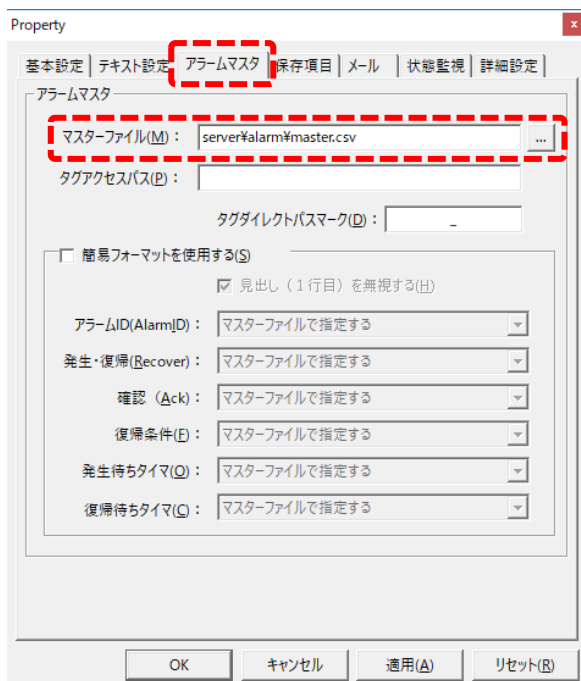
アラームマスタとは、アラームの定義ファイルです。アラームサーバーアクションは実行時にアラームマスタを読み込み、アラームの監視を行います。

アラームマスタは CSV 形式のテキストファイルであり、メモ帳などのテキストエディタや Excel など自由に編集することができます。

(アラームマスタの定義例)



アラームサーバーアクションの設定画面から、アラームマスタの参照先を指定する必要があります。



左記は、アラームサマリ画面ライブラリのサンプルに含まれている「Alarm」アクションの設定画面です。

あらかじめ、「server¥alarm¥master.csv」という参照先が設定されています。

※ この場合、サーバ設定ファイル (server.txt) が格納されているフォルダがカレントフォルダとして認識されます。

尚、左記の画面にある「簡易フォーマットを使用する」とは、旧バージョンの下位互換のために残された機能です。ほとんどの場合、簡易フォーマットを使用する必要はありません。本トレーニングでは、通常フォーマットについてのみ説明します。

アラームマスタのフォーマット

アラームマスタの記述ルールは以下のとおりです。

- ・ 1 行目：タイトル行
- ・ 2 行目：フィールドの定義
- ・ 3 行目以降：アラームの設定（発生条件、メッセージなどの定義）

（記述例）

```
アラーム I D,発生復帰,確認,発生条件,コメント 0,コメント 1,アラームレベル
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]
ALARM-001,y,y,U01.F01.M0000,サンプルアラーム 1,軽故障,1
ALARM-002,y,y,U01.F01.M0001,サンプルアラーム 2,中故障,2
ALARM-003,y,y,U01.F01.D0000 > 100,サンプルアラーム 4,重故障,3
ALARM-004,y,y,U01.F01.M0003 && U01.F01.M0004,テストアラーム 5,中故障,2
```

1 行目	<p>タイトル行。アラームマスタの列の見出しをカンマ区切りで自由に記述することができます。</p> <p>例えば、以下のように記述します。</p> <p style="text-align: center;">アラーム I D,発生復帰,確認,発生条件,コメント 0,コメント 1,アラームレベル</p>
2 行目	<p>アラームマスタに定義する各フィールドのタイプを指定します。どの列に何のフィールドを定義するのかをカンマ区切りで指定します。</p> <p>例えば、以下のように記述します。</p> <p style="text-align: center;">[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]</p> <p>ここで、[ALARMID] や [ACK] などの角括弧で囲まれた書式がフィールドを表します。上記の例では、1 列目がアラーム ID、2 列目が発生復帰の有無、3 列目が確認の有無、4 列目が発生条件・・・のように解釈されます。</p>
3 行目以降	<p>アラームの設定。2 行目で指定したフィールドの並び順で、個々のアラームの情報をカンマ区切りで定義します。</p> <p>例えば、以下のように記述します。</p> <pre>ALARM-001,y,y,U01.F01.M0000,サンプルアラーム 1,軽故障,1 ALARM-002,y,y,U01.F01.M0001,サンプルアラーム 2,中故障,2 ALARM-003,y,y,U01.F01.D0000 > 100,サンプルアラーム 4,重故障,3 ALARM-004,y,y,U01.F01.M0003 && U01.F01.M0004,テストアラーム 5,中故障,2 : : :</pre>

アラームマスタのフィールドについて

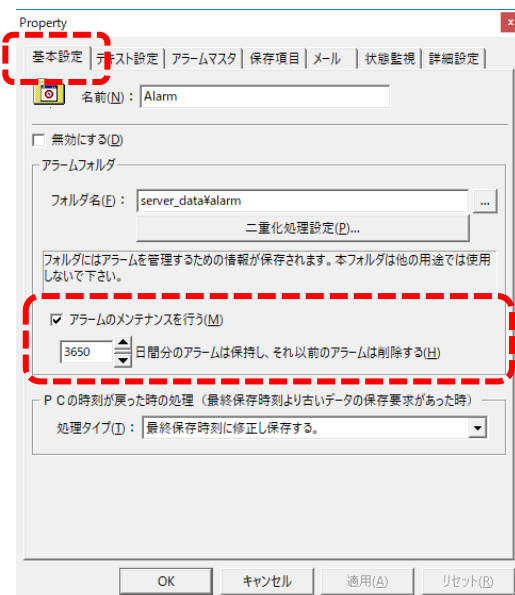
アラームマスタに定義可能なフィールドは以下のとおりです（※一部抜粋）。

アラームマスタのフィールドの詳細については、オンラインマニュアルの「サーバ作成ガイド」「アクション編」「アクションリファレンス」「アラームサーバ」「アラームマスタのフォーマット」を参照してください。

項目	フィールド名	解説
アラーム ID	[ALARMID]	アラームに付与する ID です。必ずしも数値である必要はなく、文字列を指定することも可能です。このフィールド定義を省略すると内部的に 0 から始まる連番が自動的に付与されますが、アラームマスタ内で定義する事を推奨します（特に、運用後にアラームを追加する場合を考慮し、アラーム ID は明示しておいた方が良い）。
発生・復帰	[RECOVER]	発生・復帰の有無を、半角小文字の “y” または “n” で定義します。この設定を “y”（有り）とすると、アラームの発生時と復帰時に別々のアラームイベントを発生させる事ができます。アラーム監視を行う場合は “y” を指定します。 一方、“n”（無し）とすると、発生時のみレコードが生成され、復帰時は無視されます。運転操作履歴などのイベントを登録する際に、“n” を指定します。
確認	[ACK]	確認の有無を、半角小文字の “y” または “n” で定義します。“y” にすると、画面からアラームの確認操作が行えます。“n” にすると、確認無しで自動復帰します（アラームが復帰するとアラームサマリから自動的に消去します）。
発生条件	[ONCONDITION]	アラーム発生条件を記述します。条件が TRUE になった時にアラームが発生します。
復帰条件	[OFFCONDITION]	アラーム復帰条件を記述することができます。ここで指定した条件が FALSE になった時にアラームが復帰します。省略すると、上記「発生条件」の条件が FALSE になった時を復帰とみなします。
発生待ちタイマー	[ONTIMER]	アラーム発生までのタイマーです（秒）。指定した時間の間継続して TRUE の時、アラーム発生とします。
復帰待ちタイマー	[OFFTIMER]	アラーム復帰までのタイマーです（秒）。指定した時間の間継続して FALSE の時、アラーム復帰とします。
アラームメール	[MAIL]	アラームメールを送信したい場合に定義します。発生、復帰、確認、変化など、どのタイミングでメールを送信したいのかを指定することができます。
コメント	[CM0] ～ [CMn]	コメントを自由に設定可能です。設定可能な個数に上限はありません。主にアラームメッセージなどに使用します。 [CM0]、[CM1]、[CM2]などと定義することで複数のコメントを定義できます。
タグコメント	[CM0:TAG] ～ [CMn:TAG]	タグの値をコメントとして出力できます。上記同様、設定可能な個数に上限はありません。

アラームの保存期間

アラームサーバーアクションの設定画面から、保存期間を日単位で指定することができます。



基本設定タブの中にある、「アラームのメンテナンスを行う」にチェックを入れ、保存期間を指定します。

標準フレームワークの場合、デフォルトで 3650 日（約 10 年間）となっています。

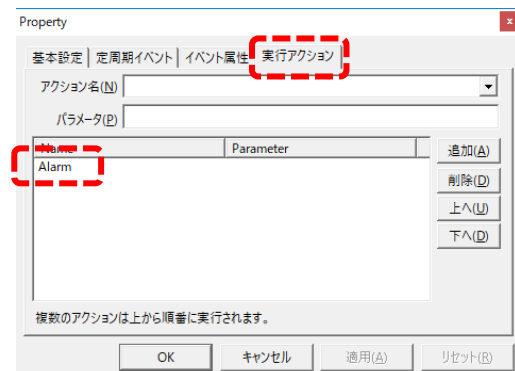
もし、保存期間を変更したい場合は、左記の設定を変更してください。

アラームサーバーアクションと定周期イベントについて

アラームサーバーアクションは、必ず定周期イベントから呼び出しを行うように設定する必要があります。

例えば、標準フレームワークのアラーム機能の場合、「Alarm」という名称の 1 秒周期の定周期イベントが登録されており、アラームサーバーアクション「Alarm」を定期的呼び出すように設定されています。

（定周期イベント「Alarm」の実行アクション）



尚、この定周期イベントは、アラームを検出する周期ではないという事に注意してください。

アラームの検出は、タグの更新周期に基づいて、タグの値が変化する都度行われます。検出されたアラームは、一旦、アラームサーバ側の内部でメッセージのキューに蓄えられる仕組みとなっており、上記の定周期イベントから呼び出されたタイミングでメッセージキューをチェックし、キューにアラームが存在する場合にディスクへの書き出しやクライアントへのアラームの通知が行われます。

Lesson 3 アラームのカスタマイズ

3-1 アラームにコメントを追加する



このセクションでは、標準のアラーム機能に対して以下のカスタマイズを行います。

- アラームマスタに「追加コメント」として [CM3] を追加する。
- アラームサマリ画面に、新たに「追加コメント」の列を追加する。
- アラーム履歴画面では、「メッセージ」列に「追加コメント」を含めて表示する。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson3-1¥Lesson3-1.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

アラームマスタの編集

教材プロジェクトの以下のフォルダ配下にある「master.csv」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-1¥server¥alarm¥master.csv

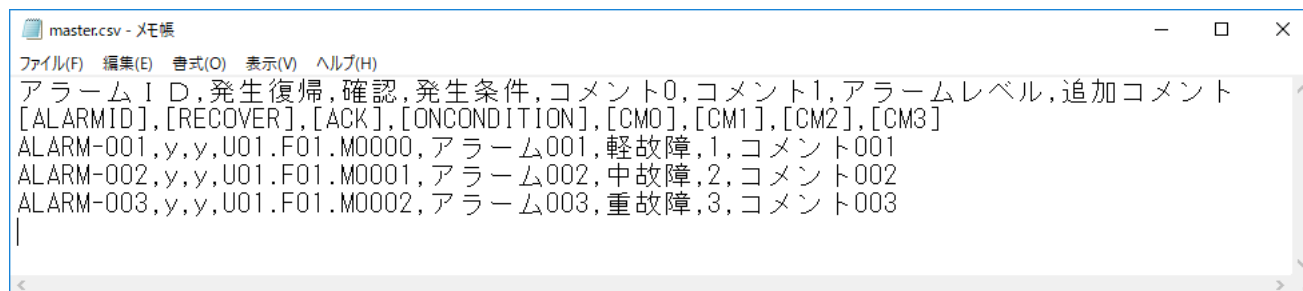
すると、以下のような内容が記述されています（※以下は、分かりやすいようにカンマの位置を調整しています）。

アラーム I D,	発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2]
ALARM-001,	y,	y,	U01.F01.M0000,	アラーム 001, 軽故障,		1
ALARM-002,	y,	y,	U01.F01.M0001,	アラーム 002, 中故障,		2
ALARM-003,	y,	y,	U01.F01.M0002,	アラーム 003, 重故障,		3

上記に対して、列の末尾に[CM3] を新たに追加します。ここでは以下のように編集してください。

アラーム I D,	発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル,	追加コメント
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2],	[CM3]
ALARM-001,	y,	y,	U01.F01.M0000,	アラーム 001, 軽故障,	1,		コメント 001
ALARM-002,	y,	y,	U01.F01.M0001,	アラーム 002, 中故障,	2,		コメント 002
ALARM-003,	y,	y,	U01.F01.M0002,	アラーム 003, 重故障,	3,		コメント 003

以下は、編集後のイメージです（メモ帳で表示した例）。

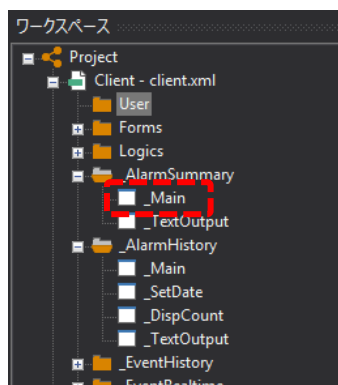


アラームサマリのレイアウト編集

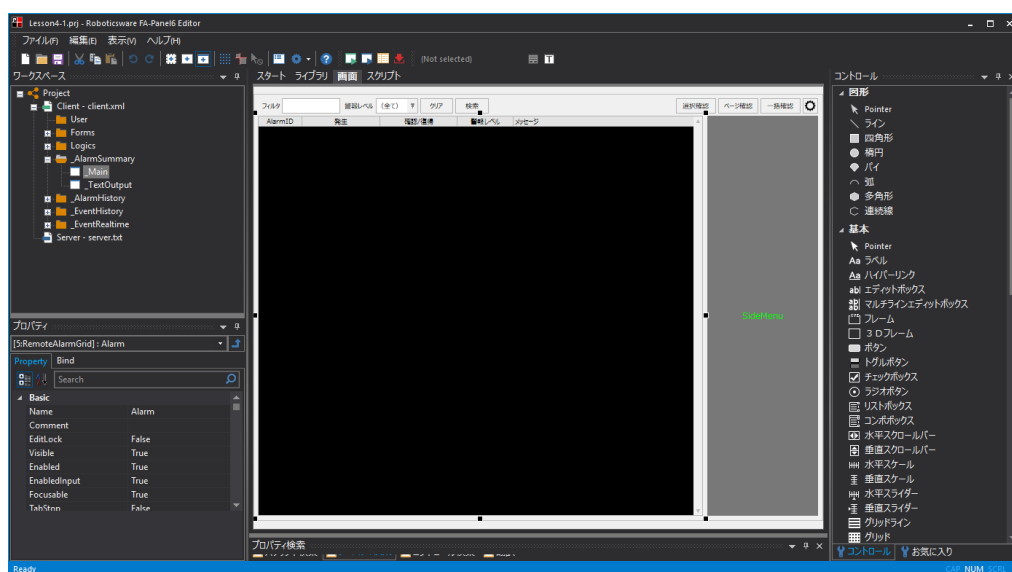
続いて、アラームサマリ画面のレイアウトの編集を行います。

1. Panel Editor でアラームサマリ画面を選択する

サンプルプロジェクトが開かれた Panel Editor の左ツリーから、Client の下にある_AlarmSummary フォルダを開き、「_Main」フォームを選択してください。

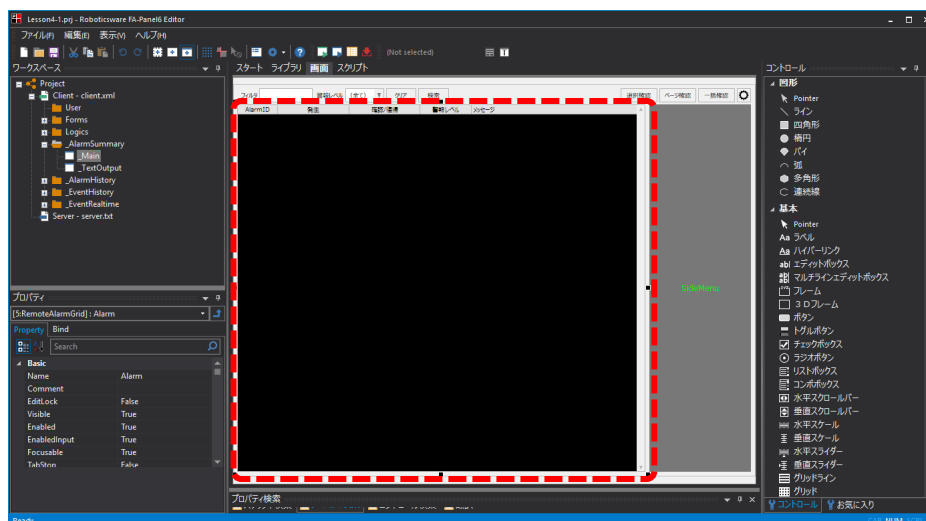


(アラームサマリ画面)

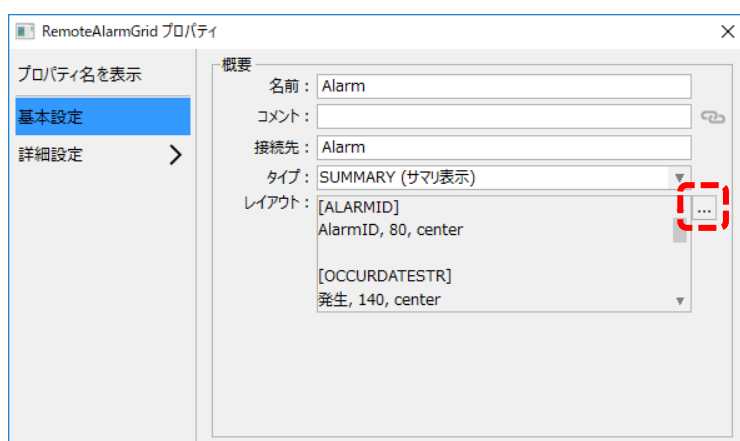


2. リモートアラームグリッドのプロパティ編集

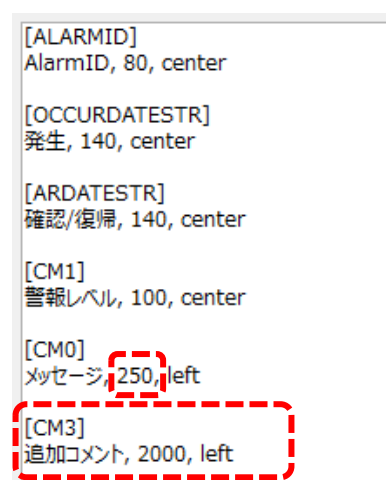
アラームサマリ画面の中央に配置されているリモートアラームグリッドをダブルクリックします。



すると、設定ダイアログが表示されるので、「レイアウト」(AlarmLayout プロパティ) の編集ボタンをクリックします。



すると、エディタが表示されます。このエディタからアラームレイアウトを編集することができます。



アラームレイアウトの定義のルールとして、グリッドの1列について各2行のパラメータで構成されます。

1行目に列に表示するフィールドの書式を、2行目に列のタイトル、幅(ピクセル単位)、文字の寄せ(center/left/right)を指定します。

ここでは、以下のように編集し、OK ボタンで保存します。

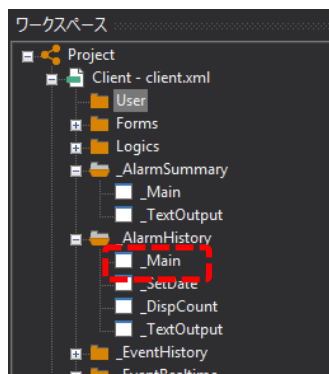
- ・ [CM0]のメッセージの幅を、2000 から 250 に変更
- ・ [CM3]の定義を新たに2行追加(幅は2000としておきます)。

アラーム履歴のレイアウト編集

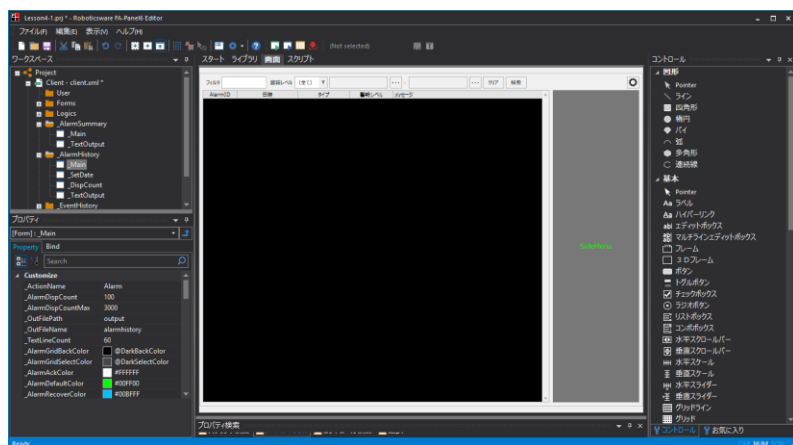
続いて、アラーム履歴画面のレイアウトの編集を行います。

1. Panel Editor でアラームサマリ画面を選択する

サンプルプロジェクトが開かれた Panel Editor の左ツリーから、Client の下にある _AlarmHistory フォルダを開き、「_Main」フォームを選択してください。

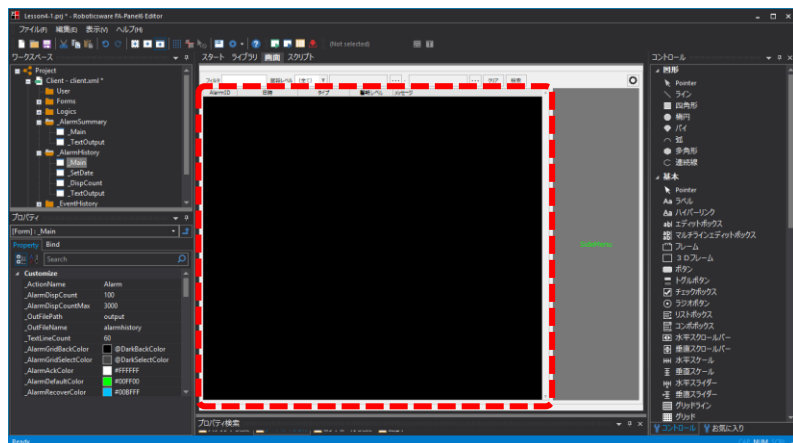


(アラーム履歴画面)

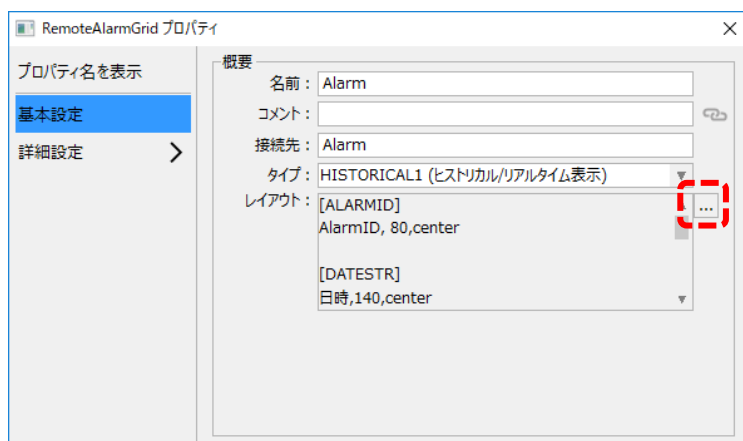


2. リモートアラームグリッドのプロパティ編集

アラーム履歴画面の中央に配置されているリモートアラームグリッドをダブルクリックします。



設定ダイアログが表示されるので、「レイアウト」(AlarmLayout プロパティ) の編集ボタンをクリックします。



すると、エディタが表示されます。



アラームサマリ画面と同様にこのエディタからアラームレイアウトの編集を行うことができます。

尚、アラーム履歴についてはアラームサマリで行った方法とは別の手法を試してみます。以下のように編集し、OK ボタンで保存してください。

編集前)

[CM0]
メッセージ, 2000, left

編集後)

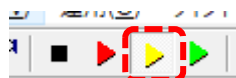
[CM0] & " " & [CM3]
メッセージ, 2000, left

※[CM0]と記述されていた行に、上記のような式を記述する。

動作確認

それでは、動作を確認してみましょう。

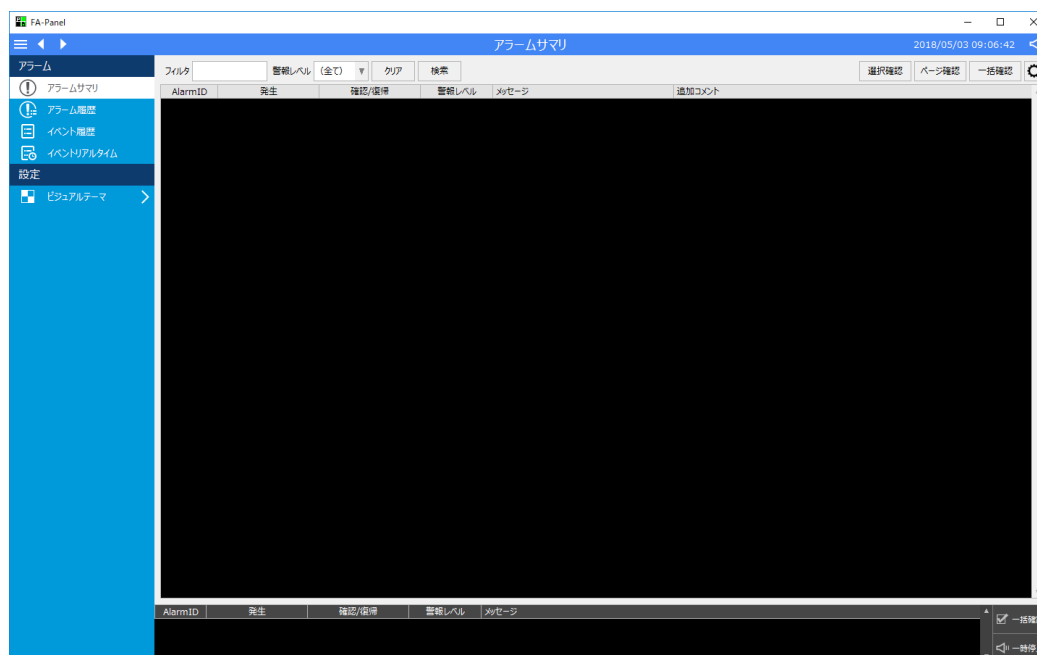
Panel Server の画面上部にあるオンライン (黄矢印) をクリックしてください。



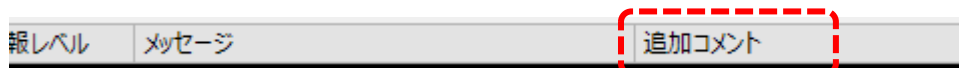
Panel Editor の画面上部にあるブラウザ実行アイコン (緑矢印) をクリックし、Panel Browser を呼び出します。



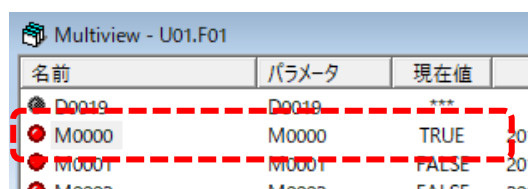
Panel Browser が起動したら、アラームサマリ画面を表示してください。



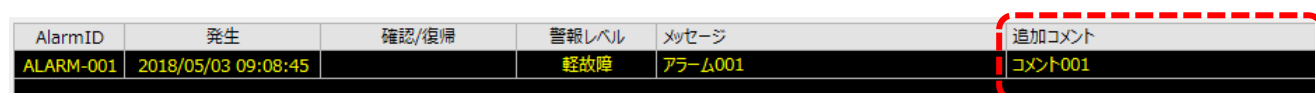
まず、アラームサマリ画面のグリッドに、先ほど新たに追加した「追加コメント」という列が追加されていることを確認してください。



続いて、Panel Server から、アラームに登録されているタグのうち、U01.F01.M0000 に TRUE を書き込んでみてください。

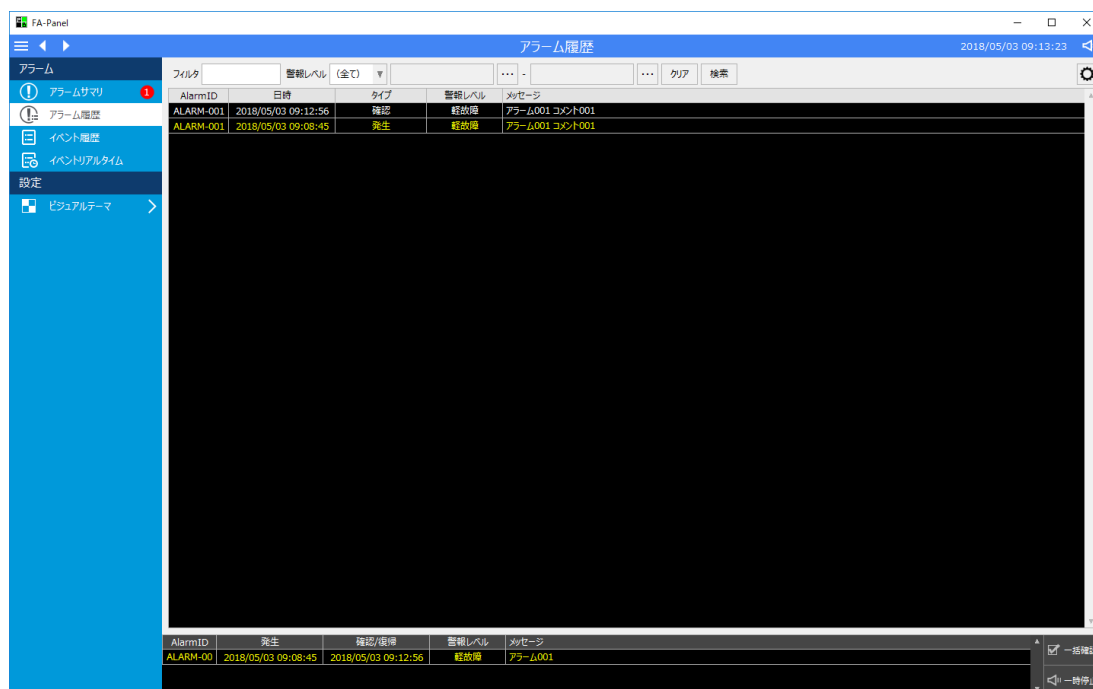


すると、アラームサマリ画面に、新たに追加した CM3 のコメントの内容が表示されました。



※アラームサマリ画面から一括確認を行っておいください。

続いて、アラーム履歴画面を表示してください。



アラーム履歴のメッセージ列を見ると、変更前のアラームコメント「アラーム 001」の後ろに、CM3 で定義した追加コメントが付加されて表示されていることが分かります。

レベル	メッセージ
軽微	アラーム001 コメント001
軽微	アラーム001 コメント001

このように、アラームグリッドに表示する内容は、新たに列を追加して表示する以外にも、1つの列の中に複数のコメントの内容を連結して表示することもできます。

※動作確認が終わったら、TRUE を書き込んだタグの値を FALSE に戻しておいてください。

■アラームのフィールドについて

アラーム機能（およびイベント機能）では、画面側のアラームグリッドの表示列の定義を行う際などに、「フィールド名」を用いて定義を行います。

例えば、標準のアラームサマリ画面のリモートアラームグリッドの「AlarmLayout」プロパティでは、以下のように記述されています。

(AlarmLayout プロパティ)

```
[ALARMID]
AlarmID, 80, center

[OCCURDATESTR]
発生, 140, center

[ARDATESTR]
確認/復帰, 140, center

[CM1]
警報レベル, 100, center

[CM0]
メッセージ, 2000, left
```

上記の「[ALARMID]」や「[OCCURDATESTR]」などが、アラームのフィールドです。この例の場合、アラームグリッドの1列目にアラーム ID、2列目に発生日時、3列目に確認復帰日時、4列目に CM1（標準のアラーム機能では警報レベル）、5列目に CM0（同じく、アラームメッセージ）を指定しています。



アラームのフィールド名に関する詳細については、オンラインマニュアルの「サーバ作成ガイド」「アクション編」「アクションリファレンス」「アラームサーバ」「フィールド名について」に記述されています。

■アラームレイアウトの演算式構文

アラームレイアウトの定義では、フィールド名を含む演算式を記述することができます。

例えば、以下のように記述すると、CM0 と CM3 の内容を連結して表示します。

[CM0] & [CM3] ・・・ 「&」演算子はビット同士の And 演算のほか、文字列の連結を行うことができます

[CM0] & " " & [CM3] ・・・ダブルクォーテーションで囲んだ文字列を連結します（この例は半角スペース）。

[CM0] & "(" & [CM3] & ")" ・・・CM3 の内容を半角括弧で囲んで表示します。

その他、条件式により表示内容を分岐させることもできます（※条件式を使用した例については、別のカスタマイズ事例にて別途紹介します）。



演算式構文に関する詳細については、オンラインマニュアルの「スクリプトガイド」「演算式構文」に記述されています。

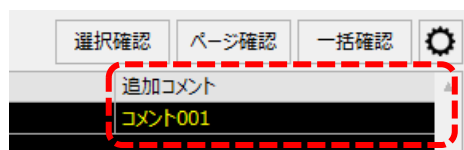
■グリッドの末尾列の列幅について

アラームサマリ画面のグリッドに設定した例では、末尾に追加した CM3 の幅に以下のように「2000」を指定しました。

[CM3]
追加コメント,2000,left

上記のように、末尾に指定する列の幅に大きな値を設定しておくことで、Panel Browser のウィンドウのサイズを広げられた際に、グリッドの右端に余白が表示されないようにすることができます。

（ウィンドウの幅を狭く縮めた例）



（ウィンドウの幅を広げた例。グリッドの右端の列が切れずに表示されている）

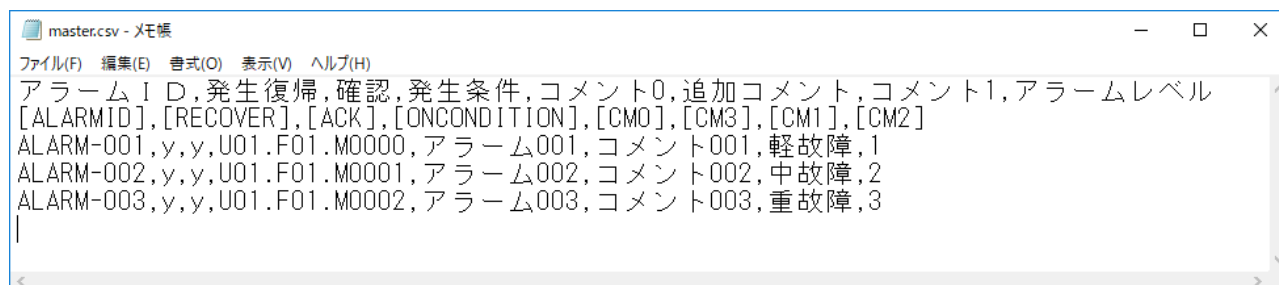


■アラームマスタの列の順序は自由に定義可能

アラームレイアウトの定義では、列の順序に制約はありません。

例えば、今回のカスタマイズ例のケースでは、以下のように定義しても同様に動作します。

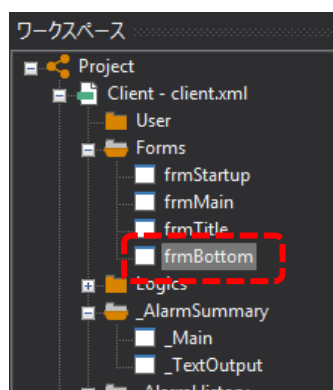
(※以下は、CM0 と CM1 の間に CM3 を追加しています)



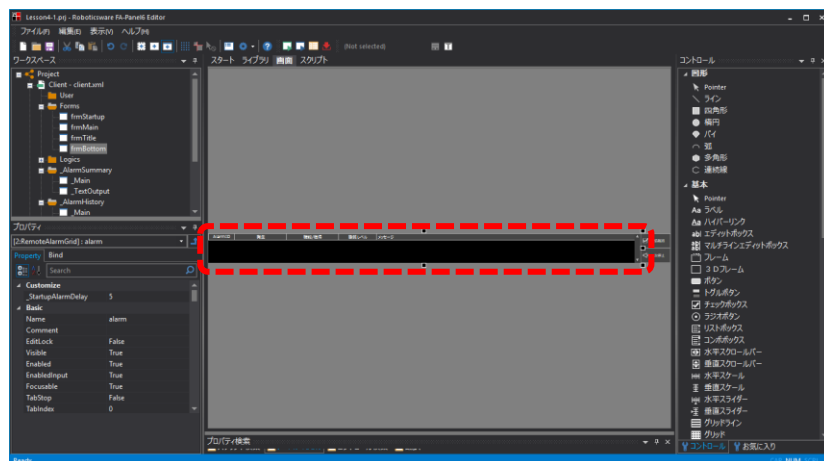
■画面下部のアラームサマリのレイアウトを編集する場合

標準フレームワークの画面下部には、アラームサマリが常時表示されています。このアラームサマリ表示についても同様にレイアウトの変更を行いたい場合は、アラームサマリ画面のレイアウト編集と同じ要領で変更することができます。

この場合、編集の対象となるフォームは、Forms フォルダ内に格納されている「frmBottom」フォームです。



(frmBottom フォームのリモートアラームグリッド)



3-2 アラームメッセージにタグの値を含める



このセクションでは、標準のアラーム機能に対して以下のカスタマイズを行います。

- アラームマスタに [CM3:TAG] を追加する。
- アラームの保存項目に [CM3:TAG] を追加する。
- アラームサマリおよびアラーム履歴のメッセージに、タグの値を含めて表示する。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson3-2¥Lesson3-2.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

アラームマスタの編集

教材プロジェクトの以下のフォルダ配下にある「master.csv」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-2¥server¥alarm¥master.csv

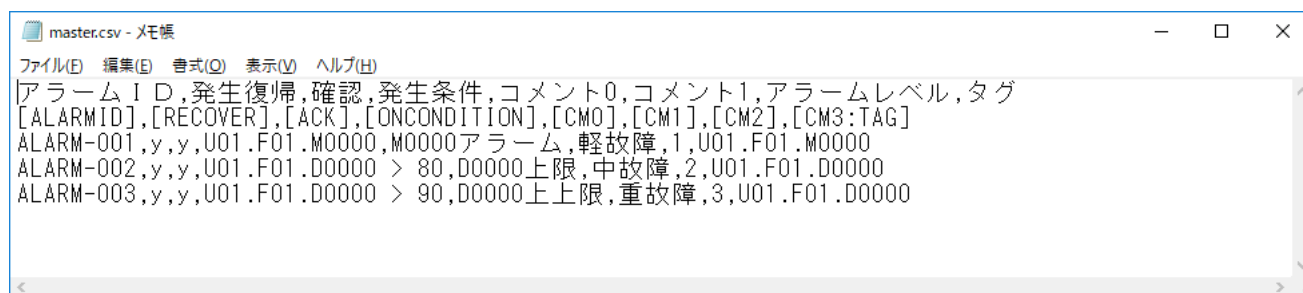
すると、以下のような内容が記述されています。この設定例では、アナログ上限、上上限の監視を行っています（※以下は、分かりやすいようにカンマの位置を調整しています）。

アラーム I D, 発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1], [CM2]
ALARM-001, y,	y,	U01.F01.M0000,	M0000 アラーム,	軽故障,	1
ALARM-002, y,	y,	U01.F01.D0000 > 80,	D0000 上限,	中故障,	2
ALARM-003, y,	y,	U01.F01.D0000 > 90,	D0000 上上限,	重故障,	3

上記に対して、列の末尾に[CM3:TAG] を新たに追加します。ここでは以下のように編集してください。

アラーム I D, 発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル,	タグ
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1], [CM2],	[CM3:TAG]
ALARM-001, y,	y,	U01.F01.M0000,	M0000 アラーム,	軽故障,	1,	U01.F01.M0000
ALARM-002, y,	y,	U01.F01.D0000 > 80,	D0000 上限,	中故障,	2,	U01.F01.D0000
ALARM-003, y,	y,	U01.F01.D0000 > 90,	D0000 上上限,	重故障,	3,	U01.F01.D0000

以下は、編集後のイメージです（メモ帳で表示した例）。



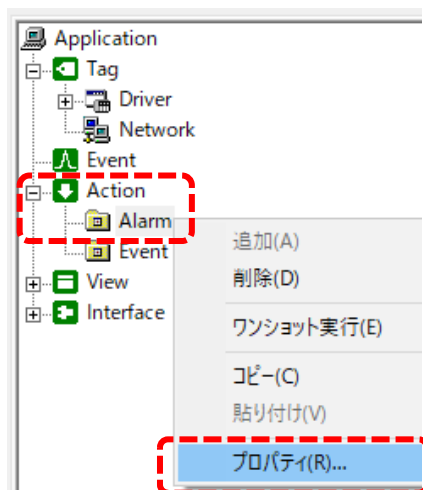
上記で追加した CM3:TAG とは、タグコメントと呼ばれるフィールドです。タグコメントを使用すると、この列に定義されたタグバスの値をアラームのログに含めることができます。

尚、上記の設定では、コメントの番号（CM0、CM1 などの番号）は重複しないように注意してください。例えば上記の例では、CM0、CM1、CM2、CM3:TAG、のように、0 から 3 の番号で定義しています。今回追加したタグコメント（CM3:TAG）は末尾に「:TAG」と付いていますが、他の通常のコメントと同様に、番号が重複しないようにしてください。

保存項目の追加

続いて、教材プロジェクトのサーバ設定ファイル（server.txt）を Panel Server で開き、アラームサーバーアクションの「保存項目」に、先ほど追加した CM3:TAG を追加します。

1. アラームサーバーアクションの設定画面を開く



サンプルプロジェクトが開かれた Panel Server の左ツリーから、Action の下にある Alarm を選択し、右クリックでプロパティを開いてください。

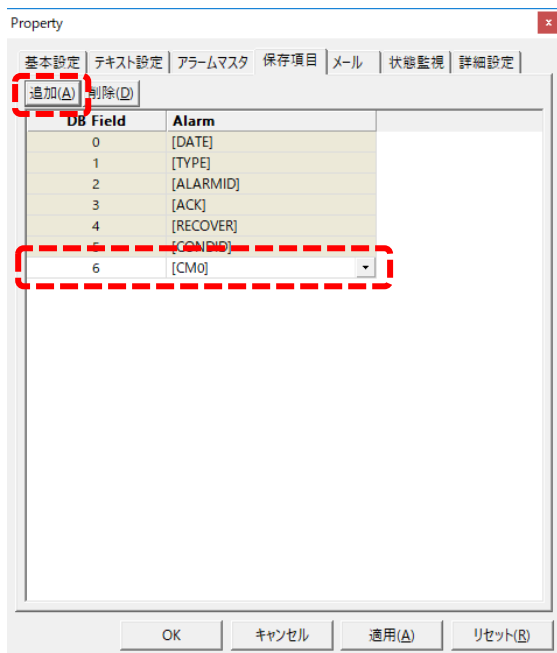
すると、アラームの設定ダイアログが表示されます。

2. 保存項目の追加

設定ダイアログから保存項目の設定を行います。

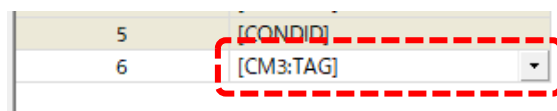


画面上部にある「保存項目」タブをクリックしてください。



画面から「追加」ボタンをクリックします。

すると、左記のように新たな行が1行追加されます。



新たに追加された左記の部分ダブルクリックで編集状態にし、[CM3:TAG]に変更してOKボタンをクリックしてください。

この設定により、アラームのログデータに、アラームマスタの CM3:TAG の列で定義したタグパスのタグ値が含まれて保存されるようになります。

※保存項目については、後ほど補足説明にて解説します。

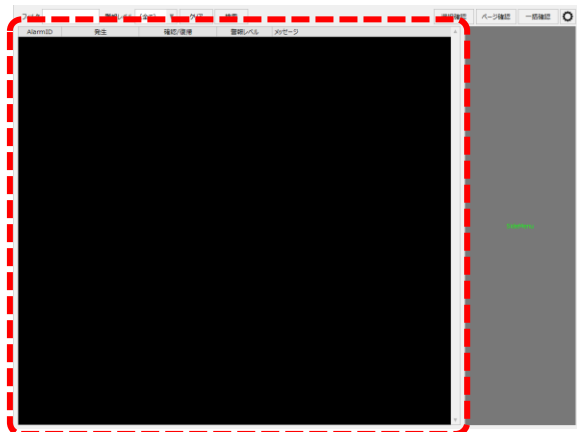
アラームサマリ／アラーム履歴のレイアウト編集

続いて、アラームサマリ画面、アラーム履歴画面のレイアウトの編集を行います。

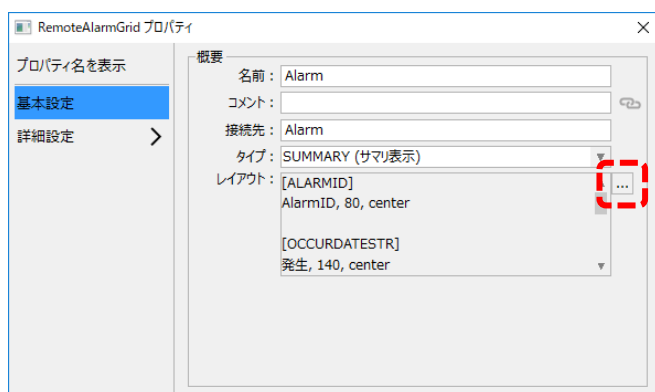
今回のカスタマイズでは、メッセージにアラーム発生時のタグの値を含めて表示されるようにします。

1. アラームサマリ画面のリモートアラームグリッド

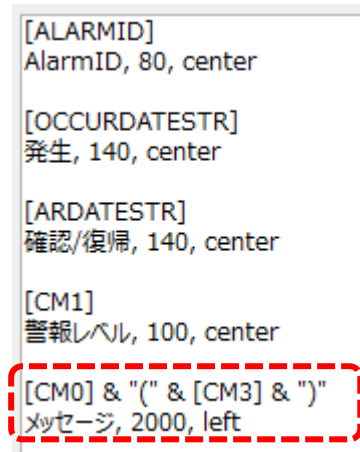
Panel Editor からアラームサマリ画面（_AlarmSummary_Main）を開き、アラームサマリ画面の中央に配置されているリモートアラームグリッドをダブルクリックします。



すると、設定ダイアログが表示されるので、「レイアウト」（AlarmLayout プロパティ）の編集ボタンをクリックします。



エディタからアラームレイアウトの編集を行います。



ここでは、以下のように編集し、OK ボタンで保存します。

編集前)

[CM0]

メッセージ, 2000, left

編集後)

[CM0] & "(" & [CM3] & ")"

メッセージ, 2000, left

2. アラーム履歴画面のリモートアラームグリッド

続いて、アラーム履歴画面（_AlarmHistory_Main）を開き、アラーム履歴画面の中央に配置されているリモートアラームグリッドをダブルクリックして設定ダイアログを開きます。

設定ダイアログから、「レイアウト」（AlarmLayout プロパティ）の編集ボタンをクリックしてください。

エディタが表示されたら、アラームレイアウトの編集を行います。

<pre>[ALARMID] AlarmID, 80,center</pre>	アラームサマリと同様に以下のように編集し、OK ボタンで保存します。
<pre>[DATESTR] 日時,140,center</pre>	編集前)
<pre>[TYPETEXT] タイプ,120,center</pre>	[CM0] メッセージ, 2000, left
<pre>[CM1] 警報レベル, 100, center</pre>	編集後)
<pre>[CM0] & "(" & [CM3] & ")" メッセージ, 2000, left</pre>	[CM0] & "(" & [CM3] & ")" メッセージ, 2000, left

動作確認

それでは、動作を確認してみましょう。

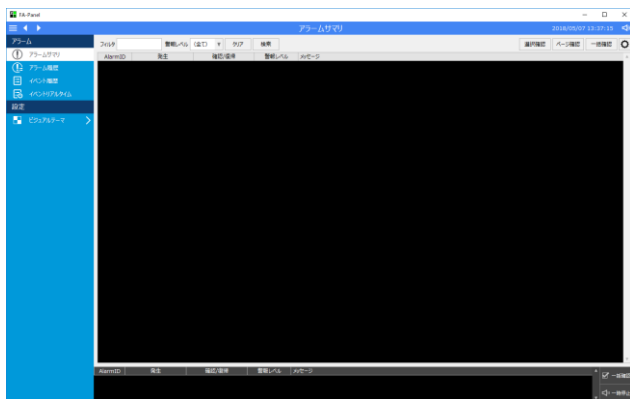
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



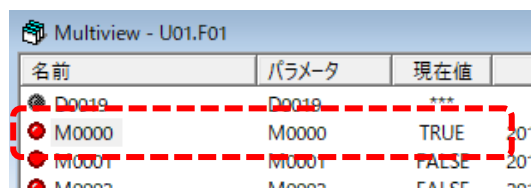
Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動したら、アラームサマリ画面を表示してください。

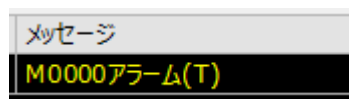


Panel Server から、アラームに登録されているタグのうち、U01.F01.M0000 に TRUE を書き込んでみてください。



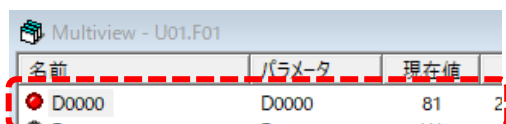
名前	パラメータ	現在値
D0019	D0019	***
M0000	M0000	TRUE
M0001	M0001	FALSE
M0002	M0002	FALSE

すると、メッセージの末尾に、括弧で囲まれた値が表示されるようになりました。この例では「(T)」と表示されています。



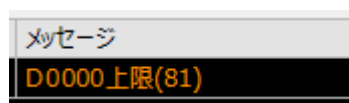
表示の確認が終わったら、Panel Server から M0000 に FALSE を書き込むとともに、アラームサマリ画面から一括確認を行っておいってください。

続いて、Panel Server から、アラームに登録されているタグのうち、U01.F01.D0000 に対して 81 を書き込んでみてください。

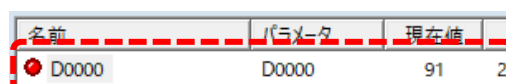


名前	パラメータ	現在値
D0000	D0000	81

すると、上限警報のメッセージの末尾に、括弧で囲まれた 81 という値が表示されました。

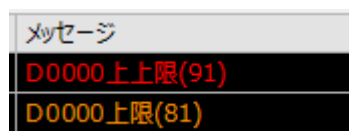


引き続き、Panel Server から D0000 の値を 91 に変更してみてください。



名前	パラメータ	現在値
D0000	D0000	91

すると、上上限警報のメッセージの末尾に、括弧で囲まれた 91 という値が表示されました。



表示の確認が終わったら、Panel Server から D0000 に 0 を書き込むとともに、アラームサマリ画面から一括確認を行っておいってください。

Panel Browser から、アラーム履歴画面を表示してみてください。

AlarmID	日時	タイプ	警報レベル	メッセージ
ALARM-003	2018/05/07 13:53:43	確認	重故障	D0000 上上限(0)
ALARM-002	2018/05/07 13:53:43	確認	中故障	D0000 上限(0)
ALARM-003	2018/05/07 13:53:05	復帰	重故障	D0000 上上限(0)
ALARM-002	2018/05/07 13:53:05	復帰	中故障	D0000 上限(0)
ALARM-003	2018/05/07 13:51:38	発生	重故障	D0000 上上限(91)
ALARM-002	2018/05/07 13:49:07	発生	中故障	D0000 上限(81)
ALARM-001	2018/05/07 13:48:09	確認	軽故障	M0000 アラーム(F)
ALARM-001	2018/05/07 13:47:04	復帰	軽故障	M0000 アラーム(F)
ALARM-001	2018/05/07 13:44:17	発生	軽故障	M0000 アラーム(T)

上記のとおり、現状ではすべてのメッセージの末尾に括弧で囲まれたタグ値が表示されていることが分かります。

上記のままでも機能としては問題ありませんが、以下の2点について改善すると、より使い勝手がよくなりそうです。

改善点：

- ・タグ値をメッセージに含めるのはアナログ値のみとする（ビット警報の値を見てもあまり意味がないため）。
- ・かつ、発生時のメッセージについてのみタグ値を表示するようにする。

引き続き、上記に対する改善を行います。

アラームマスタの編集2

先ほど編集した、教材プロジェクトの以下のフォルダ配下にある「master.csv」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-2¥server¥alarm¥master.csv

現状は以下のような内容となっています（※以下は、分かりやすいようにカンマの位置を調整しています）。

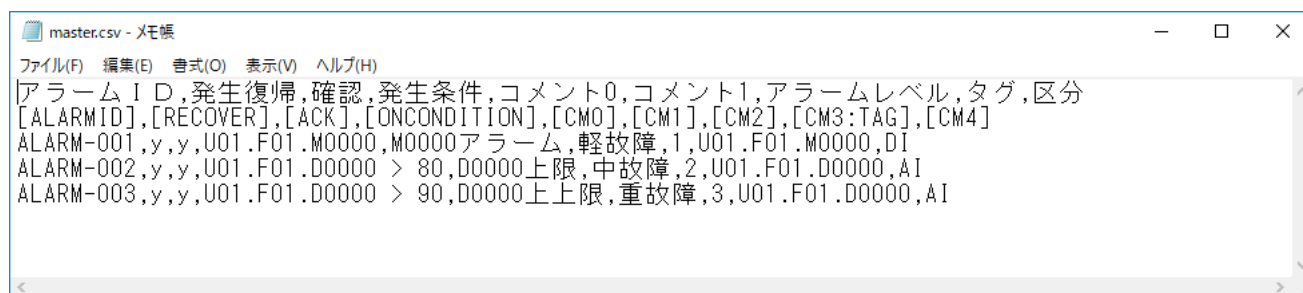
アラームID	D, 発生復帰	確認	発生条件	コメント0	コメント1	アラームレベル	タグ
[ALARMID]	[RECOVER]	[ACK]	[ONCONDITION]	[CM0]	[CM1]	[CM2]	[CM3:TAG]
ALARM-001	y,	y,	U01.F01.M0000,	M0000 アラーム	軽故障	1,	U01.F01.M0000
ALARM-002	y,	y,	U01.F01.D0000 > 80,	D0000 上限	中故障	2,	U01.F01.D0000
ALARM-003	y,	y,	U01.F01.D0000 > 90,	D0000 上上限	重故障	3,	U01.F01.D0000

上記に対して、列の末尾に区分として[CM4] を新たに追加します。

ここでは、以下のように編集してください。

アラームID	D, 発生復帰	確認	発生条件	コメント0	コメント1	アラームレベル	タグ	区分
[ALARMID]	[RECOVER]	[ACK]	[ONCONDITION]	[CM0]	[CM1]	[CM2]	[CM3:TAG]	[CM4]
ALARM-001	y,	y,	U01.F01.M0000,	M0000 アラーム	軽故障	1,	U01.F01.M0000,	DI
ALARM-002	y,	y,	U01.F01.D0000 > 80,	D0000 上限	中故障	2,	U01.F01.D0000,	AI
ALARM-003	y,	y,	U01.F01.D0000 > 90,	D0000 上上限	重故障	3,	U01.F01.D0000,	AI

以下は、編集後のイメージです（メモ帳で表示した例）。



上記の設定例では、アラームマスタの列の末尾に新たに「CM4」という列を追加し、この列に「AI」か「DI」の区分を追加しています。アラームレイアウト側の設定で、CM4 が AI のアラームについてのみメッセージに値を含めて表示するように改造を行います。

アラームサマリ／アラーム履歴のレイアウト編集2

1. アラームサマリ画面のリモートアラームグリッド

アラームサマリ画面（_AlarmSummary_Main）を開き、アラームサマリ画面の中央に配置されているリモートアラームグリッドをダブルクリックして設定ダイアログを開きます。

設定ダイアログから、「レイアウト」（AlarmLayout プロパティ）の編集ボタンをクリックしてください。

エディタが表示されたら、以下のようにアラームレイアウトの編集を行ってください。

```
[ALARMID]
AlarmID, 80, center

[DATESTR]
日時, 140, center

[TYPETEXT]
タイプ, 120, center

[CM1]
警報レベル, 100, center

{[TYPE]=="new" && [CM4]=="AI"}{[CM0] & "(" & [CM3] & ")"},{default}{[CM0]}
メッセージ, 2000, left
```

編集前)

```
[CM0] & "(" & [CM3] & ")"
メッセージ, 2000, left
```

編集後)

```
{[TYPE]=="new" && [CM4]=="AI"}{[CM0] & "(" & [CM3] & ")"},{default}{[CM0]}
メッセージ, 2000, left
```

このような定義を行うことで、条件式の判定結果に応じてメッセージの表示内容を分岐させることができます。

※上記の表記内容については、後ほど補足説明にて解説します。

2. アラーム履歴画面のリモートアラームグリッド

続いて、アラーム履歴画面（_AlarmHistory_Main）を開き、アラーム履歴画面の中央に配置されているリモートアラームグリッドをダブルクリックして設定ダイアログを開きます。

設定ダイアログから、「レイアウト」（AlarmLayout プロパティ）の編集ボタンをクリックしてください。

エディタが表示されたら、アラームレイアウトを編集してください。

```
[ALARMID]
AlarmID, 80, center

[OCCURDATESTR]
発生, 140, center

[ARDATESTR]
確認/復帰, 140, center

[CM1]
警報レベル, 100, center

{[TYPE]== "new" && [CM4]== "AI"}{[CM0] & "(" & [CM3] & ")"},{default}-{[CM0]}
メッセージ, 2000, left
```

※編集内容はアラームサマリと同様です。

動作確認2

それでは、動作を確認してみましょう。

Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください（※尚、アラームマスタの編集内容を反映するためには、必ず、Panel Server を1度オフラインにし、再度オンラインにしてください）。



Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動したら、アラーム履歴画面を表示してください。

AlarmID	日時	タイプ	警報レベル	メッセージ
ALARM-003	2018/05/07 13:53:43	確認	重故障	D0000上上限
ALARM-002	2018/05/07 13:53:43	確認	中故障	D0000上限
ALARM-003	2018/05/07 13:53:05	復帰	重故障	D0000上上限
ALARM-002	2018/05/07 13:53:05	復帰	中故障	D0000上限
ALARM-003	2018/05/07 13:51:38	発生	重故障	D0000上上限(91)
ALARM-002	2018/05/07 13:49:07	発生	中故障	D0000上限(81)
ALARM-001	2018/05/07 13:48:09	確認	軽故障	M0000アラーム
ALARM-001	2018/05/07 13:47:04	復帰	軽故障	M0000アラーム
ALARM-001	2018/05/07 13:44:17	発生	軽故障	M0000アラーム

アラーム履歴に表示されているメッセージのうち、アナログアラーム（CM4 に「AI」と定義したアラーム）かつ、発生時のメッセージについてのみ、括弧に囲まれた値が表示されていることを確認してください。

■アラームレイアウトの演算式構文

アラームレイアウトの定義では、条件式を記述することができます。演算式の評価結果に応じて表示内容を分岐させることにより、より柔軟な表示を行うことができます。

条件式と結果の定義は、半角の「{」と「}」で組み合わせて表現します。また、条件式と結果の組は、半角カンマ「,」で区切って列挙することにより、いくつでも定義することができます。

{条件式 1}{結果 1},{条件式 2}{結果 2},{条件式 3}{結果 3},・・・{default}{すべての条件以外の結果}

例えば、今回の例では、以下のような記述を行っています。

{[TYPE]=="new" && [CM4]=="AI"}{[CM0] & "(" & [CM3] & ")"},{default}{[CM0]}

上記は、TYPE が“new”（発生時）の場合かつ CM4 が“AI”の場合のみ、CM0 に値（CM3）を含めて表示し、その他の場合は CM0 のみを表示、という意味となります。



演算式構文に関する詳細については、オンラインマニュアルの「スクリプトガイド」「演算式構文」に記述されています。

■アラームの保存項目について

基本設定 テキスト設定 アラームマスタ 保存項目 メール	
追加(A) 削除(D)	
DB Field	Alarm
0	[DATE]
1	[TYPE]
2	[ALARMID]
3	[ACK]
4	[RECOVER]
5	[CONDID]
6	[CM3:TAG]

アラームサーバーアクションの設定画面から、「保存項目」の定義を行うことができます。保存項目とは、アラームのログデータに記録されるデータ項目のことです。

デフォルトでは、[DATE]、[TYPE]、[ALARMID]、[ACK]、[RECOVER]、[CONDID]フィールドが保存項目として定義されています。

今回のカスタマイズでは、[CM3:TAG]フィールドを指定する事により、タグの値を保存項目に追加しています。

ここで、タグ値をメッセージに含めて表示したい場合には、必ず保存項目に追加する必要があります。タグ値は常に変化するため、例えばアラームが発生した時点の値を保存項目として記録しておくことにより、発生時点のタグの値をアラーム履歴に含めて表示できるようになります。

尚、画面ライブラリの各アラーム機能では、メッセージ（[CM0]）は保存項目に含めるように設定されていません。この場合、アラームサマリや履歴に表示されるメッセージの元データは、アラームマスタに定義されているメッセージが表示されます。従って、例えばシステムの運用開始後にアラームマスタのメッセージを変更した場合、過去データのアラーム履歴を表示すると全て新しいメッセージで表示される動作となります。

もしも、過去のアラーム履歴については変更前のメッセージを表示したい場合（アラームマスタの編集の影響を受けないようにしたい場合）には、[CM0] を保存項目に含めるようにしてください。

3-3 アラームのしきい値を画面から設定する



このセクションでは、標準のアラーム機能に対して以下のカスタマイズを行います。

- メモリデバイスユニットを追加し、タグの値を保存できるようにする。
- アラームマスタの発生条件の比較値に、上記で追加したタグを指定する。
- フォームを追加し、上記のタグの値を画面から変更できるようにする。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

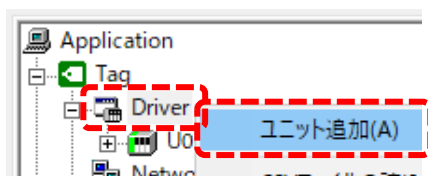
¥¥fc01¥Lesson3-3¥Lesson3-3.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

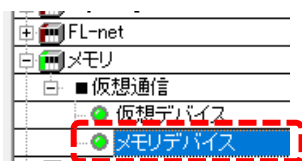
メモリデバイスユニットを追加する

教材プロジェクトのサーバ設定ファイル（server.txt）を Panel Server で開き、メモリデバイスユニットを追加します。このユニットは、画面から設定したアラームのしきい値をタグに保存できるようにするために使用します。

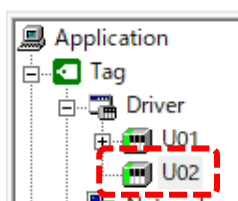
1. ユニットを追加する



サンプルプロジェクトが開かれた Panel Server の左ツリーから、Tag の下にある Driver を右クリックして「ユニット追加」を選択してください。



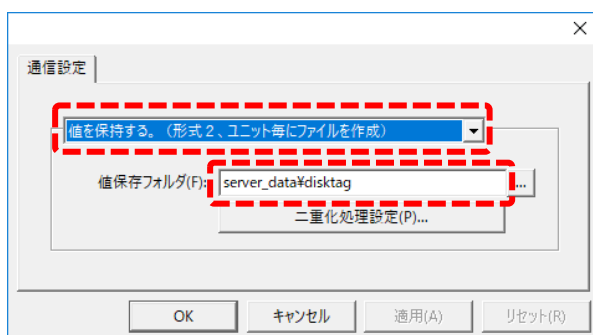
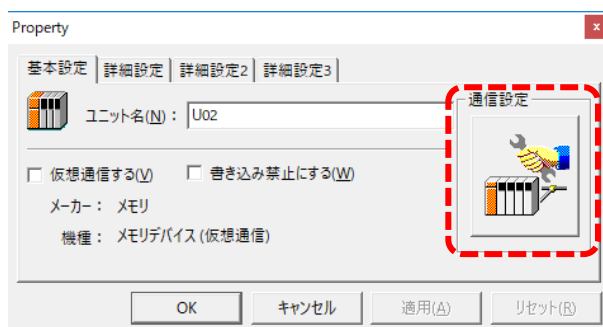
ドライバー一覧が表示されるので、一覧から、メモリ／仮想通信／メモリデバイスを選択して追加してください。



U02 ユニットが新たに追加されました。
今回は U02 という名称のまま使用します。

2. ユニットのプロパティを設定する

U02 のプロパティを表示し、通信設定ボタンをクリックします。



設定画面のリストから「値を保持する。(形式2、ユニット毎にファイルを作成)」を選択してください。

また、値保持フォルダに以下のように入力してください。

server_data¥disktag

この設定により、U02 配下に登録されたタグの値が PC 上で保持されるようになります。



特別な理由がないかぎり、値の保存形式は「形式2、ユニット毎にファイルを作成」を選択してください。

3. フォルダ、タグを作成する

U02 に「F01」フォルダを作成し、その下に以下の4点のタグを作成してください。

タグ名	デバイス	パラメータ	内容
U02.F01.D0000_HIHI	数値	変更不要 (N,1)	D0000 上上限しきい値
U02.F01.D0000_HI	数値	同上	D0000 上限しきい値
U02.F01.D0001_HI	数値	同上	D0001 上限しきい値
U02.F01.D0002_LO	数値	同上	D0002 下限しきい値

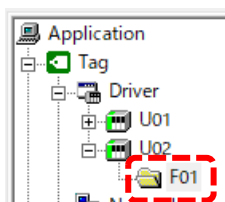
4. タグに値を書き込む

ここで、Panel Server を一度オンラインにし、上記で追加した4点のタグに値をかきこんでみましょう。

Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



Panel Server がオンライン状態になったら、ツリーから、Tag/Driver の U02.F01 を選択してください。



マルチビューには以下のように表示されています。これらのタグは現時点ではどこからも参照されていないため、ランプが消灯の状態となっています。

Multiview - U02.F01		
名前	パラメータ	現在値
● D0000_HIHI	N:N,1	***
● D0000_HI	N:N,1	***
● D0001_HI	N:N,1	***
● D0002_LO	N:N,1	***

マルチビュー上で任意のタグをクリックして選択状態にして、キーボードの「F1」キーを押してみてください。すると、以下のようにモニタリング状態になりました。

Multiview - U02.F01		
名前	パラメータ	現在値
● D0000_HIHI	N:N,1	0
● D0000_HI	N:N,1	0
● D0001_HI	N:N,1	0
● D0002_LO	N:N,1	0

この状態で個々のタグに対して値を書き込みます（値を書き込むには、F5 キーを押して書き込みダイアログを表示してください）。

各タグに対して、以下の値を書き込んでください。

タグ名	書き込む値
U02.F01.D0000_HIHI	90
U02.F01.D0000_HI	80
U02.F01.D0001_HI	70
U02.F01.D0002_LO	-10

書き込みが終わったら、値が保持されていることを確認してみます。

一旦、Panel Server の画面上部にあるオフライン（黒四角）をクリックしてください。



再び、オンライン（黄矢印）をクリックしてください。



ツリーから U02.F01 を選択し、マルチビューで F1 キーによりモニタリング状態にしてみてください。
以下のように値が保持されていることを確認してください。

名前	パラメータ	現在値
D0000_HIHI	N:N,1	90
D0000_HI	N:N,1	80
D0001_HI	N:N,1	70
D0002_LO	N:N,1	-10

このように、メモリデバイスを使用することで PC 上にタグの値を保持することができるようになります。

確認が終わったら、Panel Server をオフラインにしておいてください。



タグの値をメモリデバイスに保持する以外に、PLC にアクセスするタグを作成して PLC 上のデバイスに値を保持することもできます。

アラームマスタの編集

教材プロジェクトの以下のフォルダ配下にある「master.csv」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-3¥server¥alarm¥master.csv

すると、以下のような内容が記述されています。この設定例では、アナログ上下限の監視を行っています（※以下は、分かりやすいようにカンマの位置を調整しています）。

アラーム I D, 発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル	
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2]
ALARM-001, y,	y,	U01.F01.D0000 > 80,	D0000 上限,	中故障,	2	
ALARM-002, y,	y,	U01.F01.D0000 > 90,	D0000 上上限,	重故障,	3	
ALARM-003, y,	y,	U01.F01.D0001 > 50,	D0001 上限,	中故障,	2	
ALARM-004, y,	y,	U01.F01.D0002 < -10,	D0002 下限,	中故障,	2	

上記に対して、発生条件のしきい値として固定値として記述されている数値の部分で、先ほど追加した U02 ユニットのタグを参照するように変更します。以下のように編集してください。

アラーム I D, 発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル	
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2]
ALARM-001, y,	y,	U01.F01.D0000 > U02.F01.D0000_HI,	D0000 上限,	中故障,	2	
ALARM-002, y,	y,	U01.F01.D0000 > U02.F01.D0000_HIHI,	D0000 上上限,	重故障,	3	
ALARM-003, y,	y,	U01.F01.D0001 > U02.F01.D0001_HI,	D0001 上限,	中故障,	2	
ALARM-004, y,	y,	U01.F01.D0002 < U02.F01.D0002_LO,	D0002 下限,	中故障,	2	

以下は、編集後のイメージです（メモ帳で表示した例）。

```

master.csv - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
アラーム I D,発生復帰,確認,発生条件,コメント0,コメント1,アラームレベル
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]
ALARM-001,y,y,U01.F01.D0000 > U02.F01.D0000_HI,D0000上限,中故障,2
ALARM-002,y,y,U01.F01.D0000 > U02.F01.D0000_HIHI,D0000上上限,重故障,3
ALARM-003,y,y,U01.F01.D0001 > U02.F01.D0001_HI,D0001上限,中故障,2
ALARM-004,y,y,U01.F01.D0002 < U02.F01.D0002_LO,D0002下限,中故障,2

```



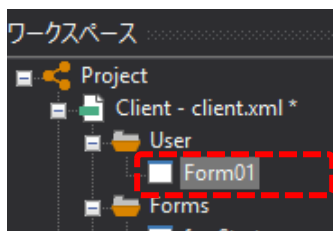
上記の例のとおり、アラームマスタの発生条件には複数のタグを含めることができます。アナログ上下限の警報しきい値等を運用開始後にユーザー自身で自由に変更できるようにしたい場合は、今回の例のように比較値をタグ経由で参照することで実現可能です。

アラーム設定画面の作成

続いて、アラームのしきい値を設定するための設定画面を作成します。

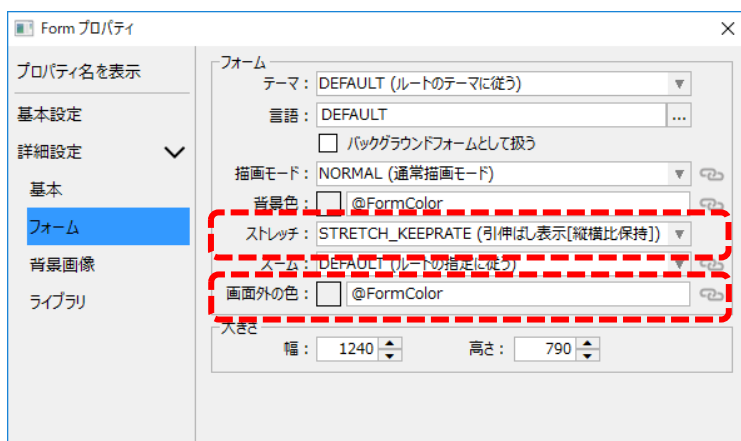
1. フォームの追加

Panel Editor から新規にフォームを追加してください。



今回は例として、User フォルダの下に Form01 というフォームを作成しています（この名称は自由に変更することもできます）。

フォームのプロパティを以下のように変更しておきます。



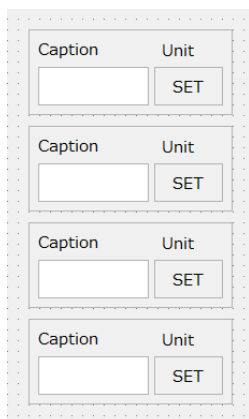
●ストレッチ（Stretch プロパティ）を「STRETCH_KEEPRATE」にしてください。

●同じく、画面外の色（ViewFillColor プロパティ）を、「@FormColor」にしてください。

2. タグ編集用の部品を配置する

しきい値が格納されたメモリデバイスのタグを編集するための部品を配置します。
今回は例として、部品ライブラリの部品を使用します

Panel Editor のライブラリタブから、部品ライブラリの数値部品を表示し、「inputnum204」部品を選択してフォームに4つ張り付けて配置してください。




左記は配置例です。

各部品をダブルクリックして設定ダイアログを表示し、各種設定を行ってください。

※設定内容は次頁の設定例を参照してください。

以下は、設定例です。

イメージ	対象部品	設定項目	設定値
	部品 1	タグ設定／入出力タグ	U02.F01.D0000_HIHI
		表示設定／キャプション	D0000 上上限
		表示設定／単位	(ブランク)
	部品 2	タグ設定／入出力タグ	U02.F01.D0000_HI
		表示設定／キャプション	D0000 上限
		表示設定／単位	(ブランク)
	部品 3	タグ設定／入出力タグ	U02.F01.D0001_HI
		表示設定／キャプション	D0001 上限
		表示設定／単位	(ブランク)
	部品 4	タグ設定／入出力タグ	U02.F01.D0002_LO
		表示設定／キャプション	D0002 下限
		表示設定／単位	(ブランク)
		表示設定／最小値	-100

3. メニューに画面呼び出し処理を追加する

フォームの編集が完了したら、メニュー（またはフォーム呼び出しボタンなど）から Form01 フォームの呼び出し処理を作成します。今回は例として、標準フレームワークの左メニューから呼び出すように MenuConf に以下の記述を行ってください。

(MenuConf プロパティの設定例)

```

[アラーム]
アラーム, _AlarmSummary_Main, アラームサマリ, sys:alarmsummary_24.png,T
アラーム, _AlarmHistory_Main, アラーム履歴, sys:alarmhistory_24.png,T
アラーム, _EventHistory_Main, イベント履歴, sys:eventhistory_24.png,T
アラーム, _EventRealtime_Main, イベントリアルタイム, sys:eventrealtime_24.png,T
アラーム, User.Form01, アラーム設定, sys:disp_details_24.png,T

[設定]
設定, [visualtheme], ビジュアルテーマ, sys:visualtheme_24.png
;設定, [closeBrowser], 終了, sys:close_24.png

```

動作確認

それでは、動作を確認してみましょう。

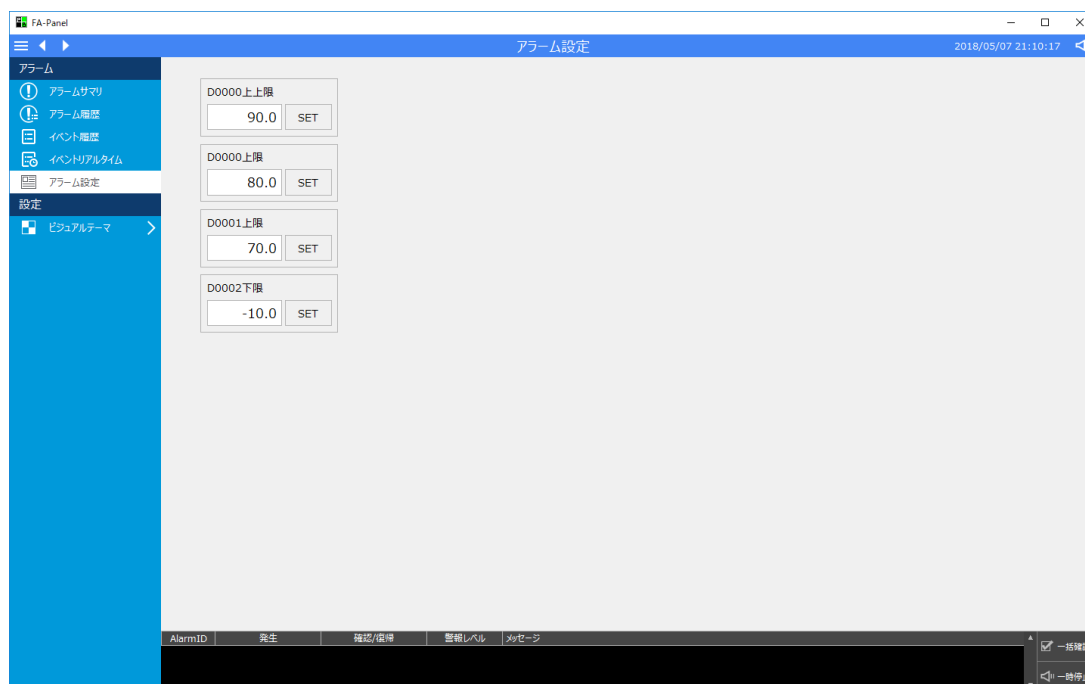
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動したら、アラーム設定画面を表示してください。



アラーム設定画面には、各アラームのしきい値の設定値が表示されています。

Panel Server のマルチビューで U02.F01 フォルダの内容を表示してみると、各タグの値が正しく表示されている事が確認できます。

Multiview - U02.F01		
名前	パラメータ	現在値
● D0000_HIHI	N:N,1	90
● D0000_HI	N:N,1	80
● D0001_HI	N:N,1	70
● D0002_LO	N:N,1	-10

D0000上上限
95.0 SET

D0000上限
85.0 SET

D0001上限
60.0 SET

D0002下限
-15.0 SET

試しに、各設定値を画面から編集してみてください（※値を変更後、「SET」ボタンをクリックすると書き込まれます）。

Panel Server のマルチビューで、各タグの値が正しく更新されている事を確認してください。

Multiview - U02.F01		
名前	パラメータ	現在値
● D0000_HIHI	N:N,1	95
● D0000_HI	N:N,1	85
● D0001_HI	N:N,1	60
● D0002_LO	N:N,1	-15

続いて、アラームを発生させてみます。

アラームに登録されている U01.F01.D0000 に対して、設定画面から設定した上限値を超える値を Panel Server から書き込んでください。ここでは例として、86 を書き込みます。

Multiview - U01.F01		
名前	パラメータ	現在値
● D0000	D0000	86
● D0001	D0001	0
● D0002	D0002	0

すると、アラームサマリに D0000 上限警報が表示されました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-001	2018/05/07 20:10:26		中故障	D0000 上限

続いて、D0000 に対して、設定画面から設定した上上限値を超える値を Panel Server から書き込んでください。ここでは例として、96 を書き込みます。

Multiview - U01.F01		
名前	パラメータ	現在値
● D0000	D0000	96
● D0001	D0001	0
● D0002	D0002	0

すると、アラームサマリに D0000 上上限警報が表示されました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-002	2018/05/07 20:13:35		重故障	D0000 上上限
ALARM-001	2018/05/07 20:10:26		中故障	D0000 上限

確認が終わったら、D0000 のタグの値を 0 に戻し、アラームサマリの画面から一括確認を行っておいください。



概要

リモートアラームグリッドコントロール（または、リモートアラームクライアントコントロール）には、アラームステートモニタと呼ばれる機能が用意されています。

アラームステートモニタを設定すると、コントロールが接続しているアラームサーバのアラーム検出状態のモニタリングを行い、アラームの状態をタグに書き出すことができます。この機能は、クライアント側で代表警報信号を生成する際などに使用します。

このセクションでは、標準のアラーム機能に対して以下のカスタマイズを行います。

- リモートアラームグリッドのアラームステートモニタ機能を設定し、内部タグで代表警報信号を作成する。
- 画面上に代表警報のランプ表示を追加する。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson3-4¥Lesson3-4.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

アラームステートモニタ機能を設定する

アラームステートモニタの設定は、クライアント側で常時存在するフォーム上に配置されたアラームグリッド（またはアラームクライアント）のうち、全体でいずれか1つのコントロールにのみ実装するようにします。複数のコントロールに同じ設定を行ってしまうと、同様のモニタリング処理が重複して実行されてしまうため、処理負荷が重くなる等の問題が発生する場合があります。

今回は例として、標準フレームワークの画面下部に常時表示されているアラームサマリに使用されているリモートアラームグリッドを使用する事とします。

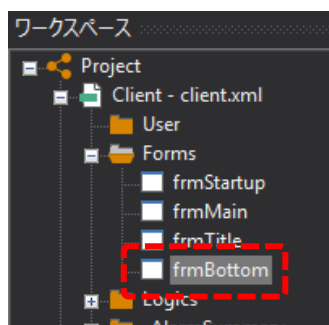


標準フレームワークのウィザード機能では、画面下部のアラームサマリ表示を非表示に設定することもできます。もし、画面下部のアラームサマリを非表示にした状態で、クライアント側で代表アラーム信号を作成したい場合は、バックグラウンドフォーム（標準フレームワークの場合、Logics.Hidden フォーム）にリモートアラームクライアントコントロールを張り付けて設定するようにします。

リモートアラームクライアントの利用に関しては、本トレーニング中の別のカスタマイズ事例にて別途紹介します。

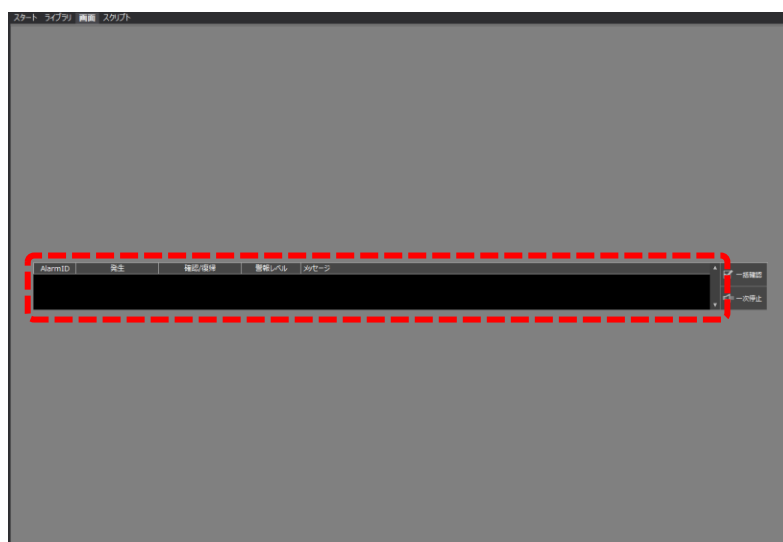
1. 画面下部のアラームサマリのフォームを選択する

Panel Editor のワークスペースのツリーから、Forms フォルダの下にある「frmBottom」を選択してください。



frmBottom フォームが表示されました。

フォームの中央に配置されているリモートアラームグリッドをダブルクリックして設定ダイアログを表示してください。



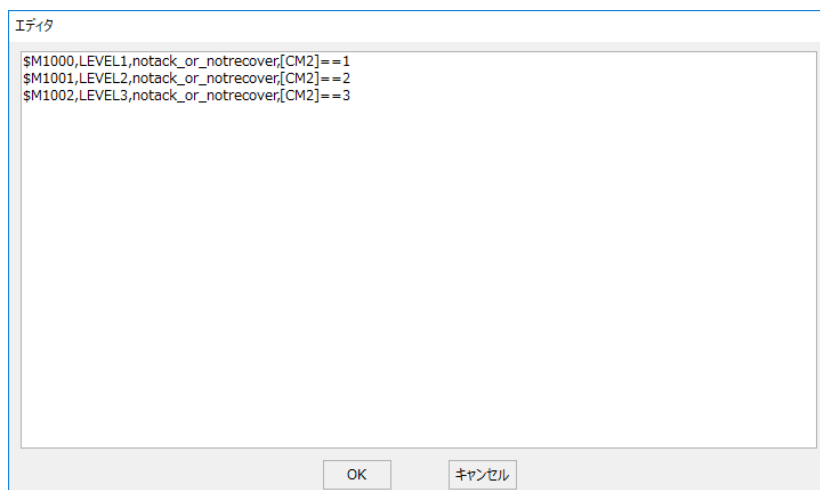
2. AlarmStateMonitor プロパティを設定する



設定ダイアログから「詳細設定」「アラーム」を選択し、代表警報（AlarmStateMonitor プロパティ）の設定ボタンをクリックします。

すると、エディタが表示されます。

表示されたエディタから以下の3行を入力し、OK ボタンをクリックしてください。



(設定内容)

```
$M1000, LEVEL1, notack_or_notrecover, [CM2]==1
$M1001, LEVEL2, notack_or_notrecover, [CM2]==2
$M1002, LEVEL3, notack_or_notrecover, [CM2]==3
```

上記の設定により、CM2 に定義されているアラームレベル毎に、発生中または未確認のアラームが1 つでも存在する場合に、一番目のパラメータで指定された内部タグに True が書き込まれます。

- ・ アラームレベル 1 (軽故障) . . . \$M1000
- ・ アラームレベル 2 (中故障) . . . \$M1001
- ・ アラームレベル 3 (重故障) . . . \$M1002

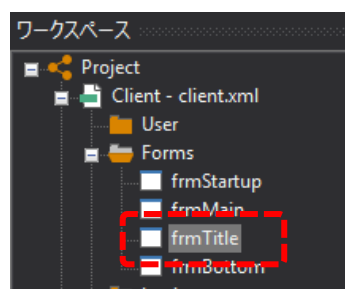
代表警報のランプを作成する

続いて、代表警報の状態を表示するためのランプを作成します。

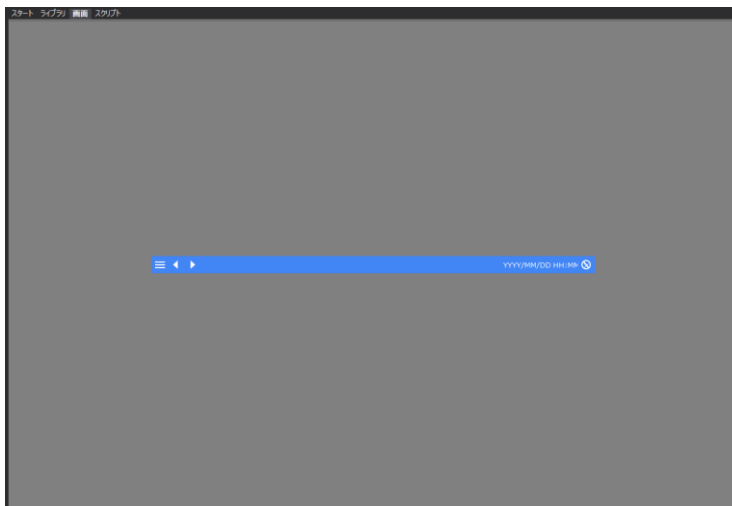
今回は例として、標準フレームワークの画面上部に常時表示されているタイトルエリアにランプを配置してみます。

1. 画面上部のタイトルエリアのフォームを選択する

Panel Editor のワークスペースのツリーから、Forms フォルダの下にある「frmTitle」を選択してください。



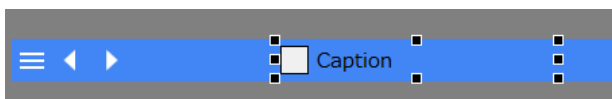
frmTitle フォームが表示されました。



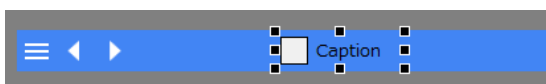
このフォーム上に部品ライブラリのランプスイッチ部品を配置して、代表アラームの表示を追加します。

2. 部品ライブラリの部品を張り付ける


Panel Editor のライブラリタブから、部品ライブラリのランプスイッチ部品を表示し、「lampswitch123」 部品を選択してフォームに 1 つ張り付けてください。



フォームに張り付けたら、部品のサイズ調整を行ってください（※Ctrl キーを押しながら部品をダブルクリックすると、部品のグループを編集モードにすることができます）。ここでは例として、Caption のラベルの幅を狭く調整し、フォントサイズを 10 ポイントに変更しています。



部品のサイズ調整が終わったら、ダブルクリックで設定ダイアログを表示して、以下を参考に設定してください。



イメージ	設定項目	設定値
	タグ設定／入力タグ	\$M1000
	表示設定／キャプション	軽故障
	表示設定／ON 色	@FAST,@AlarmColor
	表示設定／OFF 色	@NullColor
	表示設定／キャプション色	#FFFFFF

上記の設定で入力タグに指定した \$M1000 は、アラームステートモニタで軽故障の代表信号の出力先に指定した内部タグです。また、ON 色に指定した書式は、部品のアラーム色にデフォルトで設定されている「@FAST,@AlarmColor」をそのままコピーして貼り付ければ OK です。また、キャプションの色には白を固定色で指定しています。

上記の軽故障ランプの設定が終わったら、軽故障ランプをコピーして2つ並べて張り付けてください。



コピーして張り付けたら、異なる設定箇所のみ変更します。

イメージ	設定項目	設定値
	タグ設定／入力タグ	\$M1001
	表示設定／キャプション	中故障
	表示設定／ON 色	(軽故障と同じ)
	表示設定／OFF 色	(軽故障と同じ)
	表示設定／キャプション色	(軽故障と同じ)
	タグ設定／入力タグ	\$M1002
	表示設定／キャプション	重故障
	表示設定／ON 色	(軽故障と同じ)
	表示設定／OFF 色	(軽故障と同じ)
	表示設定／キャプション色	(軽故障と同じ)

以上で設定作業は完了です。

動作確認

それでは、動作を確認してみましょう。

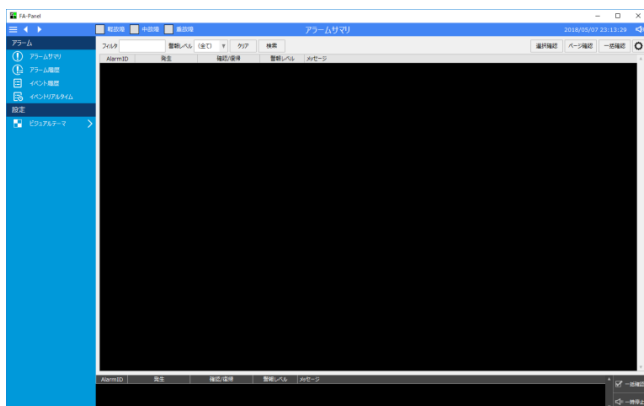
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



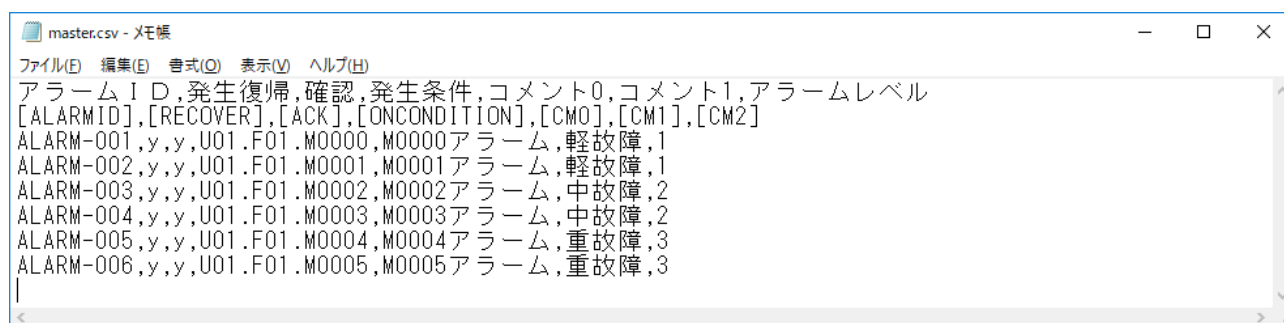
Panel Browser が起動したら、アラームサマリ画面を表示してください。



画面上部のタイトル部分に、先ほど作成した代表警報ランプが配置されていることを確認してください。
現状は一件も警報が発生していないため、すべて消灯状態となっています。

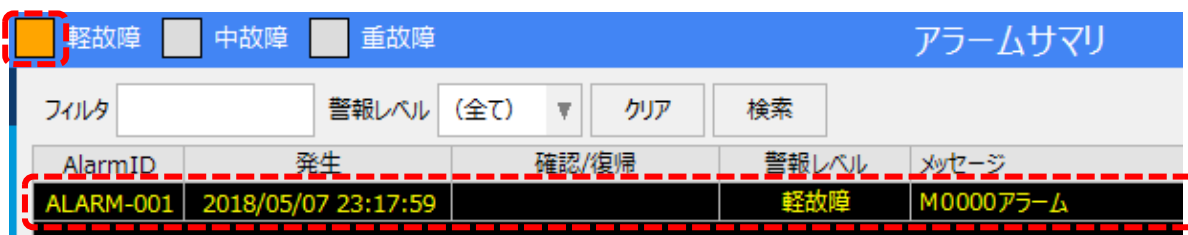


尚、教材サンプルのアラームマスタには、あらかじめ以下のようなアラームが設定されており、U01.F01.M0000 から M0005 までのタグがアラームとして登録されています。



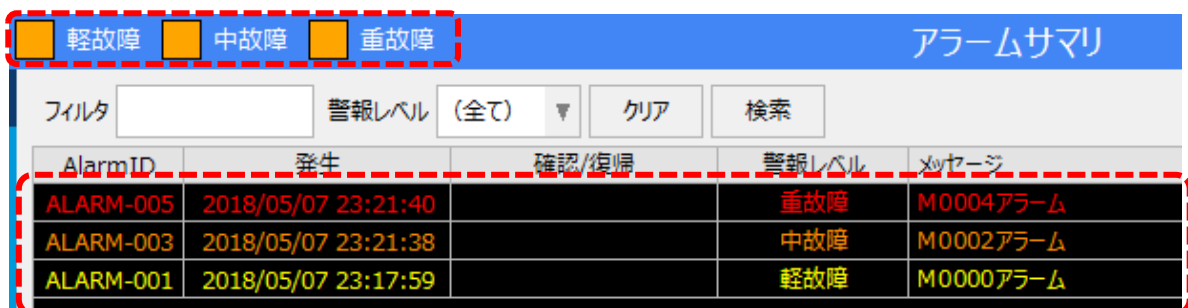
Panel Server から、U01.F01.M0000 に True を書き込んでください。

すると、アラームサマリに M0000 の軽故障が表示されるとともに、軽故障の代表警報ランプが点滅表示になります。



続いて、Panel Server から M0002、M0004 にも True を書き込んでみてください。

すると、アラームサマリに M0002 の中故障および M0004 の重故障が表示されるとともに、すべての代表警報ランプが点滅表示になりました。



続いて、Panel Server から M0000、M0002、M0004 を False に戻してください。

すると、アラームサマリの3件の警報は確認待ちの状態となりましたが、すべての代表警報ランプは点滅表示を継続しています。

アラームサマリ				
<div>軽故障 中故障 重故障</div>				
フィルタ	警報レベル	(全て)	クリア	検索
AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-005	2018/05/07 23:21:40	2018/05/07 23:24:54	重故障	M0004アラーム
ALARM-003	2018/05/07 23:21:38	2018/05/07 23:24:54	中故障	M0002アラーム
ALARM-001	2018/05/07 23:17:59	2018/05/07 23:24:54	軽故障	M0000アラーム

それでは、アラームサマリ画面からアラームを1行ずつ選択して、「選択確認」ボタンをクリックしてみます。
ここでは、重故障のM0004 アラームの行から選択確認してみます。

アラームサマリ				
<div>軽故障 中故障 重故障</div>				
フィルタ	警報レベル	(全て)	クリア	検索
				選択確認
AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-005	2018/05/07 23:21:40	2018/05/07 23:24:54	重故障	M0004アラーム
ALARM-003	2018/05/07 23:21:38	2018/05/07 23:24:54	中故障	M0002アラーム
ALARM-001	2018/05/07 23:17:59	2018/05/07 23:24:54	軽故障	M0000アラーム

すると、復帰および確認が完了した重故障のランプが消灯しました。

アラームサマリ				
<div>軽故障 中故障 重故障</div>				
フィルタ	警報レベル	(全て)	クリア	検索
AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/07 23:21:38	2018/05/07 23:24:54	中故障	M0002アラーム
ALARM-001	2018/05/07 23:17:59	2018/05/07 23:24:54	軽故障	M0000アラーム

他の警報についても1件ずつ選択確認を行い、すべての代表警報ランプが消灯することを確認してください。

<div>軽故障 中故障 重故障</div>		
------------------------	--	--

このように、アラームステートモニタを利用することによって、代表警報信号などのアラームの状態を簡単にタグに出力する事ができるようになります。内部タグに出力した代表警報は、今回の例のようにクライアント側でのランプ表示などで使用することができます。

■アラームステートモニタの設定

アラームステートモニタに定義する書式は、以下の仕様に従って定義します。

タグ名, 状態名, 監視対象アラーム, グループ条件

例えば、今回の例では、以下のような記述を行っています。

```
$M1000, LEVEL1, notack_or_notrecover, [CM2]==1  
$M1001, LEVEL2, notack_or_notrecover, [CM2]==2  
$M1002, LEVEL3, notack_or_notrecover, [CM2]==3
```

上記の1行目では、CM2の定義が1のアラーム（つまり、軽故障のアラーム）のうち、アラーム発生または未確認のアラームが1件でも存在する場合に、\$M1000 タグに True が書き込まれます。

ここで、監視対象アラームには以下のような指定も可能です。

notack . . . 未確認のアラームが1つでも存在する場合に True を書き出す。

notrecover . . . 未復帰のアラームが1つでも存在する場合に True を書き出す。

例えば確認の状態のみを見たい、あるいはアラームの実信号の復帰状態のみを見たいケースなど、必要に応じて使い分けることができます。



アラームステートモニタの出力タグには、かならず内部タグを指定するようにしてください。PLC と接続するタグを指定すると、PLC に対する書き込みが頻繁に発生する可能性があるため、システムの安定稼働に悪影響を及ぼす危険があります。



概要

アラームサーバアクションには、「状態監視」と呼ばれる機能が用意されています。

アラームサーバの状態監視機能は、クライアント側で用意されているアラームステートモニタのサーバ版機能であり、アラームの検出状態を Panel Server 側でモニタリングを行い、アラームの状態をサーバ側の任意のタグに書き出すことができます。この機能を利用する事により、メモリデバイスなどのタグに代表警報信号を出力することができます。クライアント側からは当該タグをバインド参照する等により、代表警報ランプの表示などに使用することができます。

このセクションでは、標準のアラーム機能に対して以下のカスタマイズを行います。

- メモリデバイスユニットを追加し、代表警報用のタグを用意する。
- アラームサーバアクションの状態監視機能を設定し、代表警報信号を作成する。
- 画面上に代表警報のランプ表示を追加する。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

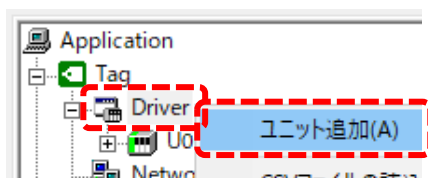
¥¥fc01¥Lesson3-5¥Lesson3-5.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

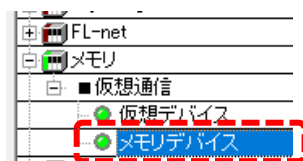
メモリデバイスユニットを追加する

教材プロジェクトのサーバ設定ファイル（server.txt）を Panel Server で開き、メモリデバイスユニットを追加してください。このユニットの下に代表警報信号を格納するためのタグを用意します。

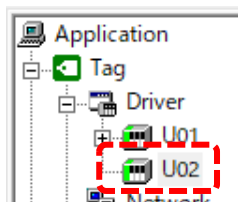
1. ユニットを追加する



サンプルプロジェクトが開かれた Panel Server の左ツリーから、Tag の下にある Driver を右クリックして「ユニット追加」を選択してください。



ドライバー一覧が表示されるので、一覧から、メモリ／仮想通信／メモリデバイスを選択して追加してください。



U02 ユニットが新たに追加されました。
今回は U02 という名称のまま使用します。

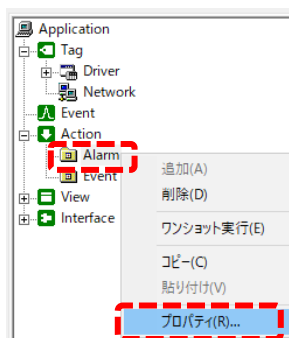
2. フォルダ、タグを作成する

U02 に「F01」フォルダを作成し、その下に以下の3点のタグを作成してください。

タグ名	デバイス	パラメータ	内容
U02.F01.ALARM_LEVEL1	ビット	変更不要 (N, 1)	軽故障の代表警報 (未確認および未復帰)
U02.F01.ALARM_LEVEL2	ビット	変更不要 (N, 1)	中故障の代表警報 (同上)
U02.F01.ALARM_LEVEL3	ビット	変更不要 (N, 1)	重故障の代表警報 (同上)

アラームサーバの状態監視機能を設定する

1. アラームサーバアクションの設定画面を開く

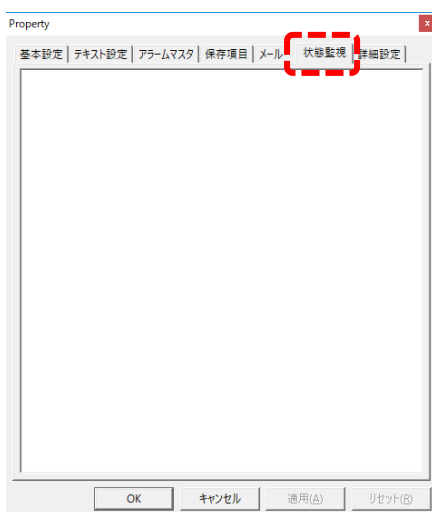


サンプルプロジェクトが開かれた Panel Server の左ツリーから、Action の下にある Alarm を選択し、右クリックでプロパティを開いてください。

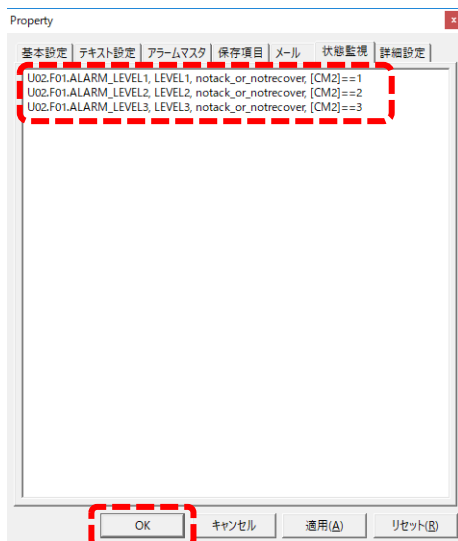
すると、アラームの設定ダイアログが表示されます。

2. アラームサーバアクションの設定画面を開く

設定ダイアログから、状態監視の設定を行います。



画面上部にある「状態監視」タブをクリックしてください。



設定画面の中央の入力エリアに、以下のように入力して OK ボタンをクリックしてください。

(設定内容)

```
U02.F01.ALARM_LEVEL1, LEVEL1, notack_or_notrecover, [CM2]==1
U02.F01.ALARM_LEVEL2, LEVEL2, notack_or_notrecover, [CM2]==2
U02.F01.ALARM_LEVEL3, LEVEL3, notack_or_notrecover, [CM2]==3
```

上記の設定により、CM2 に定義されているアラームレベル毎に、発生中または未確認のアラームが1つでも存在する場合に、一番目のパラメータで指定されたメモリデバイスのタグに True が書き込まれます。

- ・アラームレベル 1 (軽故障)・・・U02.F01.ALARM_LEVEL1
- ・アラームレベル 2 (中故障)・・・U02.F01.ALARM_LEVEL2
- ・アラームレベル 3 (重故障)・・・U02.F01.ALARM_LEVEL3

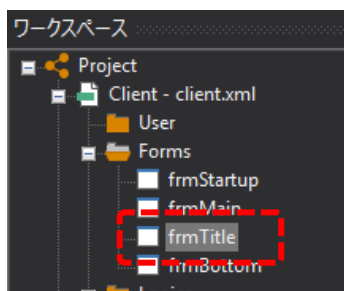
代表警報のランプを作成する

続いて、代表警報の状態を表示するためのランプを作成します。

今回は例として、教材サンプルプロジェクトの画面上部にあらかじめ配置されているランプ部品を使用します。

1. 画面上部のタイトルエリアのフォームを選択する

Panel Editor のワークスペースのツリーから、Forms フォルダの下にある「frmTitle」を選択してください。



frmTitle フォームを開くと、フォーム上にあらかじめ以下のようなランプスイッチ部品が配置されています。
このランプスイッチ部品をそのまま利用して、代表アラームのタグを割り付けてみます。



ダブルクリックで部品の設定ダイアログを表示して、以下を参考に設定してください。

イメージ	設定項目	設定値
	タグ設定／入力タグ	U02.F01.ALARM_LEVEL1
	タグ設定／入力タグ	U02.F01.ALARM_LEVEL2
	タグ設定／入力タグ	U02.F01.ALARM_LEVEL3

以上で設定作業は完了です。

動作確認

それでは、動作を確認してみましょう。

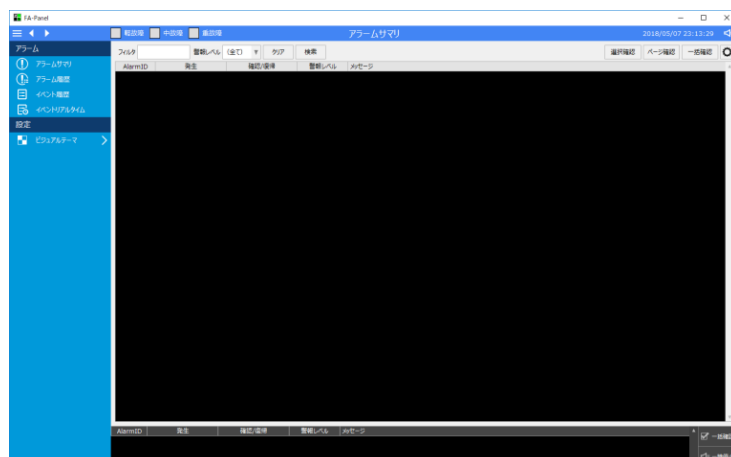
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



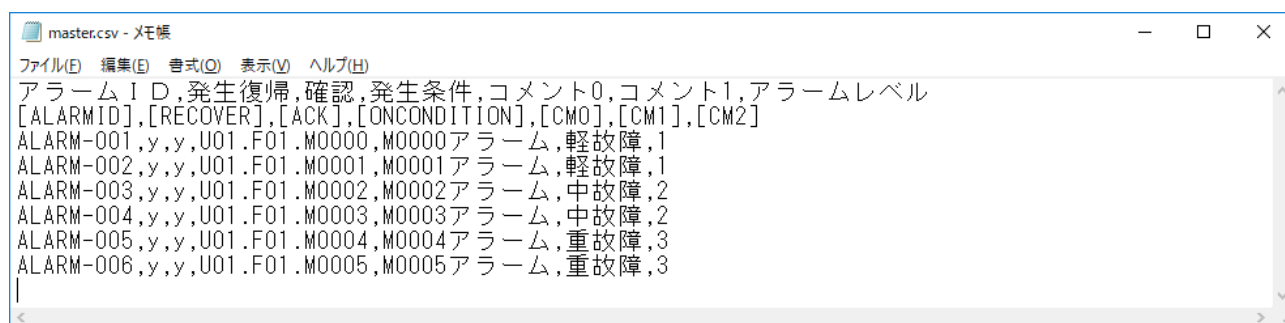
Panel Browser が起動したら、アラームサマリ画面を表示してください。



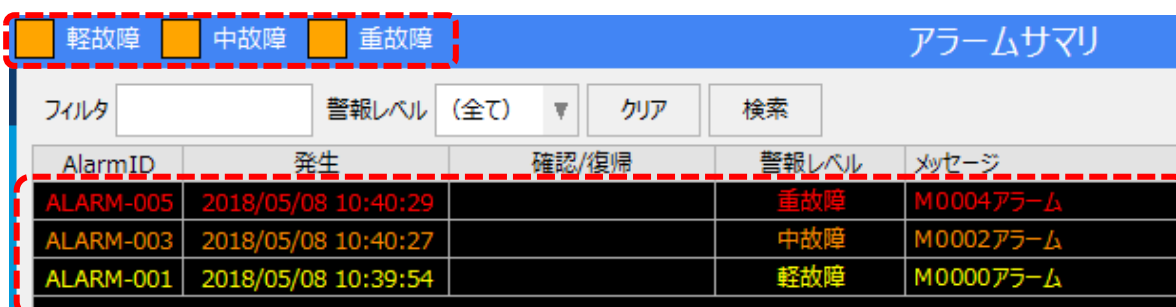
代表警報ランプは画面上部のタイトル部分に配置されています。
現状は一件も警報が発生していないため、すべて消灯状態となっています。



尚、教材サンプルのアラームマスタには、あらかじめ以下のようなアラームが設定されており、U01.F01.M0000 から M0005 までのタグがアラームとして登録されています。



Panel Server から、U01.F01.M0000 から M0005 の任意のタグに True を書き込んでアラームが発生させてみてください。すると、アラームサマリに各アラームが表示されるとともに、警報レベルに応じた代表警報ランプが点滅表示になります。



全てのアラームが復帰し、かつ確認が行われる事により、代表警報ランプが消灯することを確認してください。



このように、アラームサーバの状態監視機能を利用することによって、代表警報信号などのアラームの状態をメモリデバイスなどのサーバ側のタグに出力する事ができるようになります。

タグに出力した代表警報信号は、クライアント側でのランプ表示などで使用する他、代表警報の有無をトリガとした独自の制御の構築などに使用することができます。

■状態監視の設定

状態監視に定義する書式は、以下の仕様に従って定義します。

タグ名, 状態名, 監視対象アラーム, グループ条件

本書式は、リモートアラームグリッド等で設定可能なアラームステートモニタ機能と同様の書式です。

例えば、今回の例では、以下のような記述を行っています。

```
U02.F01.ALARM_LEVEL1, LEVEL1, notack_or_notrecover, [CM2]==1  
U02.F01.ALARM_LEVEL2, LEVEL2, notack_or_notrecover, [CM2]==2  
U02.F01.ALARM_LEVEL3, LEVEL3, notack_or_notrecover, [CM2]==3
```

上記の設定により、CM2 の定義が 1、2、3（つまり、軽故障、中故障、重故障）のアラームのうち、アラーム発生または未確認のアラームが 1 件でも存在する場合に、1 つめのパラメータで指定したタグに True が書き込まれます。



アラームサーバの状態監視機能の出力タグには、仮想デバイスやメモリデバイス等の、PLC と通信を行わないユニットのタグに書き込むようにしてください。PLC と接続するタグを指定すると、PLC に対する書き込みが頻繁に発生する可能性があるため、システムの安定稼働に悪影響を及ぼす危険があります。

■アラームステートモニタとの違い

アラームサーバの状態監視機能は、リモートアラームグリッド等のアラームステートモニタ機能と類似しています。両者の大きな違いは、アラームの発生状況の監視をクライアント側で行うのか、あるいはサーバ側で行うのかという点です。クライアント側とサーバ側のどちらの機能として実装するべきか、選択のポイントとしては以下のとおりです。

- サーバ側に実装する場合、代表警報などのアラーム監視の条件を Panel Server 側の一か所で定義する事により、システム全体でロジックを共通化することができます。一方、接続するクライアント毎に監視条件が異なるようなケースでは、個々のクライアント側で検出することで対応できます
- 両者では、使用できるアラームのフィールド名の種類に違いがあります。監視したい要件に対応したフィールドが利用可能な方を選択する必要があります。



アラームステートモニタとアラームサーバの状態監視では、使用可能なアラームのフィールド名の種類に仕様差があります。状態監視で使用可能なフィールド名の詳細については、オンラインマニュアルの「サーバ作成ガイド」「アクション編」「アクションリファレンス」「アラームサーバ」「フィールド名について」に記載されています。



概要

アラームマスタで定義可能なフィールドとして、復帰条件、発生待ちタイマー、復帰待ちタイマーがあります。

これらのフィールドを定義することにより、アラームの発生に対する復帰条件を個別に定義したり、アラーム発生時、復帰時の待ち時間を設定したりすることができるようになります。

このセクションでは、標準のアラーム機能に対して、復帰条件、発生待ちタイマー、復帰待ちタイマーの設定を行います。

■本件に関連するフィールド（抜粋）

項目	フィールド名	解説
発生条件	[ONCONDITION]	アラーム発生条件を記述します。条件が TRUE になった時にアラームが発生します。
復帰条件	[OFFCONDITION]	アラーム復帰条件を記述することができます。ここで指定した条件が FALSE になった時にアラームが復帰します。省略すると、上記「発生条件」の条件が FALSE になった時を復帰とみなします。
発生待ちタイマー	[ONTIMER]	アラーム発生までのタイマーです（秒）。指定した時間の間継続して TRUE の時、アラーム発生とします。
復帰待ちタイマー	[OFFTIMER]	アラーム復帰までのタイマーです（秒）。指定した時間の間継続して FALSE の時、アラーム復帰とします。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson3-6¥Lesson3-6.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

アラームマスタの編集

教材プロジェクトの以下のフォルダ配下にある「master.csv」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-6¥server¥alarm¥master.csv

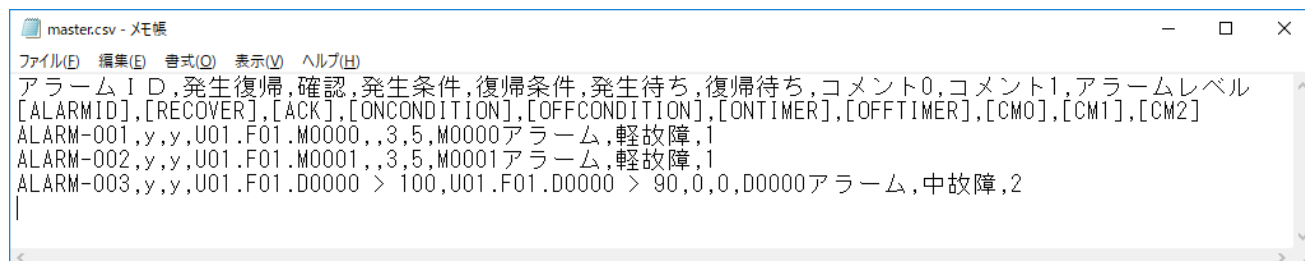
すると、以下のような内容が記述されています（※以下は、分かりやすいようにカンマの位置を調整しています）。

アラーム I D,	発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2]
ALARM-001,	y,	y,	U01.F01.M0000,	M0000 アラーム,	軽故障,	1
ALARM-002,	y,	y,	U01.F01.M0001,	M0001 アラーム,	軽故障,	1
ALARM-003,	y,	y,	U01.F01.D0000 > 100,	D0000 アラーム,	中故障,	2

上記に対して、発生条件（[ONCONDITION]）の後に、復帰条件（[OFFCONDITION]）、発生待ちタイマー（[ONTIMER]）、復帰待ちタイマー（[OFFTIMER]）の列を新たに挿入し、条件を設定します。ここでは以下のように編集してください。

アラーム I D, 発生復帰,	確認,	発生条件,	復帰条件,	発生待ち,	復帰待ち,	コメント 0,	コメント 1,	アラームレベル
[ALARMID], [RECOVER],	[ACK],	[ONCONDITION],	[OFFCONDITION],	[ONTIMER],	[OFFTIMER],	[CM0],	[CM1],	[CM2]
ALARM-001, y,	y,	U01.F01.M0000,	,	3,	5,	M0000 アラーム,	軽故障,	1
ALARM-002, y,	y,	U01.F01.M0001,	,	3,	5,	M0001 アラーム,	軽故障,	1
ALARM-003, y,	y,	U01.F01.D0000 > 100,	U01.F01.D0000 > 90,	0,	0,	D0000 アラーム,	中故障,	2

以下は、編集後のイメージです（メモ帳で表示した例）。



上記の編集内容を要約すると、以下のとおりです。

- M0000・・・OFFCONDITION：なし、発生待ち：3 秒、復帰待ち：5 秒
- M0001・・・OFFCONDITION：なし、発生待ち：3 秒、復帰待ち：5 秒
- D0000・・・OFFCONDITION：90 以下で復帰、発生待ち：0 秒、復帰待ち：0 秒

尚、復帰条件は、記述した条件式が FALSE になった時をアラーム復帰とみなします。条件式が成立した時（TRUE になった時）という解釈ではない点に注意してください。

動作確認

それでは、動作を確認してみましょう。

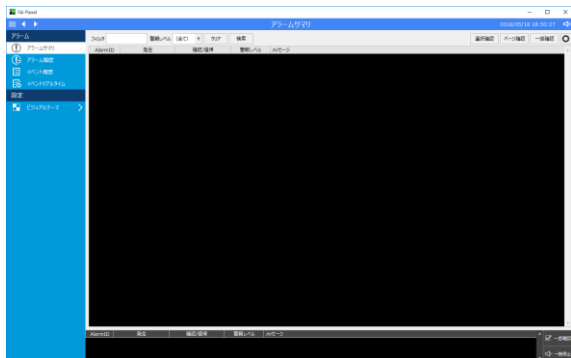
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



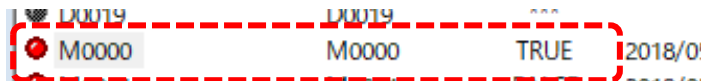
Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動したら、アラームサマリ画面を表示してください。



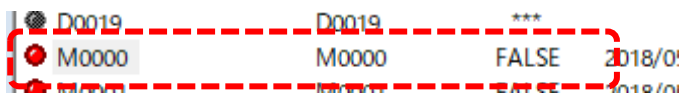
Panel Server から、U01.F01.M0000 に TRUE を書き込んでみてください。



すると、発生待ちタイマーの設定により、タグに TRUE を書き込んだ 3 秒後にアラームが発生することを確認してください。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-001	2018/05/18 18:56:37		軽故障	M0000アラーム

続いて、Panel Server から U01.F01.M0000 に FALSE を書き込んでみてください。



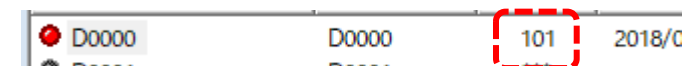
すると、復帰待ちタイマーの設定により、タグを FALSE に戻した 5 秒後にアラームが復帰することを確認してください。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-001	2018/05/18 18:56:37	2018/05/18 18:57:21	軽故障	M0000アラーム

このように、アラームの発生待ちタイマー、復帰待ちタイマーを指定することで、指定した時間（秒）の間、発生条件あるいは復帰条件が継続しているかどうかを判定に含めることができます。

※アラームサマリの画面から一括確認操作を行っておいください。

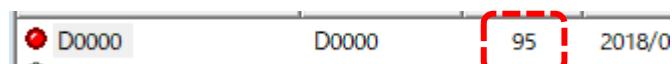
続いて、Panel Server から U01.F01.D0000 に 100 を超える数値を書き込んでみてください。



D0000 のアラーム発生条件（100 を超える場合）を満たし、発生待ちタイマーには 0 が指定されているため、アラームが即座に発生しました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/18 19:08:13		中故障	D0000アラーム

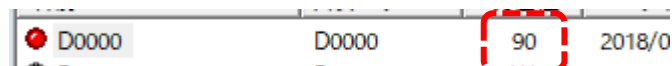
続いて、Panel Server から U01.F01.D0000 に 95 を書き込んでみてください。



D0000 のアラームの発生条件（100 を超える場合）は満たさない状態になりましたが、復帰条件（90 以下）を満たしていないため、アラームはまだ復帰しません。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/18 19:08:13		中故障	D0000アラーム

続いて、Panel Server から U01.F01.D0000 に 90 を書き込んでみてください。



すると、アラームの復帰条件（90 以下）を満たしたため、アラームが復帰しました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/18 19:08:13	2018/05/18 19:16:04	中故障	D0000アラーム

※確認が終わったら、アラームサマリの画面から一括確認操作を行ってください。

このように、アラームの復帰条件を指定することで、アラームの発生後に復帰とみなす際の値に幅を持たせることができます。特に液面などのレベルを数値で監視している場面などで、レベルが安全な範囲まで下がり切るまで復帰とみなしたくない場合などに利用することができます。



概要

イベント機能では、主に、監視対象設備における運転操作の履歴を記録します。標準のイベント機能では、アラームマスタに定義されたタグに対する発生条件に基づいてイベントの監視を行い、ログデータとして記録します。

そのほかにも、監視画面上からの操作内容をログデータに含めたいという場合があります。例えば、監視画面上に配置されたボタンの操作が行われたことを記録したい場合には、各ボタンの操作に応じてイベント記録用のタグに ON/OFF を書き込み、それらのタグをアラームマスタに登録しておくことによって操作ログとして記録する事もできます。しかしながら、この方法では画面操作の記録用にわざわざタグを用意する必要があり、システムの構築作業が煩雑になってしまいます。

そこで本セクションでは、標準のイベント機能に対して、スクリプトを利用して画面操作のイベントログを記録するための以下のカスタマイズについて学習します。

- アラームマスタ (Event) の編集
- バックグラウンドフォームに「リモートアラームクライアント」を張り付ける
- イベントを記録するスクリプトを記述する

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson3-7¥Lesson3-7.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

アラームマスタの編集 (Event)

教材プロジェクトの以下のフォルダ配下にある「master.csv」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-7¥server¥event¥master.csv

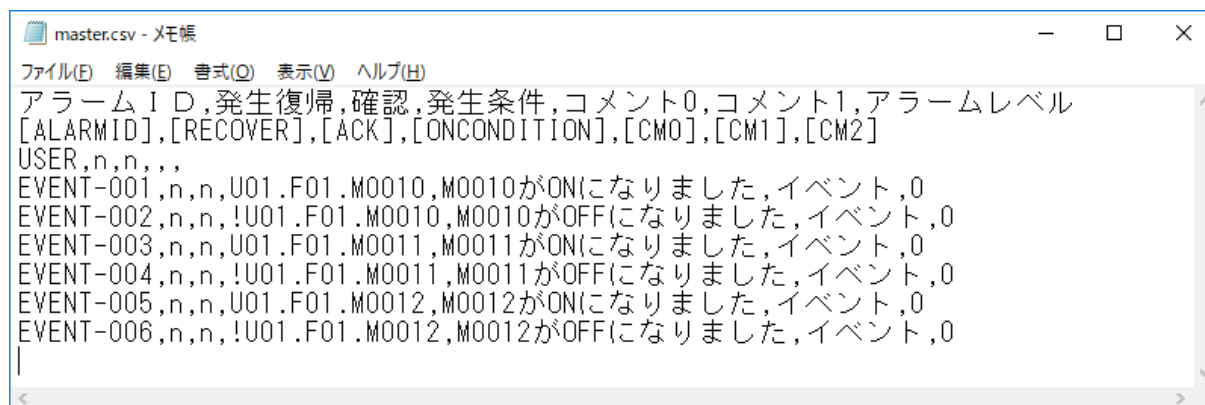
すると、以下のような内容が記述されています（※以下は、分かりやすいようにカンマの位置を調整しています）。

アラーム I D,	発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2]
EVENT-001,	n,	n,	U01.F01.M0010,	M0010 が ON になりました,	イベント,	0
EVENT-002,	n,	n,	!U01.F01.M0010,	M0010 が OFF になりました,	イベント,	0
EVENT-003,	n,	n,	U01.F01.M0011,	M0011 が ON になりました,	イベント,	0
EVENT-004,	n,	n,	!U01.F01.M0011,	M0011 が OFF になりました,	イベント,	0
EVENT-005,	n,	n,	U01.F01.M0012,	M0012 が ON になりました,	イベント,	0
EVENT-006,	n,	n,	!U01.F01.M0012,	M0012 が OFF になりました,	イベント,	0

上記に対して、3行目に新たな行を挿入し、以下の1行を記述してください。

USER, n , n , , ,

以下は、編集後のイメージです（メモ帳で表示した例）。



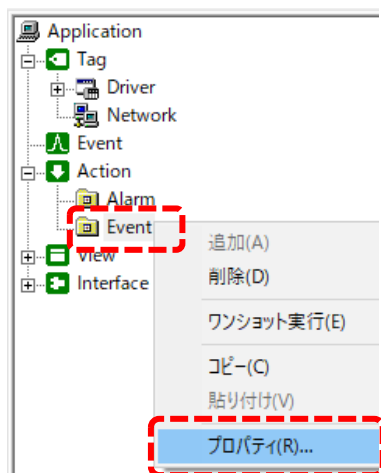
※カンマの数（列の数）が他の行と同じになるようにしてください。

※その他、詳しい内容は後ほど説明します。

保存項目の追加

続いて、教材プロジェクトのサーバ設定ファイル（server.txt）を Panel Server で開き、アラームサーバーアクションの「保存項目」に、保存が必要な項目の追加を行います。

1. アラームサーバーアクションの設定画面を開く

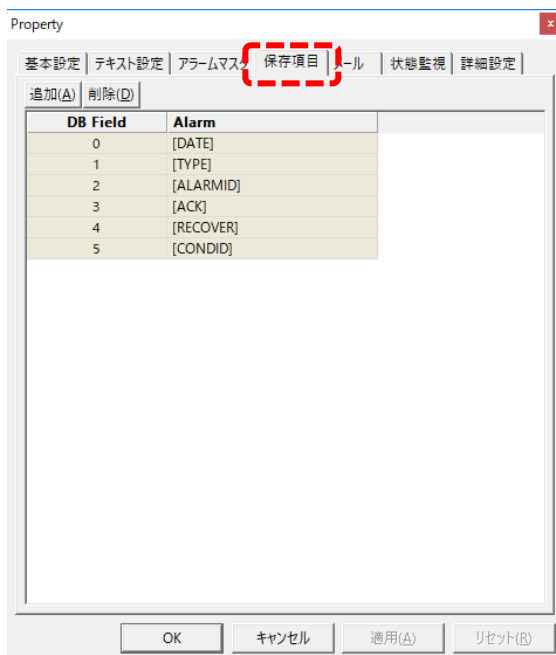


Panel Server の左ツリーから、Action の下にある Event を選択し、右クリックでプロパティを開いてください。

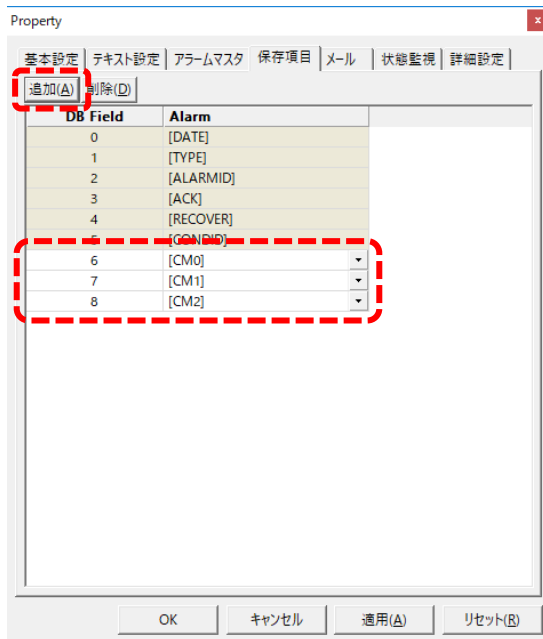
すると、アラームの設定ダイアログが表示されます。

2. 保存項目の追加

設定ダイアログから保存項目の設定を行います。



画面上部にある「保存項目」タブをクリックしてください。



画面から「追加」ボタンを3回クリックしてください。

すると左記のように、1回クリックするごとに新たな行が1行ずつ追加され、「[CM0]」「[CM1]」「[CM2]」が自動的に表示された状態となります。

今回の例では、保存項目に追加が必要なコメントはちょうどCM0、CM1、CM2 であるため、自動的に追加された内容をそのまま利用することとします。

表示内容を確認し、OK ボタンをクリックしてください。

この設定により、アラームのログデータに、CM0、CM1、CM2 の値が含まれて保存されるようになります。



アラーム履歴やイベント履歴などの画面表示では、保存項目に CM0、CM1、CM2 が含まれていなかった場合、アラームマスタの定義内容をもとに表示が行われます。しかしながら、今回行うカスタマイズでは、新たに追加したアラーム ID 「USER」のアラームマスタ内での定義内容に CM0、CM1、CM2 のいずれも存在していないため、上記の保存項目の設定画面から保存が必要な項目を追加する必要があります。この設定を行うことにより、CM0、CM1、CM2 の内容がアラームデータに含めて保存されるようになり、イベント履歴が正しく表示されるようになります。

リモートアラームクライアントを張り付ける

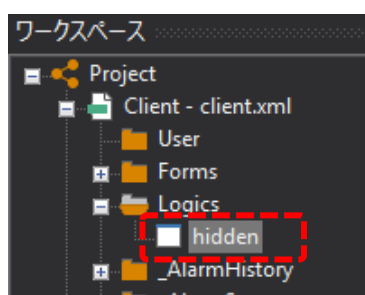
続いて、イベントの監視を行っているアラームサーバに接続するためのリモートアラームクライアントを用意します。

通常、アラームサマリなどの画面表示では、表形式の部品である「リモートアラームグリッド」コントロールが使用されています。今回使用する「リモートアラームクライアント」コントロールは、リモートアラームグリッドから表形式の表示機能が省かれたコントロールです。今回のようにアラームの表示を行わず、バックグラウンドでアラームの状態を監視したり、アラームサーバに接続して機能呼び出ししたりする場合に適しています。

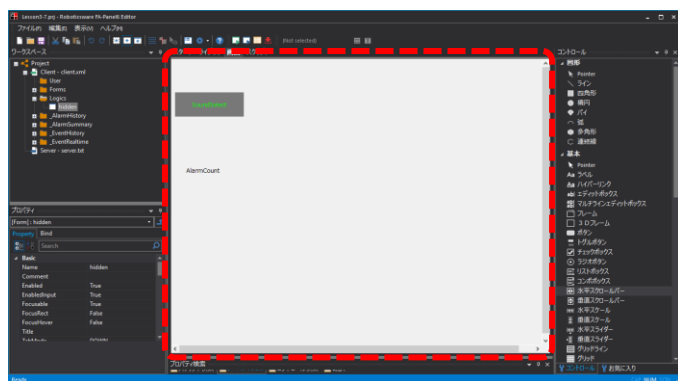
尚、この仕組みは、システムの実行中に常に存在しているフォーム上に用意する必要があります。今回の例では「hidden」フォームを使用することとします。

1. 「hidden」フォームを選択する

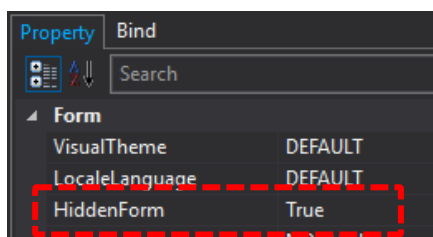
Panel Editor のワークスペースのツリーから、Logics フォルダの下にある「hidden」を選択してください。



hidden フォームが選択されました。



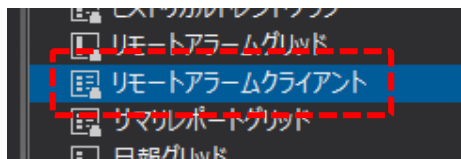
このフォームは標準フレームワークに最初から用意されているフォームの一つであり、システムの実行中に常にバックグラウンドで実行されるフォームです。



このフォームの「HiddenForm」プロパティを見てみると、左記のように「True」が設定されています。このプロパティが True に設定されているフォームは、自動的にバックグラウンドで実行された状態となります。

2. hidden フォームにリモートアラームクライアントを張り付ける

Panel Editor のコントロールから「リモートアラームクライアント」を選択し、hidden フォームに張り付けてください。

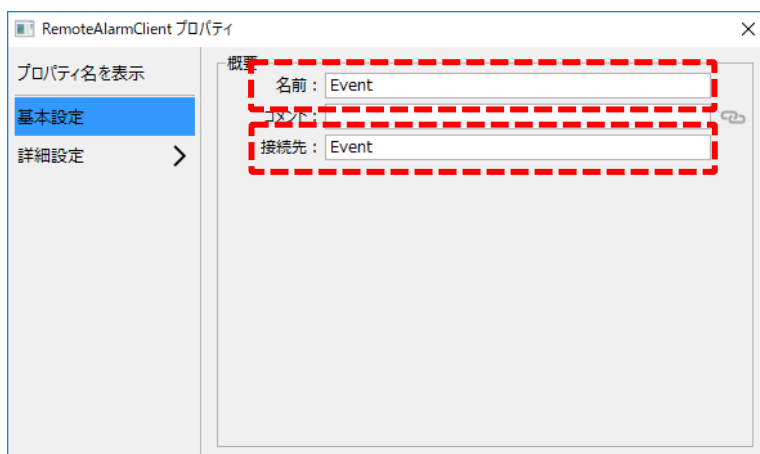


フォームに張り付けられました。



3. リモートアラームクライアントのプロパティ設定

hidden フォームに張り付けたリモートアラームクライアントに対し、以下のプロパティを設定してください。



名前 (Name プロパティ) :
Event

接続先 (AlarmServer プロパティ) :
Event

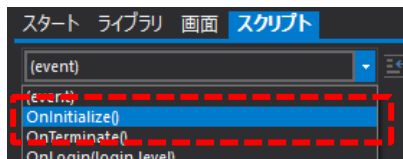
尚、上記で設定する「接続先」には、Panel Server 側で登録されているイベント監視用のアラームサーバーアクションの名称を指定します。リモートアラームクライアントは、実行時に上記で指定したアクションに対して接続します。

4. リモートアラームクライアントへのスクリプト記述

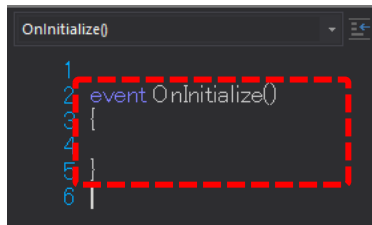
リモートアラームクライアントを選択した状態で、スクリプトページを開いてください。



スクリプトページの左上にあるイベントのコンボボックスを開き、「OnInitialize」を選択します。



すると、OnInitialize イベントが追加されました。



OnInitialize イベントに、以下のコードを記述してください。

(記述内容)

```
event OnInitialize()  
{  
    this.Connect();  
}
```

- ※ コードはイベントの「{ 」と「 }」の間に記述します。
- ※ すべて半角で記述します。
- ※ this の先頭にある余白表現を「字下げ」と呼びます。字下げを行うには Tab キーを押します（半角スペースを数個入力してもかまいません。全角スペースは入力しないでください）。

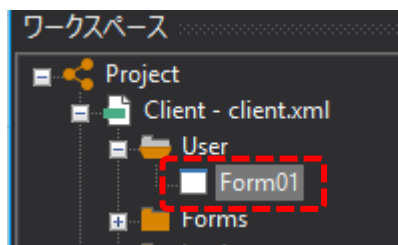
尚、上記のように Connect を呼び出すことで、AlarmServer プロパティで指定したアラームサーバーアクションに対して接続が行われます。

画面操作のイベントを記録する

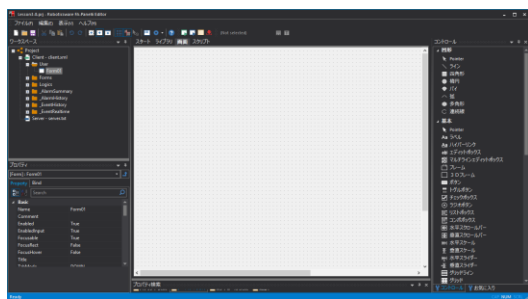
続いて、フォームの編集を行います。

1. Panel Editor で Form01 フォームを選択する

サンプルプロジェクトが開かれた Panel Editor の左ツリーから、Client の下にある User フォルダを開き、「Form01」フォームを選択してください。



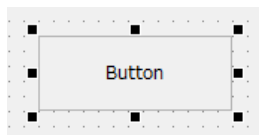
(Form01)



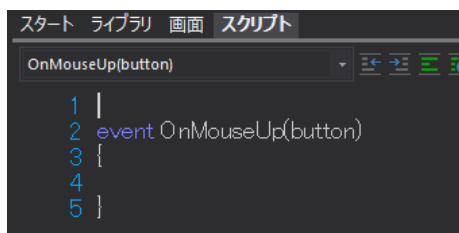
このフォームは教材サンプルにあらかじめ含まれているフォームです。フォームの呼び出し処理（_MenuConf プロパティ）の設定もすでに登録されています。このフォームはブランクの状態となっており、まだ何もオブジェクトは配置されていません。今回はこのフォームにボタンを配置して、ボタンのクリック操作をイベント履歴として記録する事とします。

2. Form01 フォームにボタンを配置して処理を記述する

Form01 フォームに、ボタンコントロールを1つ張り付けてください。



張り付けたボタンのスクリプトページを表示し、イベントのコンボボックスから「OnMouseUp」イベントを追加してください。



OnMouseUp イベントに、以下を参考にコードを記述してください。

尚、このイベントは、ボタンをマウスでクリックする時（実際には、マウスボタンをクリックして離れた時）に呼び出されます。

（記述内容）

```
event OnMouseUp(button)
{
    ::hidden.Event.FireAlarm("USER", 1, c(0,1,2), c("ボタンが操作されました","イベント",0));
}
```

上記では、リモートアラームクライアントの「FireAlarm」というメソッドを呼び出しています。この処理が実行されると、メソッドで指定したメッセージがイベントとして記録されます。

（記述例）

```
1
2 event OnMouseUp(button)
3 {
4     ::hidden.Event.FireAlarm("USER", 1, c(0,1,2), c("ボタンが操作されました","イベント",0));
5 }
6
```

以上で設定作業は完了です。

動作確認

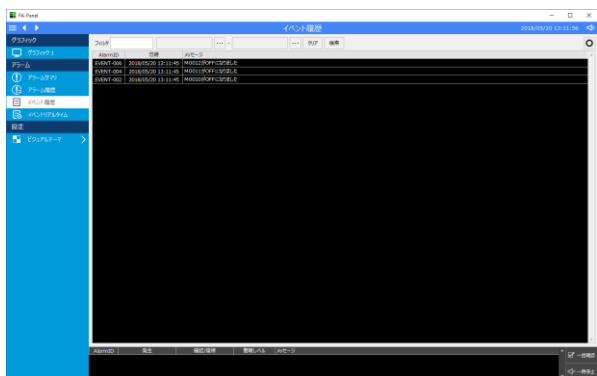
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動しました。

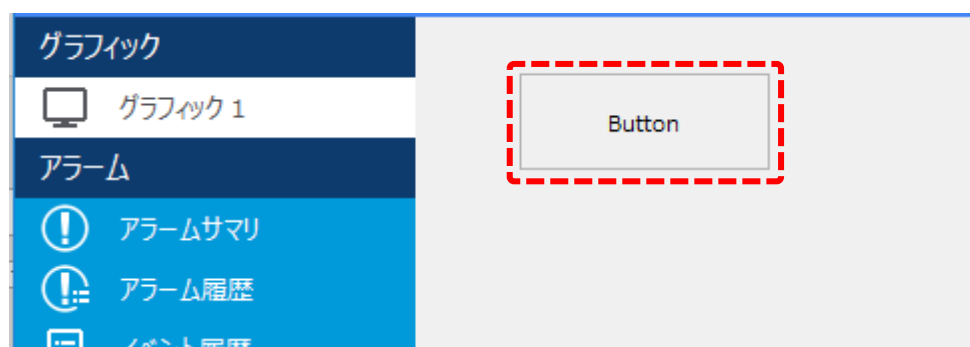


イベント履歴画面を見てみると、以下のような履歴が表示された状態となっています。

AlarmID	日時	メッセージ
EVENT-006	2018/05/20 13:11:45	M0012がOFFになりました
EVENT-004	2018/05/20 13:11:45	M0011がOFFになりました
EVENT-002	2018/05/20 13:11:45	M0010がOFFになりました

それでは、作成したボタンの動作を見てみましょう。

メニューから「グラフィック 1」を選択して Form01 を表示し、表示されたフォーム上にあるボタンを 1 回、クリックしてください。



ボタンをクリックしたら、イベント履歴画面を表示し、ボタンクリックによるメッセージが表示されることを確認してください。

AlarmID	日時	メッセージ
USER	2018/05/20 13:15:07	ボタンが操作されました
EVENT-006	2018/05/20 13:11:45	M0012がOFFになりました
EVENT-004	2018/05/20 13:11:45	M0011がOFFになりました
EVENT-002	2018/05/20 13:11:45	M0010がOFFになりました

引き続き、Form01 に戻ってボタンを複数回クリックしてみてください。

すると、ボタンをクリックした回数だけイベント履歴が新たに生成されました。

AlarmID	日時	メッセージ
USER	2018/05/20 13:17:12	ボタンが操作されました
USER	2018/05/20 13:17:12	ボタンが操作されました
USER	2018/05/20 13:17:12	ボタンが操作されました
USER	2018/05/20 13:17:11	ボタンが操作されました
USER	2018/05/20 13:17:11	ボタンが操作されました
USER	2018/05/20 13:15:07	ボタンが操作されました

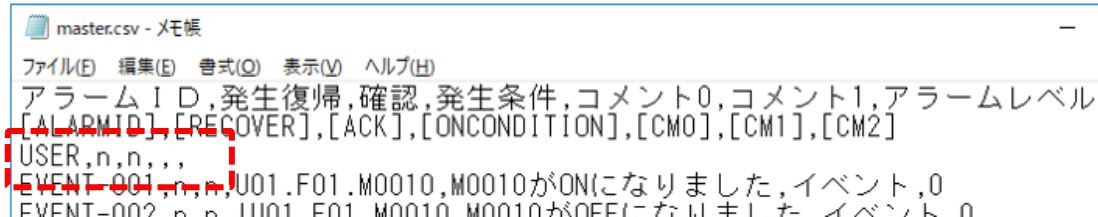
このように、スクリプトによる処理の実行により、イベント履歴にログを記録することができます。

■FireAlarm メソッドについて

リモートアラームクライアント（およびリモートアラームグリッド）では、FireAlarm メソッドを呼び出すことにより、スクリプトからアラームを発生させることができます。

FireAlarm を利用する場合、まず、アラームマスタに FireAlarm 用のアラーム ID を 1 行定義します。
今回の例では「USER」というアラーム ID を使用しました。

この定義行では、アラーム ID、発生復帰：n、確認：n とし、他の列はすべてブランク（カンマのみ）としておきます。



```

master.csv - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
アラーム ID,発生復帰,確認,発生条件,コメント0,コメント1,アラームレベル
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]
USER,n,n,,
EVENT-001,n,n,U01.F01.M0010,M0010がONになりました,イベント,0
EVENT-002,n,n,U01.F01.M0010,M0010がOFFになりました,イベント,0
  
```

そして、アラームを発生させたいスクリプトの場所から、目的のアラームサーバに接続されているリモートアラームクライアント（またはリモートアラームグリッド）を介して FireAlarm メソッドを呼び出すことによって、手動によるアラームが記録されます。

```

event OnMouseUp(button)
{
    ::hidden.Event.FireAlarm("USER", 1, c(0,1,2), c("ボタンが操作されました","イベント",0));
}
  
```

尚、上記の例では、「::hidden.Event.FireAlarm()」とすることで、hidden 上に配置されているリモートアラームクライアント（Event）の FireAlarm メソッドが呼び出されます。



FireAlarm メソッドの引数の意味などの詳細は、オンラインマニュアルの「コントロールリファレンス」「データ」「リモートアラームグリッドコントロール」「メソッド」「FireAlarm」に記述されています。



スクリプトの記述は、ユーザー定義関数を使用することにより、よりスマートに記述することができるようになります。例えば、ルートや hidden フォームのスクリプトにイベント書き出し用の関数を作成しておき、イベントを書き出したい処理の箇所からはその関数を呼び出すようにするなどの記述が可能です。

ユーザー定義関数について知りたい場合は、オンラインマニュアルの「スクリプトガイド」「スクリプト Ver2(SC2 構文)」「ユーザー定義関数」に記述されています。

■ユーザー定義関数による拡張例

//ボタンのスクリプト

event OnMouseUp(button)

```
{
    ::gusrFireAlarm(" ボタンが操作されました"); //メッセージのみを関数に渡す
}
```

//root のスクリプト

function gusrFireAlarm(msg)

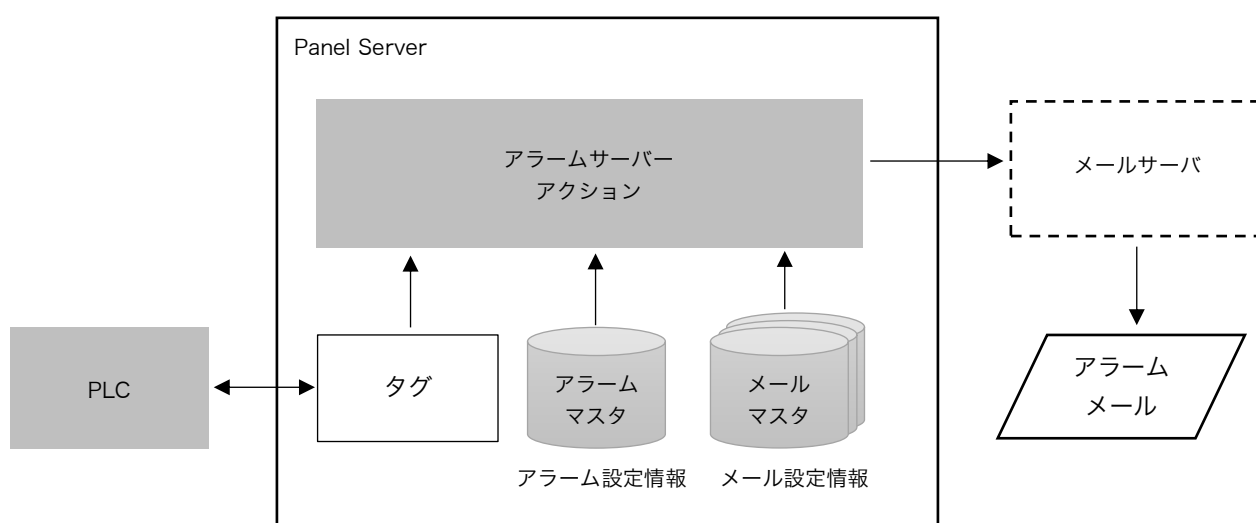
```
{
    ::hidden.Event.FireAlarm("USER", 1, c(0, 1, 2), c(msg, " イベント", 0)); //受け取ったメッセージを FireAlarm に渡す
}
```

概要

FA-Panel のアラーム機能は、アラームメールを送信することができます。

アラームメールを送信する場合、メールマスタと呼ばれるテキスト形式の設定ファイルに対してメール送信のための設定情報を定義するとともに、アラームマスタに対してメール送信用のフィールドを定義する必要があります。

また、アラームメールの送信はメールサーバを介して行われます。アラームサーバーアクションの設定画面から、いずれかのプロバイダのメールサーバと接続するためのメールサーバ名、ポート番号、アカウント、パスワードなどを、メールサーバの認証方法にあわせて設定する必要があります。尚、FA-Panel にはメールサーバとしての機能は内蔵していません。別途、メールサーバを用意する必要があります。



このセクションでは、標準のアラーム機能に対して、アラームメールを送信するためのカスタマイズ方法について学習します。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson3-8¥Lesson3-8.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

尚、アラームメールの送信には、FA-Panel とは別に、別途メールサーバが必要となります。メールの送信が行えるメールサーバと接続できる環境を用意してください。

アラームマスタの編集

教材プロジェクトの以下のフォルダ配下にある「master.csv」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-8¥server¥alarm¥master.csv

すると、以下のような内容が記述されています（※以下は、分かりやすいようにカンマの位置を調整しています）。

アラーム I D,	発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2]
ALARM-001,	y,	y,	U01.F01.M0000,	アラーム 001,	軽故障,	1
ALARM-002,	y,	y,	U01.F01.M0001,	アラーム 002,	中故障,	2
ALARM-003,	y,	y,	U01.F01.M0002,	アラーム 003,	重故障,	3

アラームメールを送信する場合、アラームメールフィールド（[MAIL]）を定義する必要があります。

MAIL フィールドには、各アラームにおけるメールを送信するタイミングを定義することができます。メール送信のタイミングとは、発生時、復帰時、確認時などのアラームの各事象です。

キーワード	フィールド名
all	すべてのタイミングでメールを送信する。
new	発生時にメールを送信する。
recover	復帰時にメールを送信する。
ack	確認時にメールを送信する。
change	アラームの状態が変化したときにメールを送信する。
（ブランク）	本フィールドがブランクのアラームはメールを送信しない。

尚、上記の定義は、半角スペース区切りで複数列挙して記述することにより、組み合わせて定義することができます。例えば、発生時と復帰時に送信したい場合は、「new recover」のように定義します。

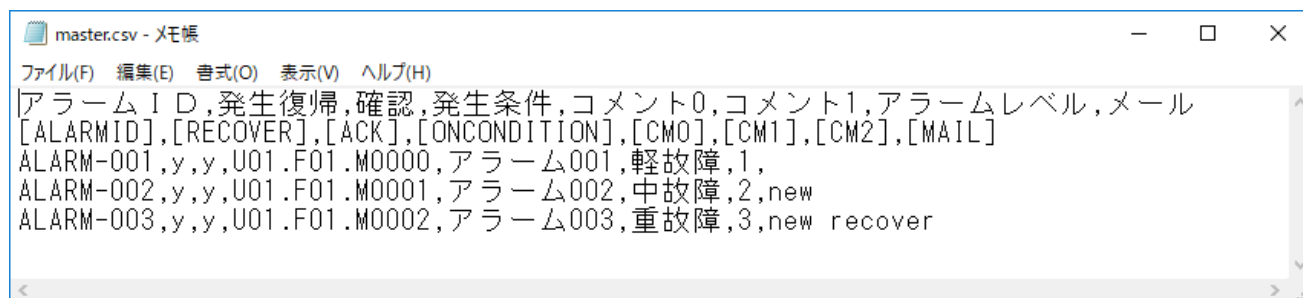
ここでは、アラームマスタを以下のように編集してください。

アラーム I D,	発生復帰,	確認,	発生条件,	コメント 0,	コメント 1,	アラームレベル,	メール
[ALARMID],	[RECOVER],	[ACK],	[ONCONDITION],	[CM0],	[CM1],	[CM2],	[MAIL]
ALARM-001,	y,	y,	U01.F01.M0000,	アラーム 001,	軽故障,	1,	
ALARM-002,	y,	y,	U01.F01.M0001,	アラーム 002,	中故障,	2,	new
ALARM-003,	y,	y,	U01.F01.M0002,	アラーム 003,	重故障,	3,	new recover

上記では、列の末尾に [MAIL] フィールドを追加して、個々のアラームのメール送信タイミングを定義しています。

- ・ ALARM-001…………… メール送信しない（ブランク）
- ・ ALARM-002…………… 発生時のみメールを送信（new）
- ・ ALARM-003…………… 発生時、復帰時にメールを送信（new recover）

以下は、編集後のイメージです（メモ帳で表示した例）。



```
master.csv - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
アラーム I D ,発生復帰,確認,発生条件,コメント0,コメント1,アラームレベル,メール
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2],[MAIL]
ALARM-001,y,y,U01.F01.M0000,アラーム001,軽故障,1,
ALARM-002,y,y,U01.F01.M0001,アラーム002,中故障,2,new
ALARM-003,y,y,U01.F01.M0002,アラーム003,重故障,3,new recover
```

メールマスタの編集

続いて、メールマスタの編集を行います。

教材プロジェクトの以下のフォルダ配下にある「mail_master.txt」をメモ帳などで開いてください。

¥¥fc01¥Lesson3-8¥server¥alarm_mail¥mail_master.txt

教材サンプルのメールマスタには以下のような内容が記述されています。

```
[MailCondition]

[MailSubject]
アラームメール

[MailSender]
xxx@xxx.co.jp

[MailSendToName]
xxx@xxx.co.jp
(end)

[MailSendToNameCC]
(end)

[MailSendToNameBCC]
(end)

[MailAttachedFile]
(end)

[MailHeader]
*****
alarm e-mail.
*****
(end)

[MailFooter]
bottom of e-mail.
(end)

[MailMessage]
[DATESTR:%Y/%m/%d %H:%M:%S] & " " & [CM1]
[ALARMID] & " " & [CM0] & " " & [TYPEJP]
"_"
(end)
```

ここでは、以下の部分を実際の環境にあわせて変更し、上書き保存してください。

[MailSender]

xxx@xxx.co.jp ←送信元メールアドレスを指定してください

[MailSendToName]

xxx@xxx.co.jp ←送信先メールアドレスを指定してください

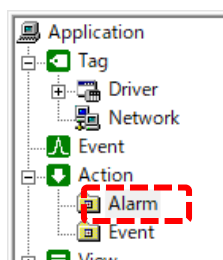
(end)

尚、動作確認が行えるように、送信先メールアドレスの設定には、自分自身の PC のメールアドレスなどの受信可能なアドレスを設定しておいてください（※メールマスタの詳細については後ほど説明します）。

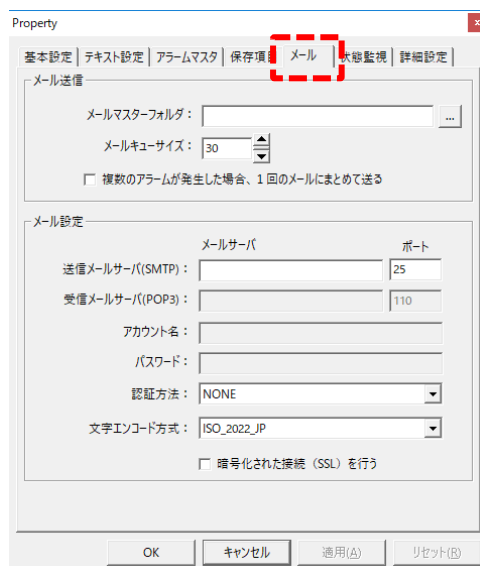
アラームサーバーアクションのメール設定

続いて、アラームサーバーアクションのメール設定を行います。

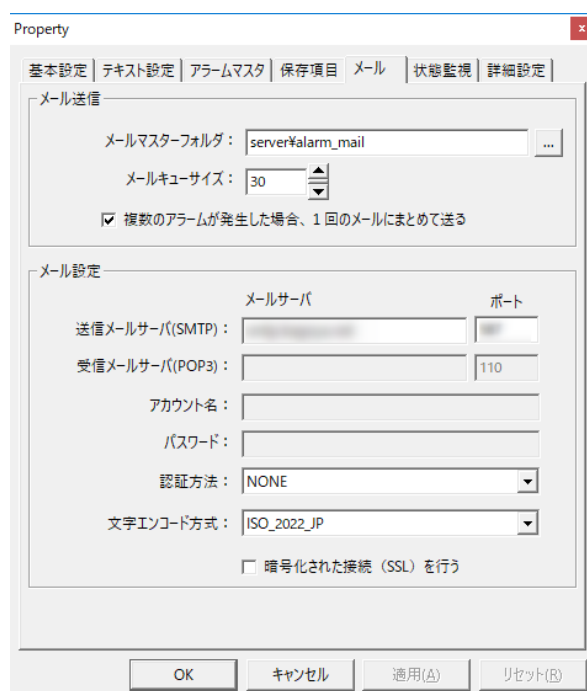
Panel Server から、Action の配下にある「Alarm」を選択し、プロパティを表示してください。



プロパティ設定画面から「メール」タブを選択します。



メールタブからメール設定を行います。



●まず、メールマスタのフォルダを指定します。
ここでは、先ほど編集したメールマスタが格納されているフォルダ「server¥alarm_mail」を指定してください。

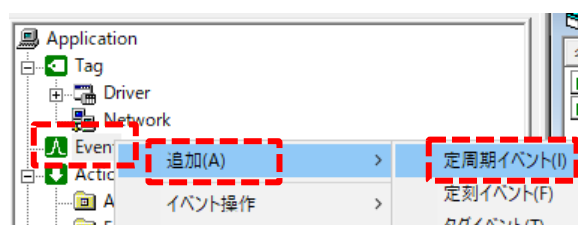
●「複数のアラームが発生した場合、一回のメールにまとめて送る」にチェックを入れておきます（※詳細は後ほど説明します）。

●メール設定の部分に、メールサーバとの接続に必要な情報を設定します。実際の環境にあわせて設定を行ってください。

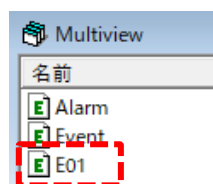
メール送信用の定周期イベントの追加

続いて、アラームサーバーアクションのメール送信を呼び出すための定周期イベントを設定します。

Panel Server のツリーから、Event を右クリックし、追加／定周期イベントを選択してください。

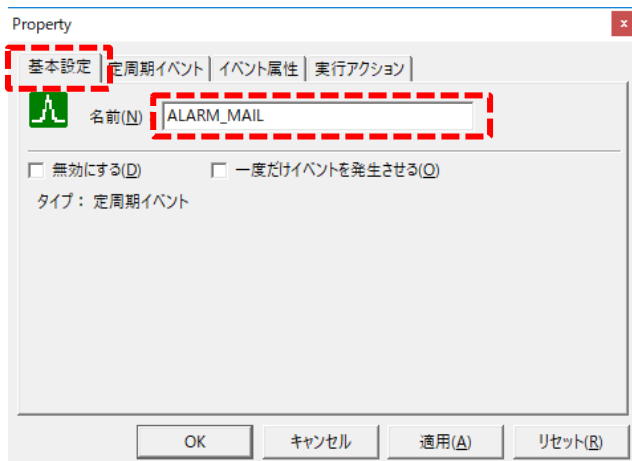


マルチビューに、「E01」というイベントが新たに追加されました。



E01 イベントのプロパティを表示してください。

基本設定のタブから、名前を分かりやすい名称に変更します。



Property

基本設定 | 定周期イベント | イベント属性 | 実行アクション

名前(N) ALARM_MAIL

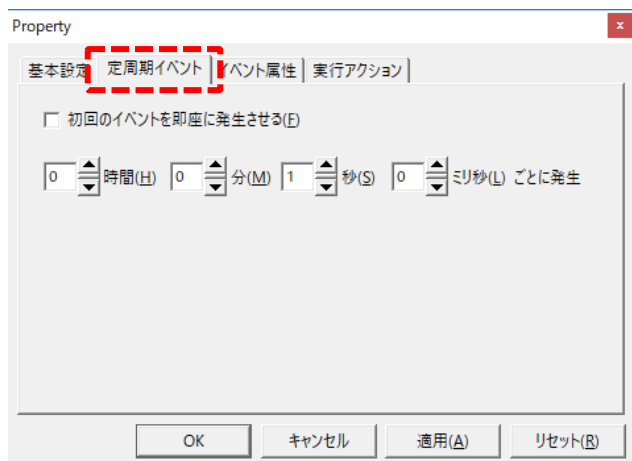
☐ 無効にする(D) ☐ 一度だけイベントを発生させる(O)

タイプ: 定周期イベント

OK キャンセル 適用(A) リセット(R)

ここでは例として、「ALARM_MAIL」としておきます。

続いて、定周期イベントの周期を確認します。



Property

基本設定 | 定周期イベント | イベント属性 | 実行アクション

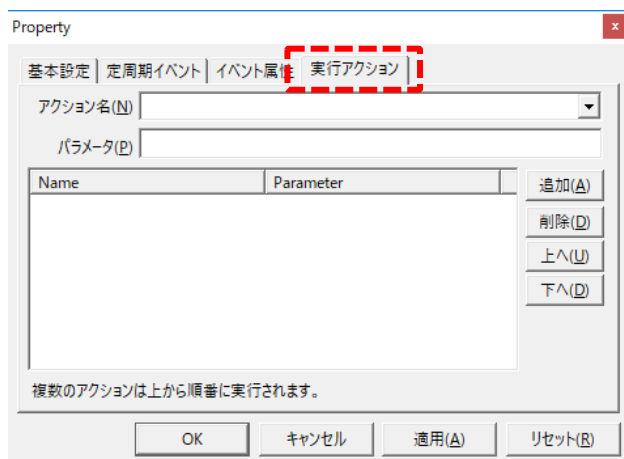
☐ 初回のイベントを即座に発生させる(E)

0 時間(H) 0 分(M) 1 秒(S) 0 ミリ秒(L) ごとに発生

OK キャンセル 適用(A) リセット(R)

今回はデフォルトの1秒周期のまま使用します。

続いて、実行アクションの設定を行います。



Property

基本設定 | 定周期イベント | イベント属性 | 実行アクション

アクション名(N) [Dropdown]

パラメータ(P) [Text]

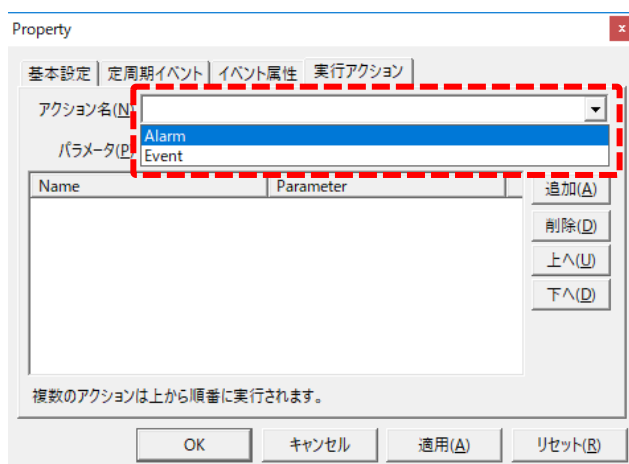
Name	Parameter
------	-----------

追加(A) 削除(D) 上へ(U) 下へ(D)

複数のアクションは上から順番に実行されます。

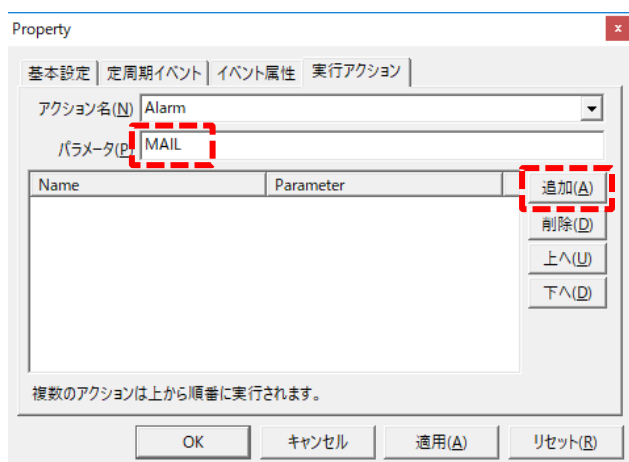
OK キャンセル 適用(A) リセット(R)

実行アクションタブを選択します。



アクション名のコンボボックスを開きます。このリストにはすでに登録されているアクションが一覧表示されます。

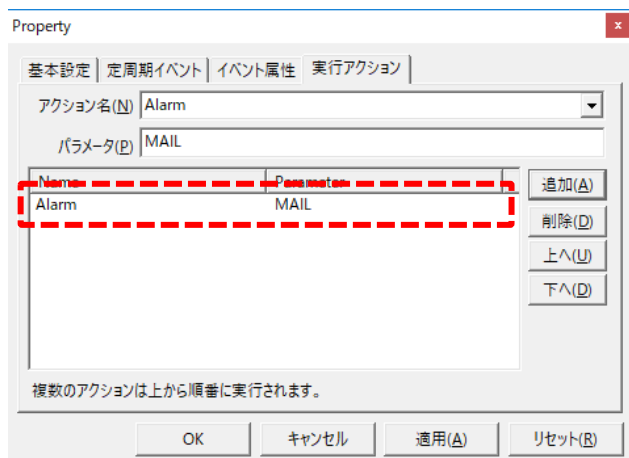
リストから「Alarm」アクションを選択してください。



続いて、パラメータを指定します。

パラメータに「MAIL」と入力し、「追加」ボタンをクリックしてください。

※すべて半角文字の大文字で入力してください。



すると、実行アクションのリストに、Alarm が追加され、パラメータに MAIL が設定されました。

OK ボタンで閉じてください。

以上で設定作業は完了です。

動作確認

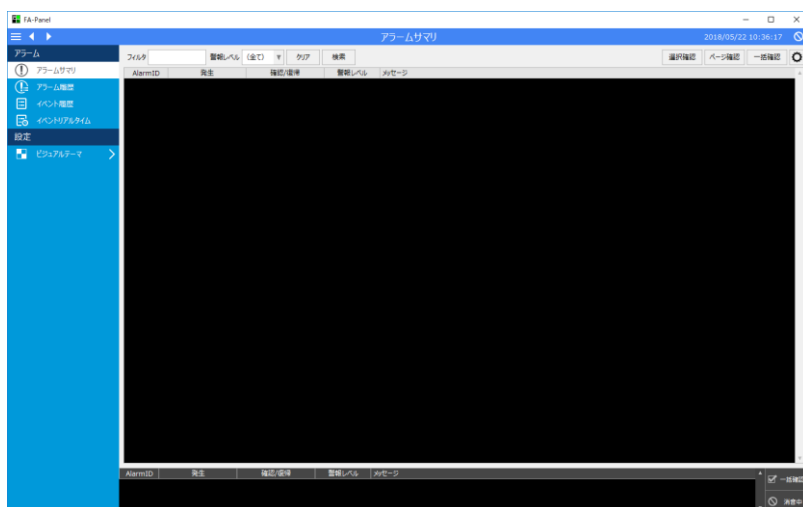
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



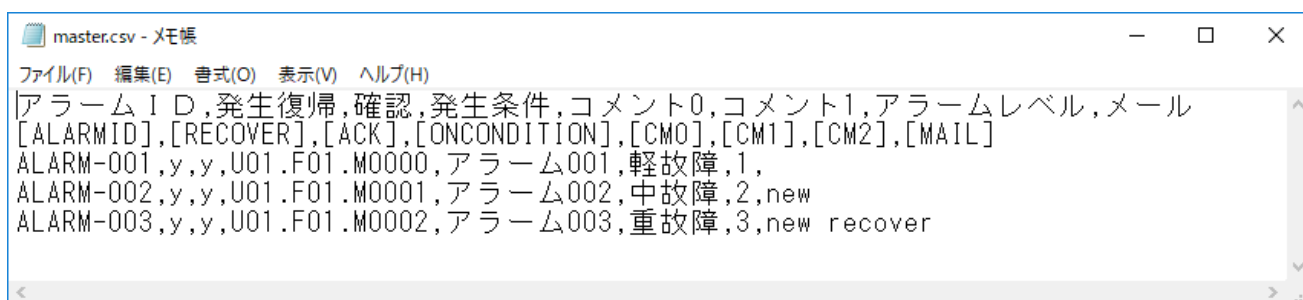
Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動したら、アラームの状態を確認できるようにアラームサマリ画面を表示しておきます。又、アラームメールを受信できる環境を準備しておいてください。



現状、アラームマスタには、以下のようなアラームが設定されており、U01.F01.M0000、M0001、M0002 までのタグがアラームとして登録されています。又、MAIL フィールドの設定では、M0000 はブランク（メール送信なし）、M0001 は new（発生時のみ）、M0002 は new recover（発生時および復帰時）として設定されています。



Panel Server から、U01.F01.M0000 に True を書き込んでください。

アラームサマリに M0000 の軽故障が表示されましたが、M0000 のアラームマスタの MAIL フィールドはブランク（送信しない）となっているため、メールは送信されません。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-001	2018/05/22 10:48:42		軽故障	アラーム001

※M0000 を False に戻し、画面から確認操作を行っておいてください。

続いて、Panel Server から、U01.F01.M0001 に True を書き込んでください。

アラームサマリに M0001 の中故障が表示されました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-002	2018/05/22 10:53:20		中故障	アラーム002

アラームメールとして以下の内容が送信されます。

メールの内容がアラームサマリの表示内容と一致していることを確認してください。

```
*****
alarm e-mail.
*****
2018/05/22 10:53:20 中故障
ALARM-002 アラーム 002 発生
--
bottom of e-mail.
```

続いて、Panel Server から、U01.F01.M0001 に False を書き込んでください。

アラームサマリに M0001 の中故障が復帰し、確認待ちの状態になりました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-002	2018/05/22 10:53:20	2018/05/22 10:57:22	中故障	アラーム002

M0001 のアラームマスタの MAIL フィールドは「new」となっているため、発生時のみメールが送信されます。上記の復帰操作によるメールは送られてこない事を確認してください。

※画面から確認操作を行っておいてください。

続いて、Panel Server から、U01.F01.M0002 に True を書き込んでください。

アラームサマリの M0002 の重故障が表示されました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/22 11:00:54		重故障	アラーム003

アラームメールとして以下の内容が送信されます。

メールの内容がアラームサマリの表示内容と一致していることを確認してください。

```
*****
alarm e-mail.
*****
2018/05/22 11:00:54 重故障
ALARM-003 アラーム 003 発生
--
bottom of e-mail.
```

続いて、Panel Server から、U01.F01.M0002 に False を書き込んでください。

アラームサマリの M0002 の重故障が復帰し、確認待ちの状態になりました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/22 11:00:54	2018/05/22 11:03:03	重故障	アラーム003

M0002 のアラームマスタの MAIL フィールドは「new recover」となっているため、発生時、復帰時それぞれのタイミングでメールが送信されます。今回の例では、以下のようなアラームメールが送信されます。
メールの内容がアラームサマリの表示内容と一致していることを確認してください。

```
*****
alarm e-mail.
*****
2018/05/22 11:03:03 重故障
ALARM-003 アラーム 003 復帰
--
bottom of e-mail.
```

※画面から確認操作を行っておいてください。

続いて、Panel Server から、U01.F01.M0001 と M0002 に、同時に True を書き込んでみてください。

すると、アラームサマリに M0001 と M0002 のアラームが表示されました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/22 11:08:42		重故障	アラーム003
ALARM-002	2018/05/22 11:08:42		中故障	アラーム002

アラームメールとして以下の内容が送信されます。

```
*****
alarm e-mail.
*****
2018/05/22 11:08:42 中故障
ALARM-002 アラーム 002 発生
--
2018/05/22 11:08:42 重故障
ALARM-003 アラーム 003 発生
--
bottom of e-mail.
```

上記は、アラームサーバーアクションのメール設定で「複数のアラームが発生した場合、一回のメールにまとめて送る」にチェックを入れたため、複数のアラームが1つのメールにまとめて送信されました。本設定のチェックを外すと、上記は別々のメールに分けて送信されます。

※Panel Server から M0001、M0002 を False に戻し、画面から確認操作を行っておいてください。

■メールマスタについて

メールマスタは、メールの送信先情報やフォーマットを定義するテキストファイルです。メールマスタのファイル名にはとくに制限はなく、自由な名前を付けることができます。メールマスタはアラームサーバーアクションで指定したメールマスタ用のフォルダに格納します。このフォルダにはメールマスタ以外のファイルは配置しないでください。

メールマスタは複数のファイルに分けて定義することもできます。各メールマスタの「MailCondition」の定義により、どのような条件のときにメールを送るのかをメールマスタごとに使い分けることができます。例えば、メールマスタ1は中故障以上のメール（[CM2]>=2）、メールマスタ2は重故障のみに限ったメール（[CM2]==3）などのように定義し、それぞれの送信先メールアドレスを別々に定義するといった使い分けが可能となります。

以下に、メールマスタの定義における主な設定を紹介します（※一部抜粋）。

設定項目	内容
[MailCondition]	<p>メール送信条件を指定することができます。送信条件はフィールド名を使用して記述します。例えば[CM2]がアラームレベルをあらわすフィールドだとすると、レベルが2以上のアラームのみ送信したい場合は、以下のように定義します。</p> <pre>[MailCondition] [CM2] >= 2</pre>
[MailSubject]	メールの件名です。
[MailSender]	メールの送信元のメールアドレスを指定します。
[MailSendToName]	<p>メールの送信先のメールアドレスを指定します。複数の送信先を指定する場合、以下のように複数行に分けて記述します。このセクションの終わりの行は、必ず(end)とする必要があります（※CC、BCCについても同様のルールで記述します）。</p> <pre>[MailSendToName] xxx@xxx.co.jp yyy@xxx.co.jp (end)</pre>
[MailHeader]	メール本文のヘッダを指定します。
[MailFooter]	メール本文のフッタを指定します。
[MailMessage]	<p>メールに含めるアラームメッセージのフォーマットを記述します。フォーマットの定義にはアラームのフィールド名を使用することができます。このセクションの終わりの行は、必ず(end)とする必要があります。</p> <pre>[MailMessage] [DATESTR:%Y/%m/%d %H:%M:%S] & " " & [CM1] [ALARMID] & " " & [CM0] & " " & [TYPEJP] "--" (end)</pre> <p>上記の例では、1行目に発生日時の後に警報レベルの文字列（重故障など）を半角スペース区切りで連結し、2行目にアラームID、コメント、アラームの事象を連結し、3行目にハイフン2文字「--」を入れて区切りとしています。</p>

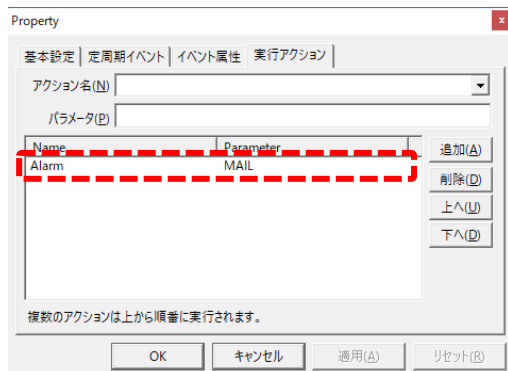


アラームメールのメールマスタに関する詳細は、オンラインマニュアルの「監視システム構築ガイド」
「主要機能の構築」「アラーム」「アラームのカスタマイズ」「アラームメールを送信する」に記述されています。

■メール送信用の定周期イベントについて

アラームメールを送信する場合、アラームサーバーアクションを定周期で呼び出すための通常のイベントとは別に、メール送信処理を呼び出すための定周期イベントを別途作成するようにします。

(メール送信呼び出し用の定周期イベント)



定周期イベントからの実行アクションの定義では、パラメータに「MAIL」を指定するようにします。アラームサーバーアクションは MAIL パラメータ付きで呼び出される事によってメールの送信を行います。

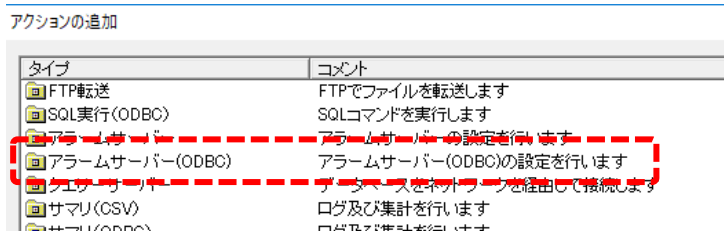
尚、アラームメールの送信メッセージは、一旦、アラームサーバー側でメールキューに蓄えられる仕組みとなっています。アラームサーバーはイベントから MAIL パラメータ付きで呼び出される事によって、キューに存在するメッセージをもとにメールの送信を実行します。



概要

FA-Panel のアラーム機能には、「アラームサーバー (ODBC)」アクションが用意されています。

(Panel Server のアクション追加ダイアログ)



本アクションを使用することで、CSV ファイル形式の保存ではなく、データベース上に用意したテーブルに ODBC 経由でアラームデータを保存する事ができます。

このセクションでは、標準のアラーム機能に対して以下のカスタマイズを行います。

- データベースを用意し、アラーム用のテーブルを作成する。
- ODBC ドライバの設定を行う。
- アラームサーバー (ODBC) アクションを追加し、設定を行う。

尚、画面ライブラリの各種アラーム機能は、アラームサーバーが ODBC であっても、CSV の場合と全く同じように動作します。従って、標準のアラーム機能を CSV 形式からデータベースに変更したいような場合は、アラームサーバーアクションを「アラームサーバー (ODBC)」に変更するのみで、画面側はそのまま利用することができます。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson3-9¥Lesson3-9.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って実際に作業を行ってください。

本セクションでは、アラーム情報を ODBC 経由でデータベースに保存するための設定方法について学習します。弊社製品から接続可能なデータベース環境を用意してください (※各 DB の対応可能なバージョンは、弊社 Web サイトの最新情報を参照してください)。

尚、本テキストでは、以下の環境を用いた構築手順について記載します。

OS： Windows10

DBMS： SQL Server 2017

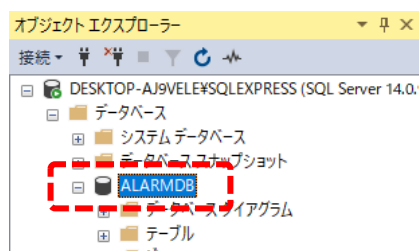
データベースの設定

※ここでは例として、SQL Server 2017 を用いたデータベースの設定例について紹介します。

■アラームデータ用のデータベースを用意する

まず、アラームデータを格納するためのデータベースを1つ用意してください。データベースの名称は自由に付与することができますが、今回は例として、「ALARMDB」という名称のデータベースが作成されているものとします。

以下は、SQLServer Management Studio のオブジェクトエクスプローラーで表示した例です。



尚、アラームサーバー（ODBC）アクションは、必ずしもアラームデータ用に専用のデータベースを用意する必要はありません。接続先のデータベースは、アラームデータを格納するテーブル以外の他のテーブルが混在する環境であっても特に問題はありません。

■データベースにアラームデータ用のテーブルを作成する

続いて、上記のデータベースに、アラームデータを格納するためのテーブルを用意してください。

テーブル名は自由に付与することができますが、今回は例として、ALARMDB データベースに「ALARMTABLE」という名称のテーブルが作成されているものとします。

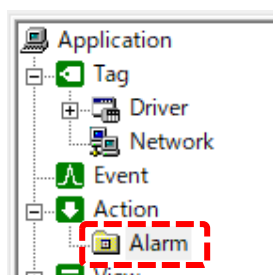
ALARMTABLE テーブルを作成したら、テーブルに以下のフィールドを作成してください。

フィールド名	データ型 (SQLServer)	補足
ALARMDATE	datetime	日付型
ALARMTYPE	varchar(10)	テキスト型
ALARMID	varchar(20)	テキスト型（※フィールドサイズについてはアラームマスタの定義内容により適切なサイズとすること）。
ALARMACK	smallint	数値型
ALARMRECOVER	smallint	数値型
ALARMCONDID	smallint	数値型
SECINDEX	smallint	数値型

アラームサーバー（ODBC）アクションの設定

続いて、Panel Server にアラームサーバー（ODBC）アクションの追加を行い、データベースに接続するための設定を行います。ここでは例として、Windows10 環境で SQL Server 2017 のデータベースと接続するための設定について紹介します。

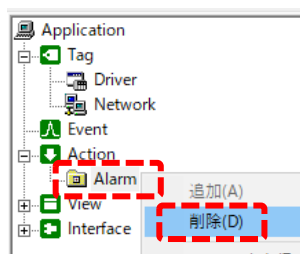
尚、教材サンプルのサーバ設定ファイルでは、標準フレームワークにデフォルトで含まれている「Alarm」という名称のアラームサーバー（CSV）アクションが登録された状態となっており、定周期イベント「Alarm」から当アクションを定周期で呼び出すように設定されています。



そこで、本セクションの作業手順としては、もともと登録されていた「Alarm」アクションを削除して、新たにアラームサーバー（ODBC）アクションを「Alarm」という名称で追加することで、その他のイベントやクライアント側の設定はそのまま変更せずに利用することとします。

1. 「Alarm」アクションを削除する

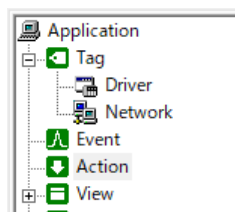
Panel Server の左ツリーから、Action の Alarm を右クリックし、メニューから「削除」を選択してください。



確認画面が表示されるので、「はい」を選択して削除してください。



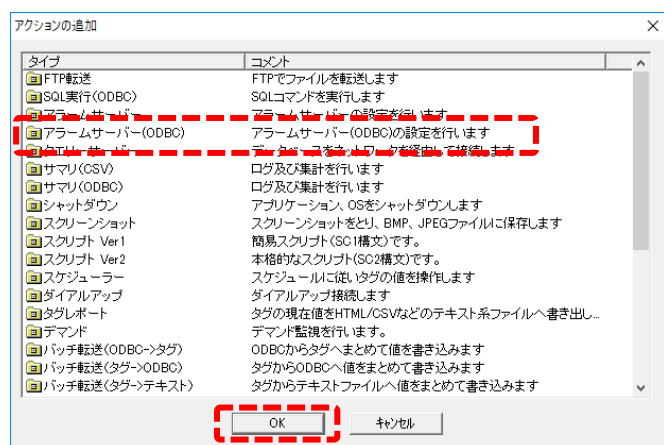
削除されました。



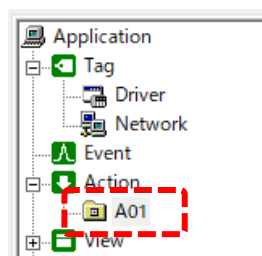
2. アラームサーバー (ODBC) アクションを追加する

Panel Server の左ツリーから Action を右クリックし、追加／アクションを選択してください。

アクションの追加画面が表示されるので、「アラームサーバー (ODBC)」を選択して OK をクリックします。

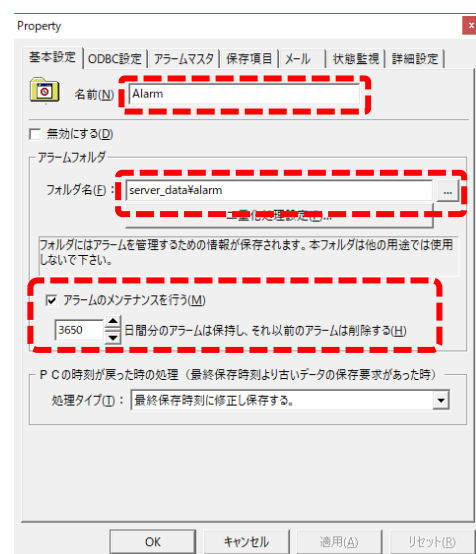


アクション「A01」が追加されました。



3. アラームサーバー (ODBC) アクションの設定を行う

A01 のプロパティを表示し、各種設定を行います。



【基本設定タブ】

名前を「Alarm」に変更します。

アラームフォルダに以下を設定します。

server_data¥alarm

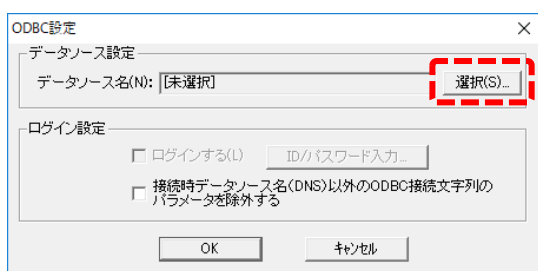
※教材サンプルの場合、server_date フォルダ配下に alarm フォルダが用意されています。

アラームのメンテナンスを行うにチェックを入れ、ここでは 3650 と設定します (※保存日数については、実際の環境にあわせて適切な値を設定してください)。



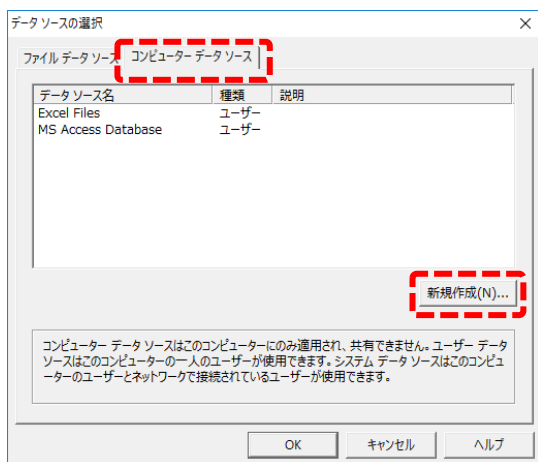
【ODBC 設定タブ】

データソースは、現在未選択となっています。
左記の選択ボタンをクリックしてください。



すると、左記のダイアログが表示されます。

画面から「選択」ボタンをクリックします。

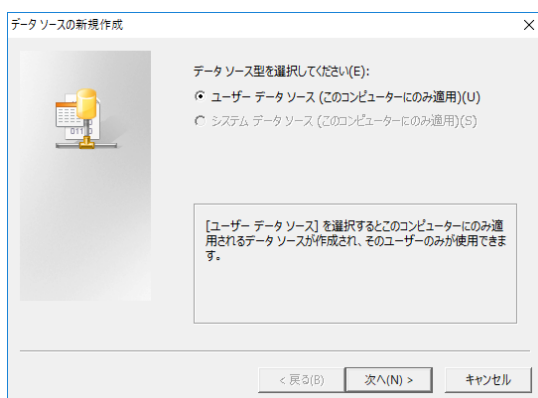


すると、データソースの選択画面が表示されます。

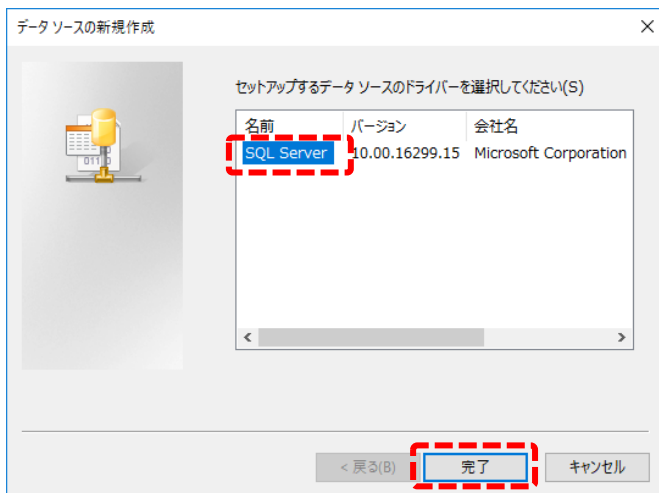
画面から「コンピューターデータソース」のタブを選択し、「新規作成」ボタンをクリックしてください。

※目的の ODBC データソースが既に存在している場合は、リストから対象の DSN を選択して OK をクリックしてください。

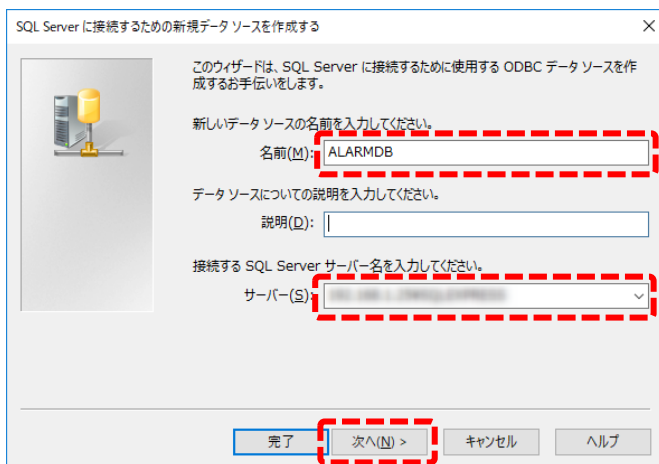
※以降の手順は、実行環境により表示される内容が異なる場合があります。



ユーザーデータソースを選択し、「次へ」ボタンをクリックします。



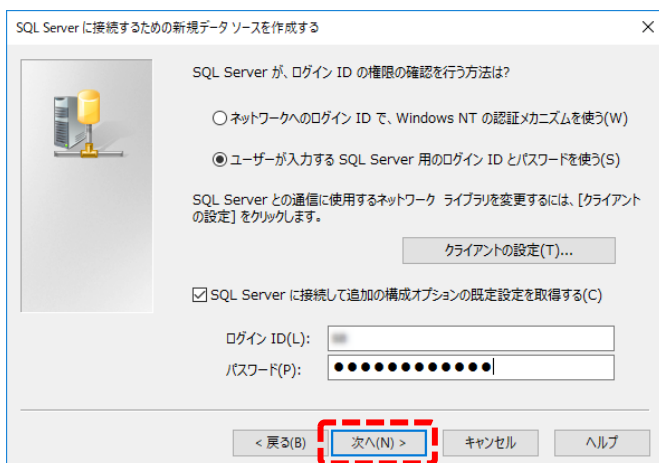
SQL Server を選択し、「完了」ボタンをクリックします。



データソース名に任意の名前を付与します。
ここでは例として「ALARMDB」とします。

また、実際の環境にあわせて SQL Server のサーバー名を指定してください。

設定を入力したら「次へ」をクリックします。



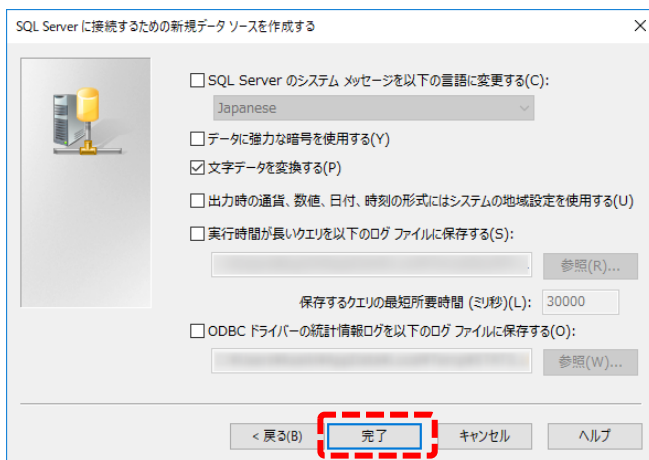
SQL Server にログインするための設定を登録します。実際の環境にあわせて適切な設定を行ってください。

設定を入力したら「次へ」をクリックします。



既定のデータベースに、先ほど用意した「ALARMDB」を指定しておきます。

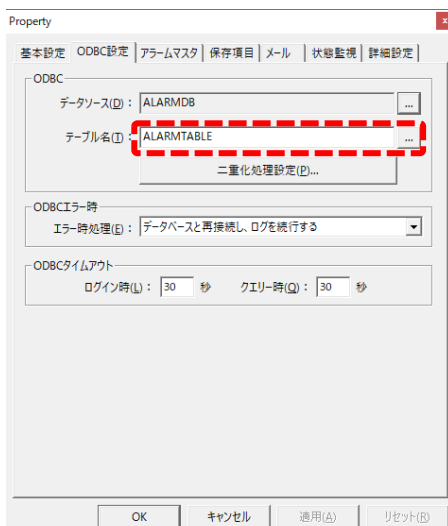
設定を入力したら「次へ」をクリックします。



その他、各オプションについては、実際の環境にあわせて適切な値を設定してください。

設定を入力したら「完了」をクリックします。

上記が完了すると、Panel Server のアラームサーバーアクションの ODBC タブに戻ります。データソースには先ほど追加した「ALARMDB」が表示された状態となります。



続いて、テーブル名に、アラームデータを格納するテーブルを指定します。ここでは例として「ALARMTABLE」と指定しています。

その他の設定はデフォルト値としておきます。

【アラームマスタタブ】

マスターファイルに、以下を指定します。

server¥alarm¥master.csv

※教材サンプルの場合、server¥alarm フォルダの下にアラームマスタが格納されています。

【保存項目タブ】

追加ボタンをクリックして、保存項目を1行追加してください。
追加した行に以下のように設定します。

DB Field ・・ SECINDEX

Alarm ・・ [SECINDEX]

尚、上記で追加した SECINDEX フィールドには、同一秒に発生したアラームに 0 から始まる連番が自動的に付与されて格納されます。
このフィールドの使用は任意ですが、データベースにアラームデータを格納する場合は、SECINDEX フィールドを使用することを強く推奨します。

その他のタブは変更不要です。OK ボタンを押して設定を保存してください。

尚、データベース上のテーブルに作成するフィールド名は、上記の保存項目の定義に合わせる必要があります。具体的には、上記の「DB Field」の列に表示されている名称とフィールド名が一致している必要があります。

以下は、SQL Server の設定例です。

DESKTOP-AJ9VELE...- dbo.ALARMTABLE			
	列名	データ型	NULL を許容
▶	ALARMDATE	datetime	<input checked="" type="checkbox"/>
	ALARMTYPE	varchar(10)	<input checked="" type="checkbox"/>
	ALARMID	varchar(20)	<input checked="" type="checkbox"/>
	ALARMACK	smallint	<input checked="" type="checkbox"/>
	ALARMRECOVER	smallint	<input checked="" type="checkbox"/>
	ALARMCONDID	smallint	<input checked="" type="checkbox"/>
	SECINDEX	smallint	<input checked="" type="checkbox"/>



アラームサーバー（ODBC）アクションの保存項目の設定では、データベース側のテーブルのフィールド名を自由に変更することができます。テーブルのフィールド名を変更したい場合、アラームサーバーアクション側で保存項目の DB Field の列に表示されている名称を変更するとともに、変更した名称とテーブル側のフィールド名を一致させてください。

尚、タグコメントを使用する場合などは、アラームサーバーアクション側の設定で保存項目の追加を行うとともに、データベース側のテーブルにもフィールドを追加するようにします。

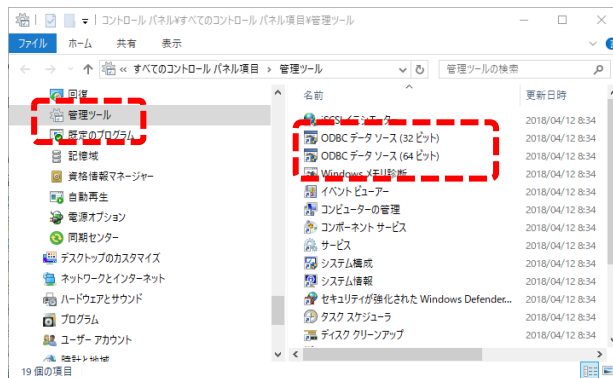
（アラームサーバーアクションの保存項目設定）

基本設定	ODBC設定	アラームマスタ	保存項目	メール
追加(A) 削除(D)				
DB Field	Alarm			
ALARMDATE	[DATE]			
ALARMTYPE	[TYPE]			
ALARMID	[ALARMID]			
ALARMACK	[ACK]			
ALARMRECOVER	[RECOVER]			
ALARMCONDID	[CONDID]			
SECINDEX	[SECINDEX]			



上記の手順で紹介した ODBC 設定では、アラームサーバー（ODBC）アクションの ODBC タブから ODBC のデータソースの追加を行いました。その他の追加方法として、Windows のコントロールパネルの管理ツールから ODBC データソースを追加することもできます。

注意点として、64 ビット環境の Windows のコントロールパネルから ODBC データソースを追加する場合は、使用している FA-Panel のバージョンが 32 ビット版か 64 ビット版かを確認のうえ、同じ実行環境の ODBC データソースを追加するようにしてください。



（Windows10 の例）

動作確認

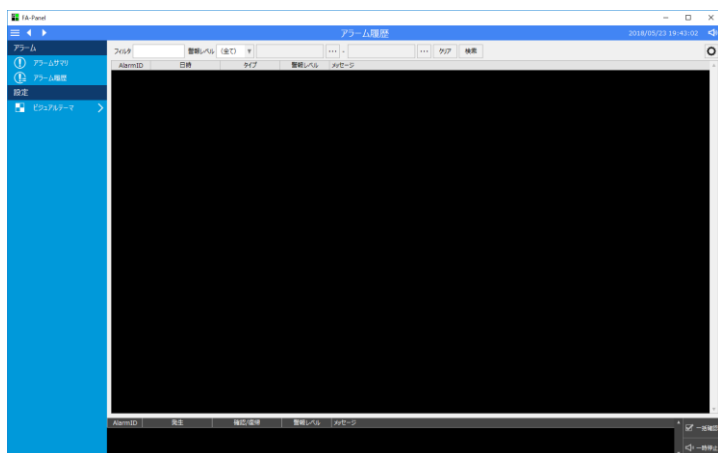
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



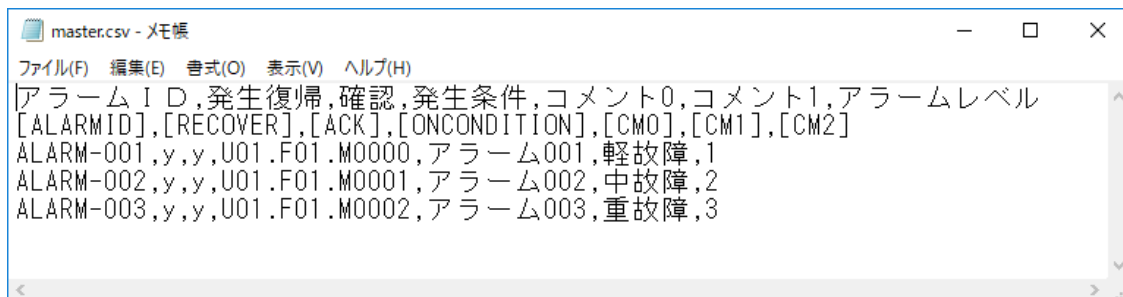
Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動しました。



現状、アラームマスタには、以下のようなアラームが設定されています。



Panel Server から、U01.F01.M0000 に True を書き込んでください。

アラームサマリに M0000 のアラームが表示されました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-001	2018/05/25 09:53:55		軽故障	アラーム001

SQL Server 側のテーブルに、以下のようなレコードが保存されていることを確認してください。

ALARMDATE	ALARMATYPE	ALARMID	ALARMACK	ALARMRECOV...	ALARMCONDID	SECINDEX
2018-05-25 09:53:55.000	new	ALARM-001	0	0	1	0

引き続き、Panel Server から U01.F01.M0001、M0002 に True を書き込んでください。

アラームサマリに M0001、M0002 のアラームが表示されました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-003	2018/05/25 09:56:28		重故障	アラーム003
ALARM-002	2018/05/25 09:56:28		中故障	アラーム002
ALARM-001	2018/05/25 09:53:55		軽故障	アラーム001

SQL Server 側のテーブルには、以下のようなレコードが保存されています。

ALARMDATE	ALARMTYPE	ALARMID	ALARMACK	ALARMRECOV...	ALARMCONDID	SECINDEX
2018-05-25 09:53:55.000	new	ALARM-001	0	0	1	0
2018-05-25 09:56:28.000	new	ALARM-002	0	0	1	0
2018-05-25 09:56:28.000	new	ALARM-003	0	0	1	1

画面から一括確認操作を行い、Panel Server から U01.F01.M0000、M0001、M0002 を False に戻してください。

アラームサマリ上からはアラーム表示が消えました。

AlarmID	発生	確認/復帰	警報レベル	メッセージ

アラーム履歴画面を表示し、アラーム履歴データが正しく表示されることを確認してください。

AlarmID	日時	タイプ	警報レベル	メッセージ
ALARM-003	2018/05/25 09:58:02	復帰	重故障	アラーム003
ALARM-002	2018/05/25 09:58:02	復帰	中故障	アラーム002
ALARM-001	2018/05/25 09:58:02	復帰	軽故障	アラーム001
ALARM-003	2018/05/25 09:57:52	確認	重故障	アラーム003
ALARM-002	2018/05/25 09:57:52	確認	中故障	アラーム002
ALARM-001	2018/05/25 09:57:52	確認	軽故障	アラーム001
ALARM-003	2018/05/25 09:56:28	発生	重故障	アラーム003
ALARM-002	2018/05/25 09:56:28	発生	中故障	アラーム002
ALARM-001	2018/05/25 09:53:55	発生	軽故障	アラーム001

SQL Server 側のテーブルには、以下のようなレコードが保存された状態になりました。

ALARMDATE	ALARMTYPE	ALARMID	ALARMACK	ALARMRECOV...	ALARMCONDID	SECINDEX
2018-05-25 09:53:55.000	new	ALARM-001	0	0	1	0
2018-05-25 09:56:28.000	new	ALARM-002	0	0	1	0
2018-05-25 09:56:28.000	new	ALARM-003	0	0	1	1
2018-05-25 09:57:52.000	ack	ALARM-001	1	0	1	0
2018-05-25 09:57:52.000	ack	ALARM-002	1	0	1	1
2018-05-25 09:57:52.000	ack	ALARM-003	1	0	1	2
2018-05-25 09:58:02.000	recover	ALARM-001	1	1	1	0
2018-05-25 09:58:02.000	recover	ALARM-002	1	1	1	1
2018-05-25 09:58:02.000	recover	ALARM-003	1	1	1	2

Lesson 4 アラームの詳細手順

4-1 アラーム機能の手動組み込み



アラームの各画面機能は、「画面ライブラリ」として提供されています。

画面ライブラリの機能を利用する方法としては、標準フレームワークによる新規プロジェクト作成時のウィザード機能によってあらかじめプロジェクトに組み込んでおく方法と、Panel Editor のライブラリタブから手動で組み込む方法があります。

このセクションでは、画面ライブラリのうち、「アラームサマリ」と「アラーム履歴」を手動で組み込む方法について説明します。標準フレームワークのプロジェクトにアラーム機能を後から追加したい場合や、独自のプロジェクトを構築してアラームを組み込む場合には、本手順を参考にしてください。

レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson4-1¥Lesson4-1.prj

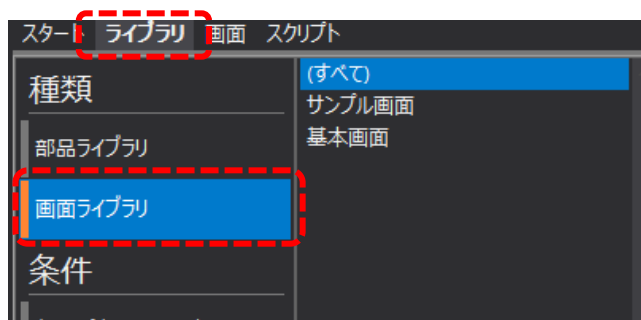
上記のサンプルプロジェクトに対して、以下に示す手順に沿って画面ライブラリの「アラームサマリ」と「アラーム履歴」を実際に組み込んでみましょう。

アラーム画面ライブラリの組み込み

画面ライブラリのうち、「アラームサマリ」「アラーム履歴」を手動で組み込んでみましょう。

1. ライブラリタブから画面ライブラリを選択する

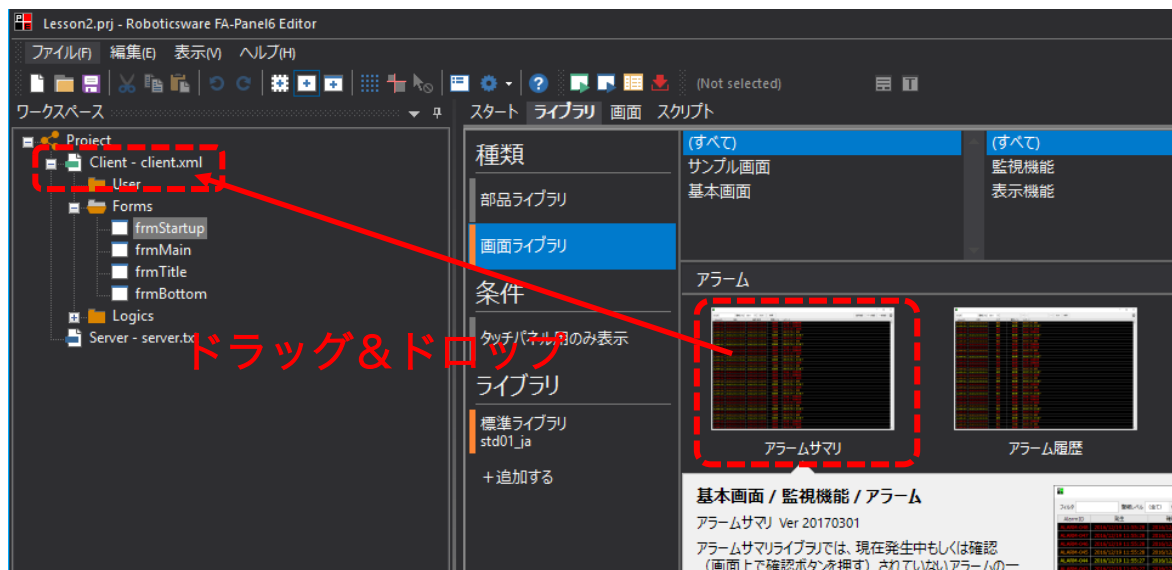
Panel Editor でライブラリタブを選択し、種類を「画面ライブラリ」にしてください。



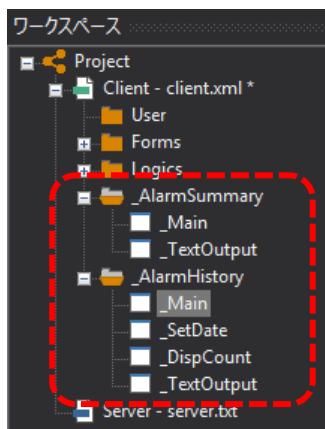
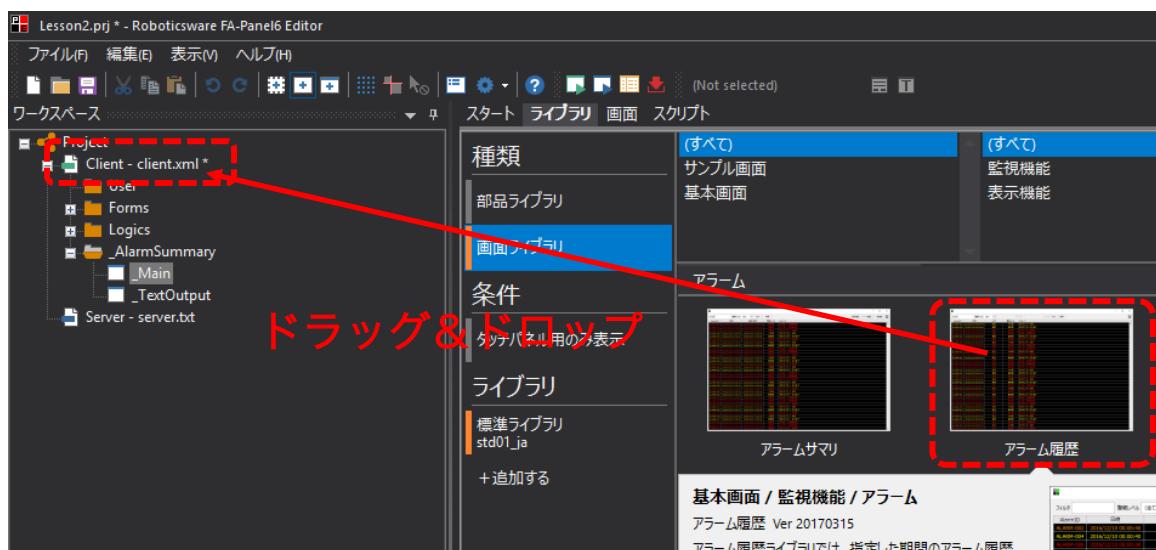
すると、画面ライブラリの選択画面に切り替わります。

2. 画面ライブラリから「アラームサマリ」と「アラーム履歴」を取り込む

画面ライブラリのリストの中から「アラームサマリ」を選択し、プロジェクトのツリーの「client.xml」にドラッグ&ドロップしてください。



同様の手順により、画面ライブラリのリストの中から「アラーム履歴」を選択し、プロジェクトのツリーの「client.xml」にドラッグ&ドロップします。



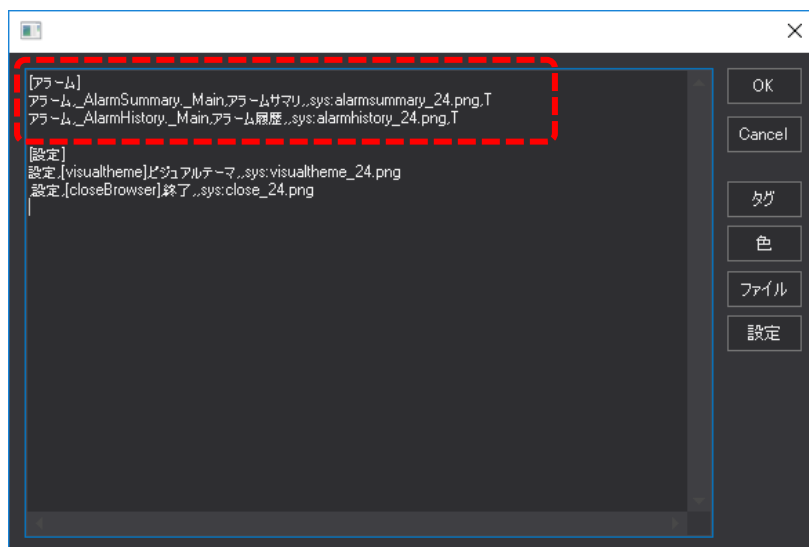
すると、左記のように、「_AlarmSummary」と「_AlarmHistory」というフォルダが作成され、画面ライブラリとして必要なフォーム一式がコピーされました。

_AlarmSummary . . . アラームサマリー式

_AlarmHistory . . . アラーム履歴式

3. 画面呼び出しの記述を追加する

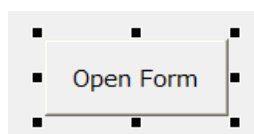
Client.xml（ルート）の「_MenuConf」プロパティに、アラームサマリとアラーム履歴の画面ライブラリを呼び出すための処理を記述してください。



以下は記述例です。

```
[アラーム]
アラーム,_AlarmSummary._Main,アラームサマリ,,sys:alarmsummary_24.png,T
アラーム,_AlarmHistory._Main,アラーム履歴,,sys:alarmhistory_24.png,T
```

尚、上記は標準フレームワークのメニューから呼び出す場合の例です。その他の方法による画面呼び出しを行いたい場合、例えば、左メニュー以外のボタンから画面を呼び出したい場合や、標準フレームワークを利用しない独自プロジェクトで画面の呼び出しを行いたい場合は、部品ライブラリの OpenForm 部品などを使用して画面の呼び出し処理を構築してください。



(OpenForm 部品)



部品ライブラリの OpenForm 部品などから画面ライブラリの画面呼び出しを行う場合、呼び出し先の対象フォーム名に「フォルダ名.フォーム名」（_AlarmSummary._Main、など）で指定するようにします。尚、画面ライブラリとして提供される各画面は、いずれも「_Main」というフォームが呼び出し対象の親フォームとなっています。

サーバ設定の取り込み (Alarm)

続いて、画面ライブラリのサンプルからサーバ側の設定を手動で取り込みます。

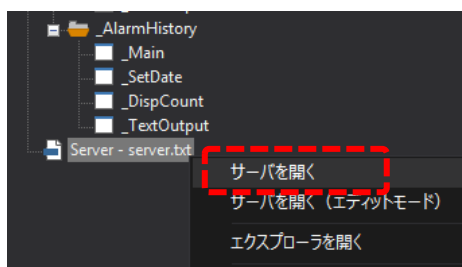
尚、標準フレームワークを利用して生成されたプロジェクトの場合、通常はアラームサーバの設定があらかじめ組み込まれた状態となっており、各設定を手動で取り込む必要はありません。

ここで紹介する手順は、以下のような場合に行ってください。

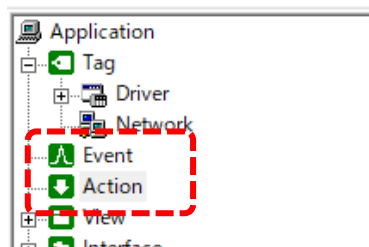
- 標準フレームワークを使用しない独自プロジェクトの場合。
- 標準フレームワークを使用しているが、アラームサーバの設定が削除されている場合（※例えば、導入後にアラーム機能を後から追加する場合など）。

1. Panel Server を開く

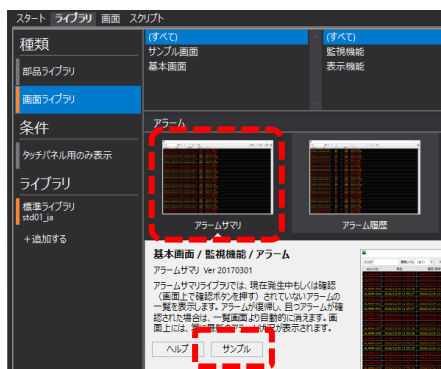
現在作業中のプロジェクトが開かれた状態の Panel Editor のツリーから、「server.txt」を右クリックしてメニューを表示し、「サーバを開く」で Panel Server を呼び出してください。



Panel Server が起動したら、Panel Server のツリーから Action と Event を見てみます。すると、まだ何も登録されていない状態である事がわかります。



2. アラームサマリのサンプルを開く



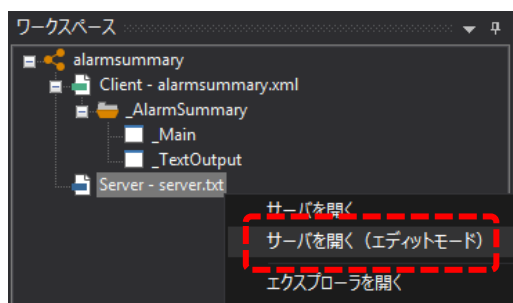
Panel Editor のライブラリタブを選択し、画面ライブラリから「アラームサマリ」のサムネイルを選択してください。

するとサムネイルの下部に機能説明が表示されるので、あわせて表示されている「サンプル」ボタンをクリックしてください。

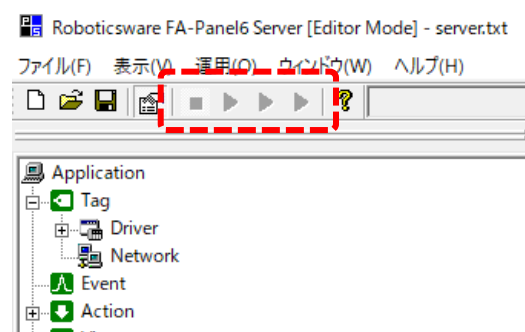
すると、現在作業中の Panel Editor とは別にもう 1 つの Panel Editor が新たに起動し、サンプルプロジェクトが開かれた状態となります。

3. サンプルプロジェクトのサーバ設定ファイルをエディットモードで開く

新たに開かれた Panel Editor の左ツリーから「server.txt」を右クリックしてメニューを表示し、「サーバを開く（エディットモード）」を選択します。



すると新たに、サンプルのサーバ設定ファイルが開かれた状態でエディットモードの Panel Server がもう 1 つ起動します。



エディットモードの Panel Server では、オンライン操作のための各種アイコンは左記のように選択不可の状態であり、オンライン実行することができません。

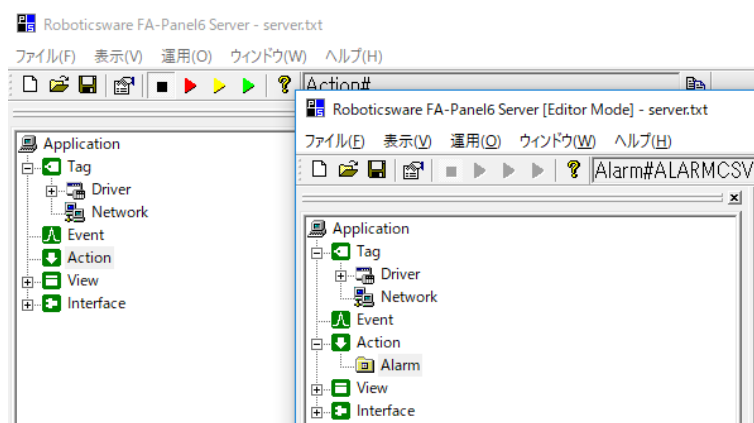
このモードでサンプルのサーバ設定ファイルを開くことにより、アクションやイベントなどの設定を Panel Server 間でコピーすることが可能となります。



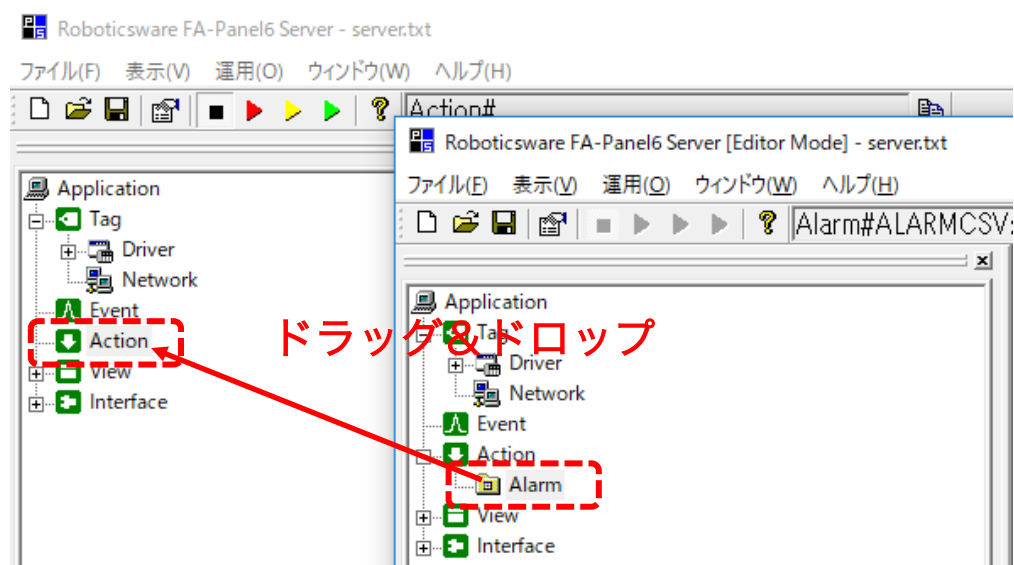
Panel Server は通常モードでは同時に複数起動することはできません。エディットモードで開く場合のみもう 1 つ起動することができます。

4. サンプルから「Alarm」アクションを取り込む

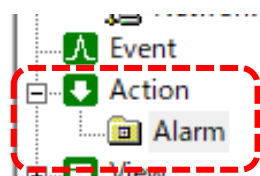
まず、以下のようなイメージで、現在作業中の Panel Server と、サンプルをエディットモードで開いた Panel Server のそれぞれのウィンドウが、両方とも見える状態にします。



この状態で、サンプル側の Panel Server の Action の下に登録されている「Alarm」アクションを、現在作業中の Panel Server の「Action」の上に、ドラッグ&ドロップします。



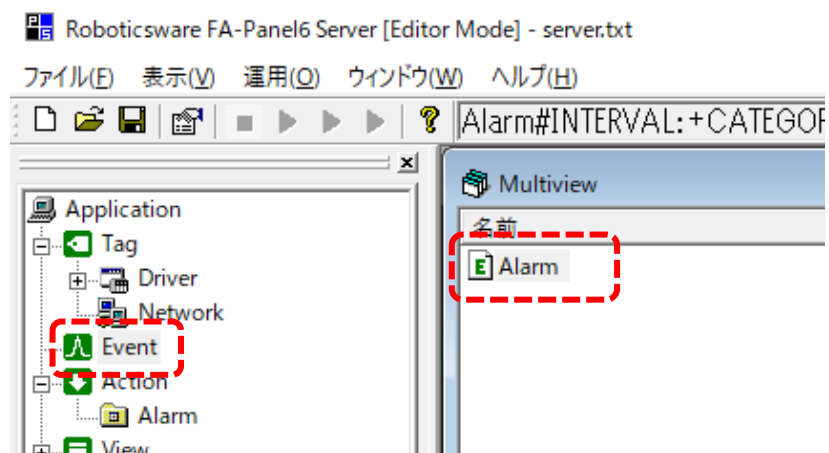
すると、現在作業中の Panel Server 側の Action の下に、Alarm がコピーされました。



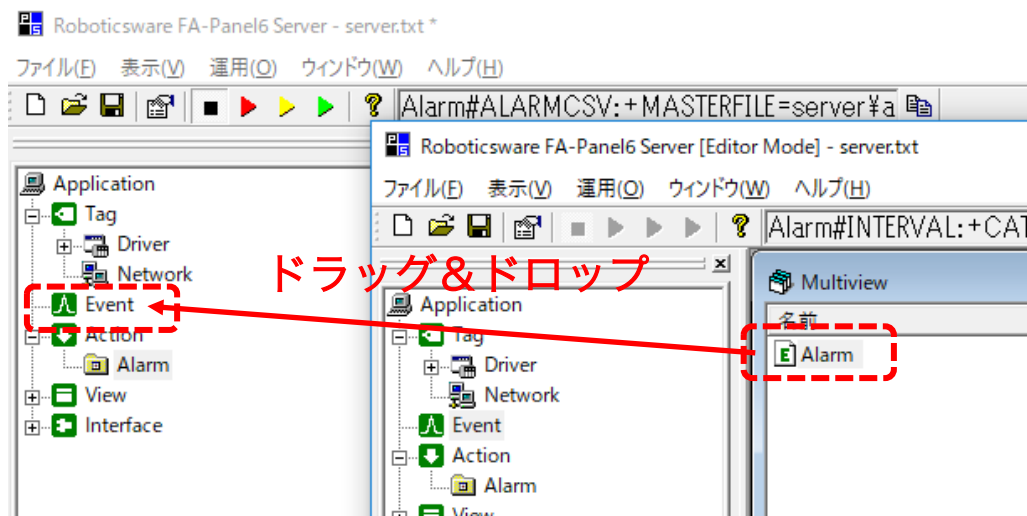
5. サンプルから「Alarm」イベントを取り込む

続いて、必要となるイベントを取り込みます。

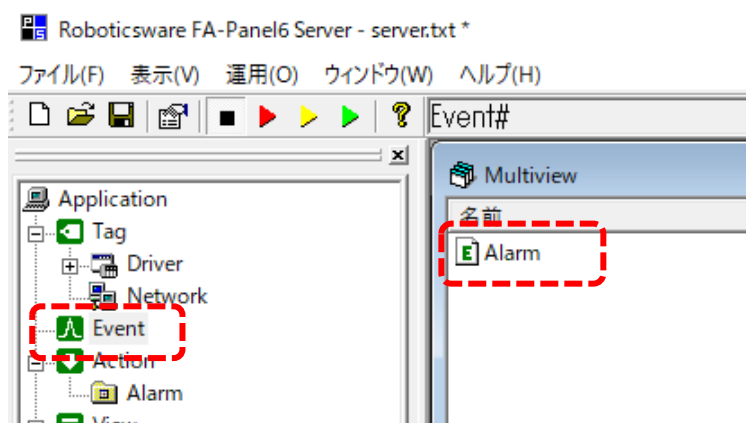
まず、サンプル側の Panel Server 側で「Event」をクリックして、画面中央のマルチビューに「Alarm」イベントが表示された状態にします。尚、このイベントは、Alarm アクションを定期的に呼び出す定周期イベントです。



アクションを取り込んだ手順と同様に、サンプル側の Panel Server の Event に登録されている「Alarm」イベントをマルチビュー上でクリックし、現在作業中の Panel Server の「Event」の上に、ドラッグ&ドロップしてください。

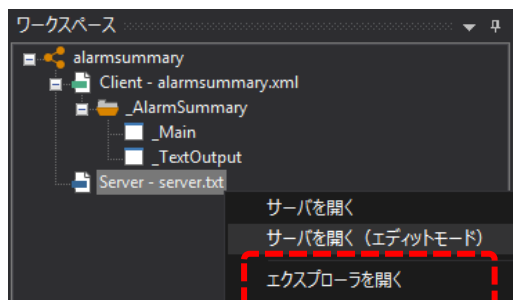


すると、現在作業中の Panel Server のイベントに「Alarm」イベントがコピーされました。

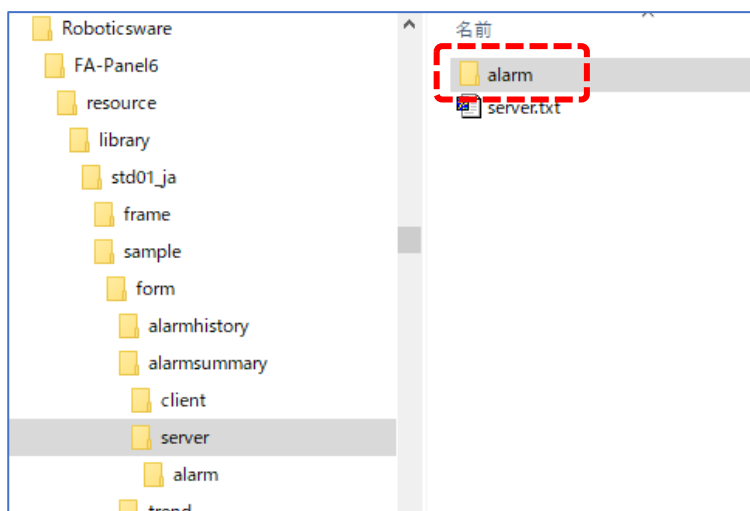


6. サンプルから必要なフォルダを取り込む

続いて、サンプルプロジェクトが開かれた Panel Editor の左ツリーから「server.txt」を右クリックしてメニューを表示し、「エクスプローラを開く」を選択します。



すると、サンプルのサーバ設定ファイルが格納されている場所が開かれた状態でエクスプローラが起動します。



ここで、取り込みが必要なフォルダは、サンプルプロジェクトの server フォルダの下にある「alarm」フォルダです。
現在作業中のプロジェクトフォルダの server フォルダの下に、「alarm」フォルダをフォルダごとコピーして取り込んでください。

例)

¥¥fc01¥Lesson4-1¥server¥alarm

以上で、必要な設定の取り込みは完了です。

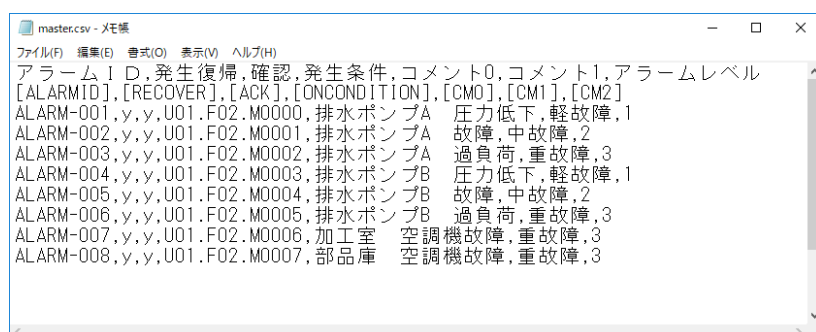
取り込みが完了したら、サンプルプロジェクトが開かれた Panel Editor 及び、エディットモードで開かれた Panel Server を閉じておいてください。

アラームマスタの編集 (Alarm)

「Alarm」アクション用のアラームマスタは、以下のフォルダに格納されています。

¥¥fc01¥Lesson4-1¥server¥alarm¥master.csv

上記のファイルをメモ帳で開いて見ると、以下のような内容が記述されています。



上記はサンプルデータです。実運用の際には不要な行を削除するとともに、アラームマスタの書式のルールに従ってアラームを定義する必要があります。

ここでは例として、以下のように編集してみてください。

- ・ 1 行目、2 行目を残し、3 行目以降を削除する。
- ・ 以下を参考に、3 行目以降を記述する。

```
アラーム I D,発生復帰,確認,発生条件,コメント 0,コメント 1,アラームレベル  
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]  
ALARM-001,y,y,U01.F01.M0000,テストアラーム 1,軽故障,1  
ALARM-002,y,y,U01.F01.M0001,テストアラーム 2,中故障,2  
ALARM-003,y,y,U01.F01.M0002,テストアラーム 3,中故障,2  
ALARM-004,y,y,U01.F01.M0003,テストアラーム 4,重故障,3
```

尚、教材サンプルには、仮想デバイスとして「U01」があらかじめ登録されています。今回の例では、U01.F01.M0000 から M0003 までのタグをアラームとして使用します。

編集が終わったら、アラームマスタを上書き保存してください。

動作確認

それでは、アラーム機能の動作確認を行ってみましょう。

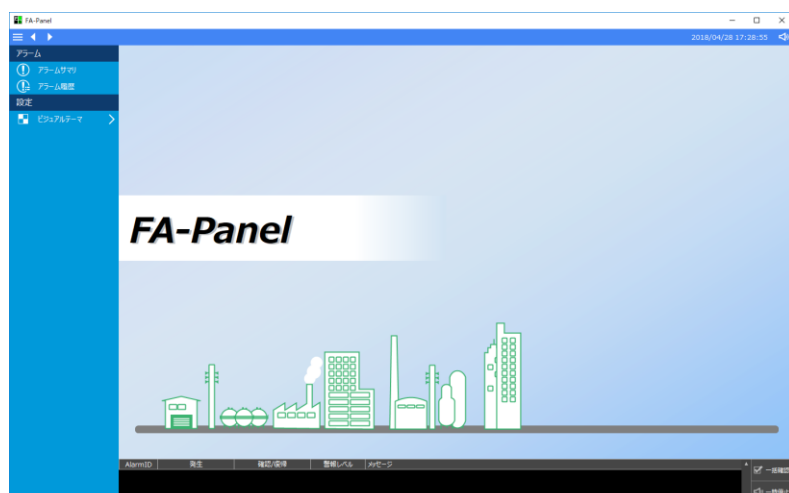
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。

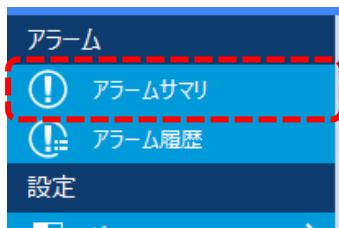


Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動しました。

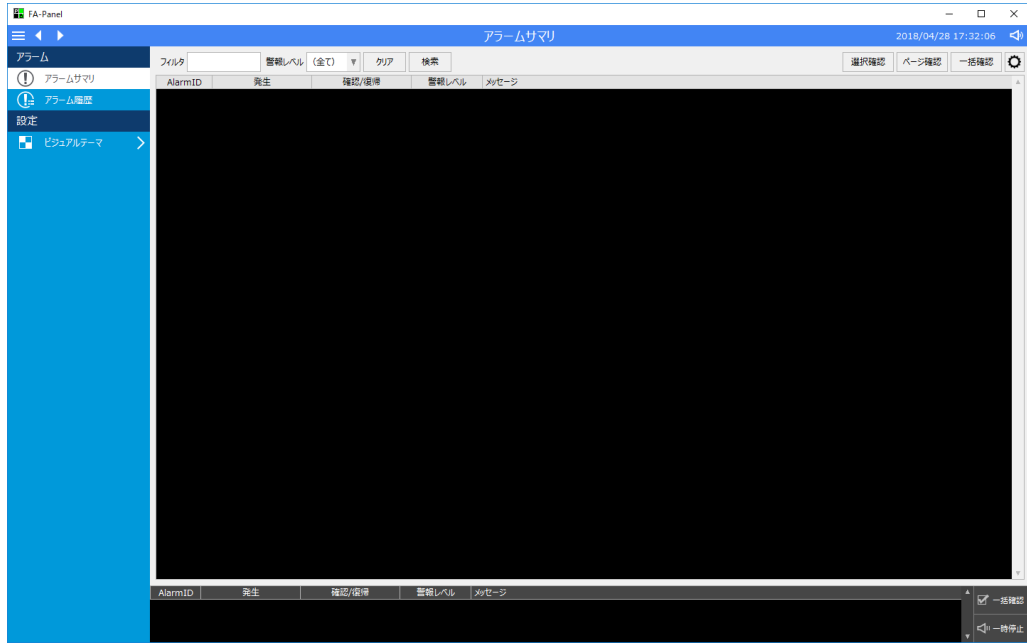




画面左のメニューを見ると、先ほど追加した「アラームサマリ」および「アラーム履歴」が追加されています。

メニューから「アラームサマリ」をクリックします。

アラームサマリ画面が表示されました。



Panel Server から、U01.F01.M0000 から M0003 のタグを操作し、アラームが正しく表示されることを確認してください。

(参考) イベント機能の手動組み込み手順

画面ライブラリには、イベント監視のための機能として、「イベント履歴」と「イベントリアルタイム」が用意されています。ここで言うイベントとは、設備の運転履歴などの記録のことを指します。又、これらの機能はアラーム機能と同様に、アラームサーバーアクションを利用して構築されています。

尚、イベント機能の組み込み方法は、アラーム機能の組み込みと同等の考え方で行うことができます。

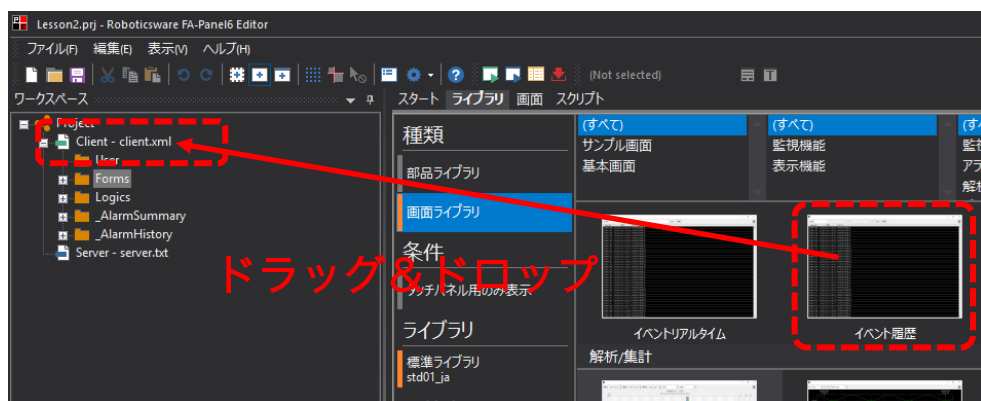
本セクションでは、前セクションで編集した「Lesson4-1.prj」に対して、イベント機能を手動で組み込むための手順をダイジェストとして紹介します。

■イベント画面ライブラリの組み込み

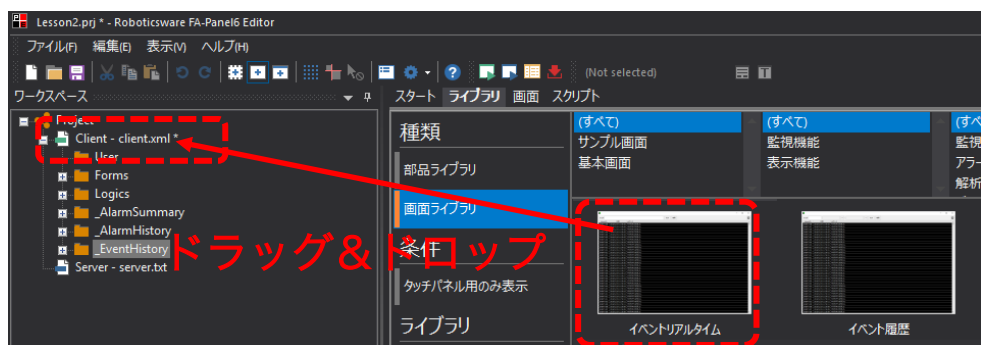
1. 画面ライブラリから「イベント履歴」と「イベントリアルタイム」を取り込む

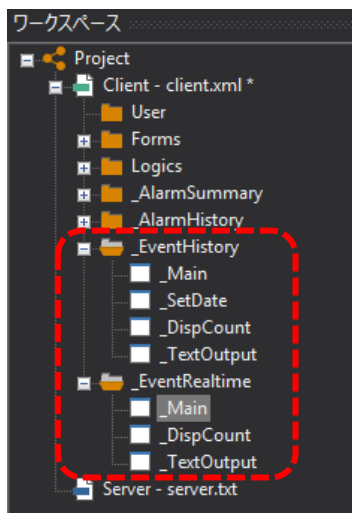
Panel Editor でライブラリタブを選択し、種類を「画面ライブラリ」にします。

画面ライブラリのリストの中から「イベント履歴」を選択し、プロジェクトのツリーの「client.xml」にドラッグ&ドロップします。



同様の手順により、画面ライブラリのリストの中から「イベントリアルタイム」を選択し、プロジェクトのツリーの「client.xml」にドラッグ&ドロップします。



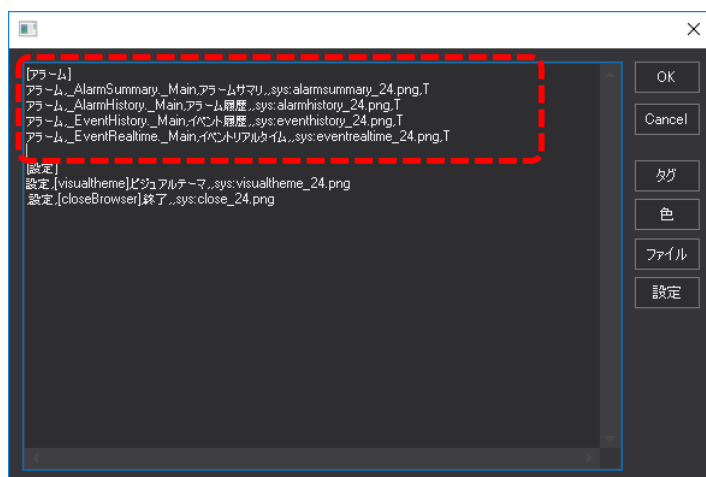


すると、左記のように、「_EventHistory」と「_EventRealtime」というフォルダが作成され、画面ライブラリとして必要なフォーム一式がコピーされました。

_EventHistory . . . イベント履歴一式
_EventRealtime . . . イベントリアルタイム一式

2. 画面呼び出しの記述を追加する

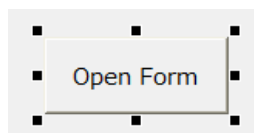
Client.xml (ルート) の「_MenuConf」プロパティに、イベント履歴とイベントリアルタイムの画面ライブラリを呼び出すための処理を記述します。



以下は記述例です。

```
[アラーム]
アラーム._AlarmSummary._Main,アラームサマリ,sys:alarmsummary_24.png,T
アラーム._AlarmHistory._Main,アラーム履歴,sys:alarmhistory_24.png,T
アラーム._EventHistory._Main,イベント履歴,sys:eventhistory_24.png,T
アラーム._EventRealtime._Main,イベントリアルタイム,sys:eventrealtime_24.png,T
```

尚、標準フレームワークのメニュー以外の方法による画面呼び出しを行いたい場合には、部品ライブラリの OpenForm 部品などを使用して画面の呼び出し処理を構築します。



(OpenForm 部品)

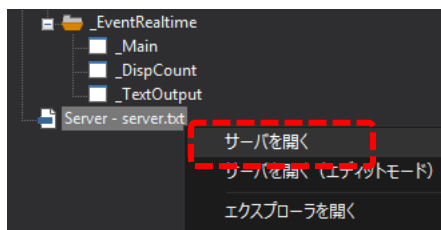
■サーバ設定の取り込み (Event)

アラームと同様の方法で、画面ライブラリのサンプルからサーバ側の設定を手動で取り込みます。

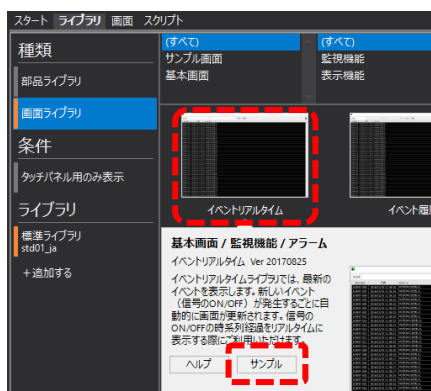
尚、アラーム監視と同様に、標準フレームワークを利用して生成されたプロジェクトの場合、通常はイベント用のアラームサーバの設定があらかじめ組み込まれた状態となっており、各設定を手動で取り込む必要はありません。

1. Panel Server を開く

現在作業中のプロジェクトが開かれた状態の Panel Editor のツリーから、「server.txt」を右クリックしてメニューを表示し、「サーバを開く」で Panel Server を呼び出してください。



2. イベントリアルタイムのサンプルを開く



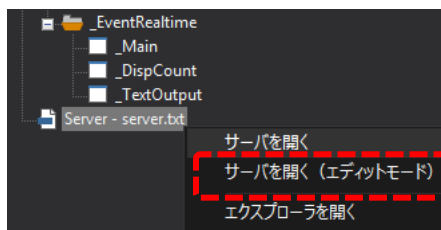
Panel Editor のライブラリタブを選択し、画面ライブラリから「イベントリアルタイム」のサムネイルを選択します。

するとサムネイルの下部に機能説明が表示されるので、あわせて表示されている「サンプル」ボタンをクリックします。

すると、現在作業中の Panel Editor とは別にもう 1 つの Panel Editor が新たに起動し、サンプルプロジェクトが開かれた状態となります。

3. サンプルプロジェクトのサーバ設定ファイルをエディットモードで開く

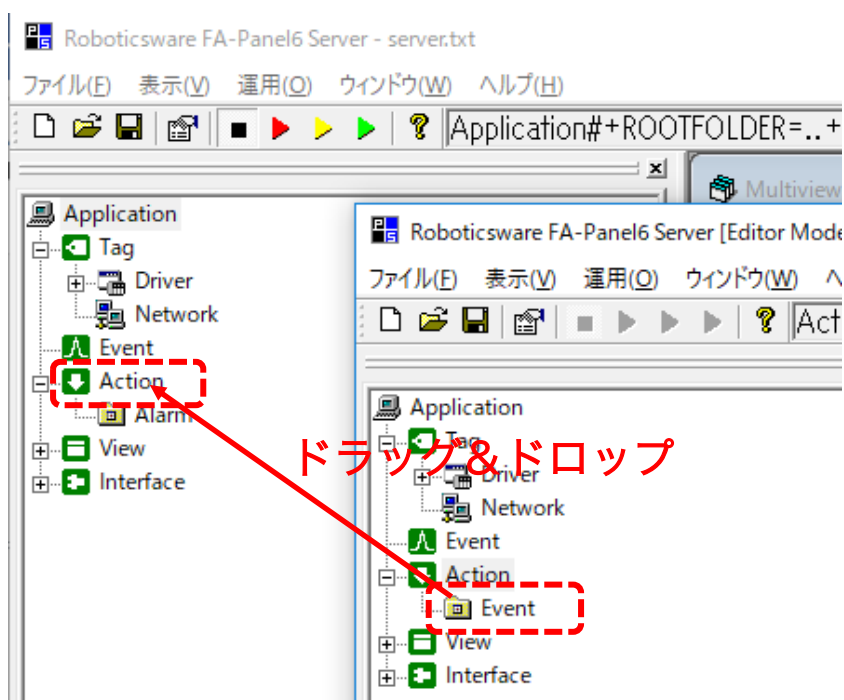
新たに開かれた Panel Editor の左ツリーから「server.txt」を右クリックしてメニューを表示し、「サーバを開く (エディットモード)」を選択します。



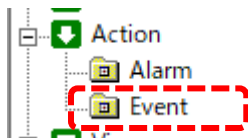
すると新たに、サンプルのサーバ設定ファイルが開かれた状態でエディットモードの Panel Server がもう 1 つ起動します。

4. サンプルから「Event」アクションを取り込む

サンプル側の Panel Server の Action の下に登録されている「Event」アクションを、現在作業中の Panel Server の「Action」の上に、ドラッグ&ドロップします。

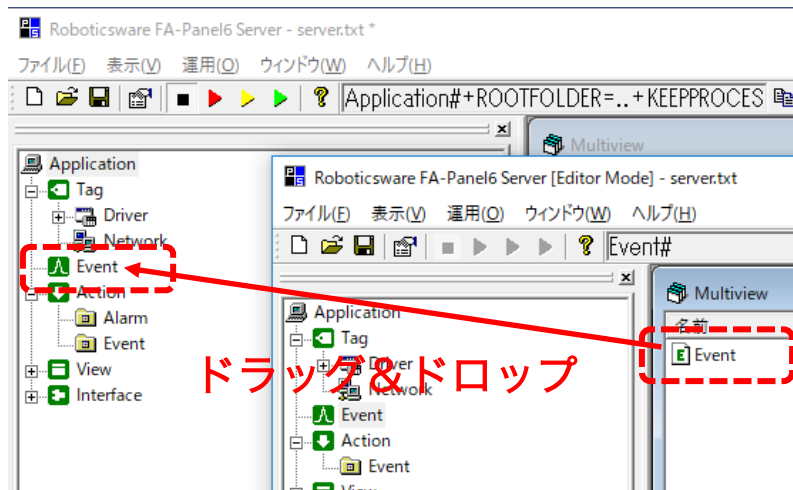


現在作業中の Panel Server 側の Action の下に、「Event」アクションがコピーされました。

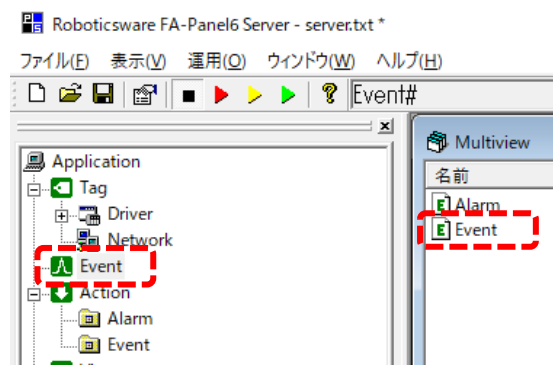


5. サンプルから「Event」イベントを取り込む

続いて、サンプル側の Panel Server の Event に登録されている「Event」イベントをマルチビュー上でクリックし、現在作業中の Panel Server の「Event」の上に、ドラッグ&ドロップします。

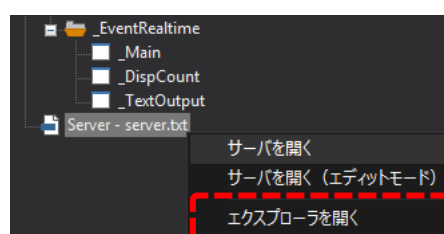


現在作業中の Panel Server のイベントに「Event」 イベントがコピーされました。

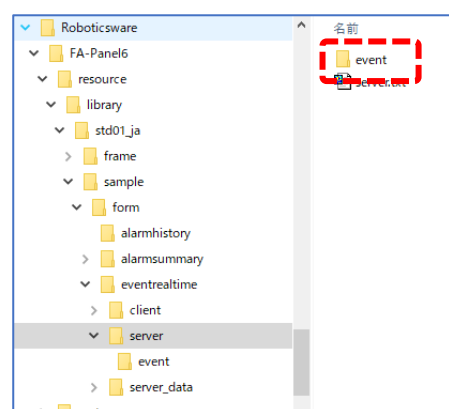


6. サンプルから必要なフォルダを取り込む

続いて、サンプルプロジェクトが開かれた Panel Editor の左ツリーから「server.txt」を右クリックしてメニューを表示し、「エクスプローラを開く」を選択します。



すると、サンプルのサーバ設定ファイルが格納されている場所が開かれた状態でエクスプローラが起動します。



ここで、取り込みが必要なフォルダは、サンプルプロジェクトの server フォルダの下にある「event」フォルダです。
現在作業中のプロジェクトフォルダの server フォルダの下に、「event」フォルダをフォルダごとコピーして取り込んでください。

例)

¥¥fc01¥Lesson4-1¥server¥event

以上で、必要な設定の取り込みは完了です。

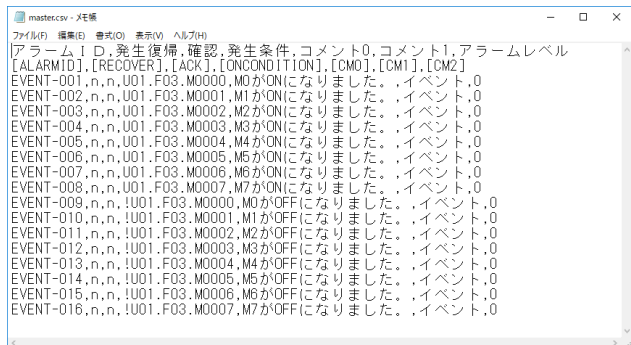
■アラームマスタ (Event) の編集

イベントの監視を行う「Event」アクション用のアラームマスタを編集します。「Event」アクション用のアラームマスタは、server フォルダの下に「event」フォルダ内に「master.csv」というファイル名で格納されています。

¥¥fc01¥Lesson4-1¥server¥event¥master.csv

アラームマスタ「master.csv」を、メモ帳などのテキストエディタで開いてみてください。

すると、以下のような内容が記述されています。



上記はサンプルデータです。実運用の際には不要な行を削除するとともに、アラームマスタの書式のルールに従ってイベントの条件を定義する必要があります。

- ・ 1 行目、2 行目を残し、3 行目以降を削除する。
- ・ 以下を参考に、3 行目以降を記述する。

```
アラーム I D,発生復帰,確認,発生条件,コメント 0,コメント 1,アラームレベル  
[ALARMID],[RECOVER],[ACK],[ONCONDITION],[CM0],[CM1],[CM2]  
EVENT-001,n,n,U01.F01.M0010,M0010 が ON になりました,イベント,0  
EVENT-002,n,n,U01.F01.M0010,M0010 が OFF になりました,イベント,0  
EVENT-003,n,n,U01.F01.M0011,M0011 が ON になりました,イベント,0  
EVENT-004,n,n,U01.F01.M0011,M0011 が OFF になりました,イベント,0  
EVENT-005,n,n,U01.F01.M0012,M0012 が ON になりました,イベント,0  
EVENT-006,n,n,U01.F01.M0012,M0012 が OFF になりました,イベント,0
```

尚、上記の例では、あらかじめ教材プロジェクトに登録されている U01 ユニットを使用し、U01.F01. M0010 から M0012 までのタグをイベントとして使用しています。

編集が終わったら、アラームマスタを上書き保存します。

以上で、イベント機能の手動取り込み作業は完了です。

4-2 独自プロジェクトに警報音を実装する



概要

標準フレームワークを利用せずに、独自に構築されたプロジェクトでアラーム機能を利用したい場合、最も簡単な方法は、画面ライブラリのアラーム機能を組み込んで利用する方法です。対象のプロジェクトに対して画面ライブラリの各種アラーム機能を手動で組み込むことにより、アラームサマリやアラーム履歴などの標準機能を簡単に利用することができます。

しかしながら、例えば、警報音を鳴らす機能については、標準フレームワークの枠組みに組み込まれて提供される機能のため、警報音の機能を独自プロジェクトで実現したい場合は、ユーザーによる作り込みが必要となります。

そこで、本セクションでは、標準フレームワークを利用しない独自プロジェクトで警報音を鳴らすための仕組みを構築する方法について学習します。

- 教材サンプルの独自プロジェクトにバックグラウンドフォームを追加する
- バックグラウンドフォームに必要な部品を張り付け、警報音を鳴らすロジックを実装する
- 警報音の停止、有効/無効の切替などの操作ボタンを実装する

(独自プロジェクトの画面イメージ)



レッスンの準備

このレッスンには教材サンプルが用意されています。

Panel Editor で以下のプロジェクトを開いてください。

¥¥fc01¥Lesson4-2¥Lesson4-2.prj

上記のサンプルプロジェクトに対して、以下に示す手順に沿って作業を行ってください。

バックグラウンドフォームを追加する

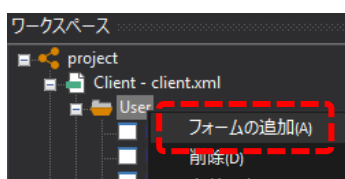
本セクション用の教材サンプルとして用意されているプロジェクトは、標準フレームワークを利用していない独自のプロジェクトです。

このプロジェクトでは、あらかじめアラームサマリとアラーム履歴機能が手動で組み込まれた状態となっています。また、各機能はフォームの左側に配置されているボタンから呼び出すように構築されています。しかし、このプロジェクトにはバックグラウンドフォームが用意されていません。警報音を鳴らすためには、バックグラウンドフォームを利用するのが最適な方法です。

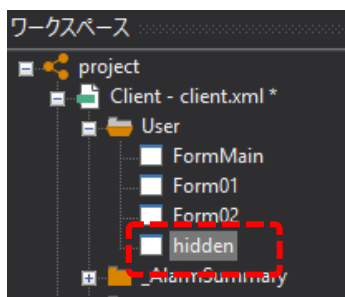
そこで、教材サンプルプロジェクトに対してバックグラウンドフォームの追加を行います。

1. フォームを追加する

Panel Editor のワークスペースのツリーから、User フォルダを右クリックしてフォームの追加を行ってください。



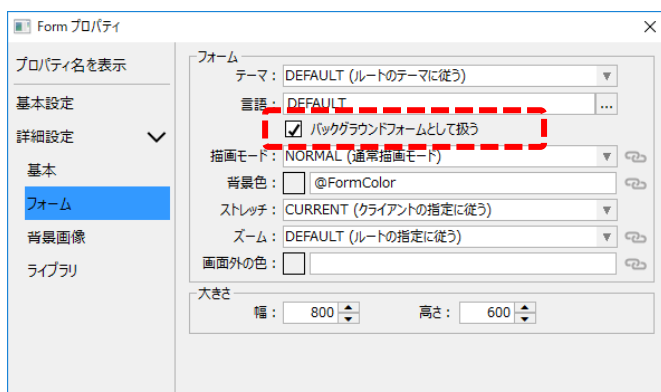
新たに追加されたフォームの名称を「hidden」に変更してください。



尚、バックグラウンドフォームに設定する際、フォームの名称は特に関係ありません。また、フォームの格納場所についても特に制約はありません。ユーザー側で任意の名称を付与しても問題ありません。

2. フォームをバックグラウンドフォームに設定する

上記の「hidden」フォームのプロパティを開き、詳細設定／フォームにある「バックグラウンドフォームとして扱う」(HiddenForm プロパティ) にチェックを入れてください。



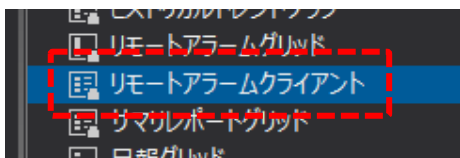
左記のプロパティを有効にすると、このフォームはバックグラウンドフォームとなります。バックグラウンドフォームとして設定されたフォームは、クライアントの実行中、常に裏で開かれている状態となります。

警報音を鳴らすロジックの実装

続いて、先ほど追加したバックグラウンドフォームに、警報音を発生させるために必要となる部品を張り付け、ロジックを実装します。

1. hidden フォームにリモートアラームクライアントを張り付ける

Panel Editor のコントロールから「リモートアラームクライアント」を選択し、hidden フォームに張り付けてください。

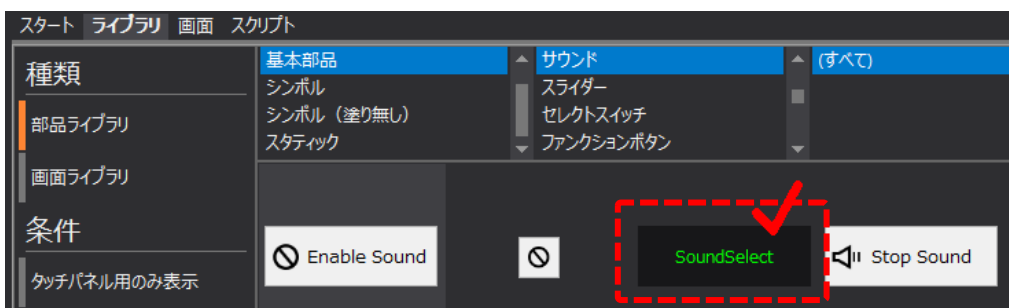


フォームに張り付けられました。

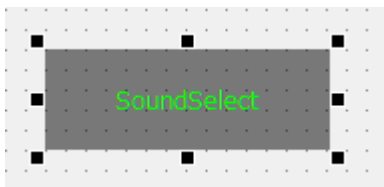


2. hidden フォームにサウンドセレクト部品を張り付ける

ライブラリタブから部品ライブラリを選択して、基本部品／サウンドと絞り込み、サウンドセレクト部品を hidden フォームに張り付けてください。

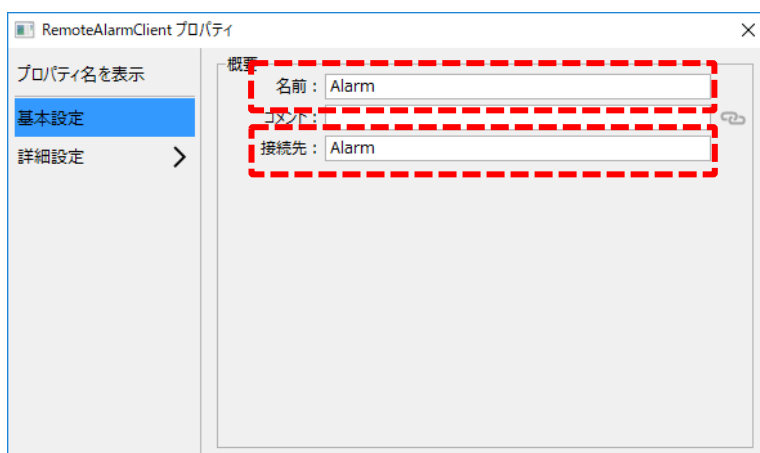


フォームに張り付けられました。



3. リモートアラームクライアントのプロパティ設定

Panel Editor で hidden フォームに張り付けたリモートアラームクライアントを選択し、以下のプロパティを設定してください。



名前 (Name プロパティ) :

Alarm

接続先 (AlarmServer プロパティ) :

Alarm

※接続先には、Panel Server に登録されているアラームサーバ名を指定する。

4. リモートアラームクライアントコントロールにスクリプトを記述する

リモートアラームクライアントを選択した状態で、スクリプトページを開いてください。

OnInitialize イベントを追加し、以下のコードを記述してください。

```
event OnInitialize()
{
    this.Connect();
    this.SubscribeSummary("");
    this.AddTag("$SOUND0");
}
```

※ コードはイベントの「{ 」と「 } 」の間に記述します。

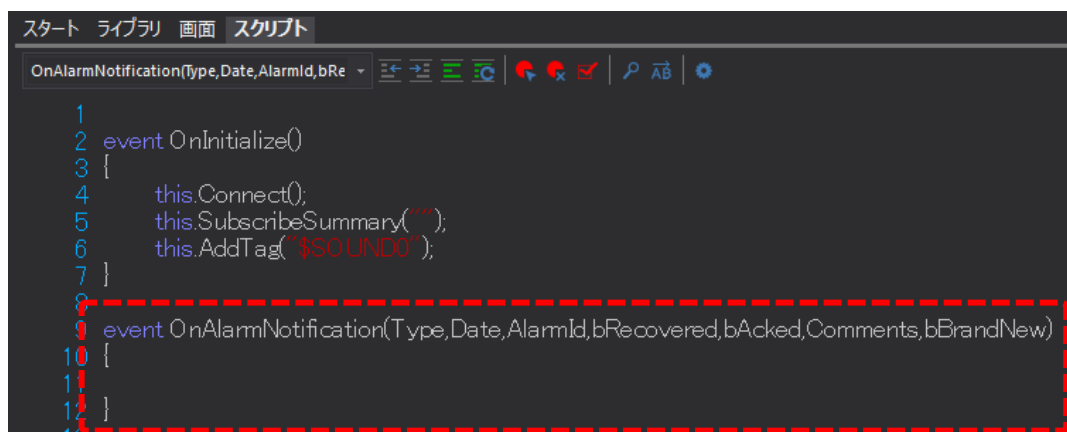
※ すべて半角で記述します。

※ 大文字と小文字を区別してください。尚、\$SOUND0 の1つめの「O」は大文字オー、末尾は半角ゼロです。

(記述例)

```
1
2 event OnInitialize()
3 {
4     this.Connect();
5     this.SubscribeSummary("");
6     this.AddTag("$SOUND0");
7 }
```

続いて、OnAlarmNotification イベントを追加してください。



```
1
2 event OnInitialize()
3 {
4     this.Connect();
5     this.SubscribeSummary("");
6     this.AddTag("$SOUND0");
7 }
8
9 event OnAlarmNotification(Type, Date, AlarmId, bRecovered, bAcked, Comments, bBrandNew)
10 {
11 }
12
13
```

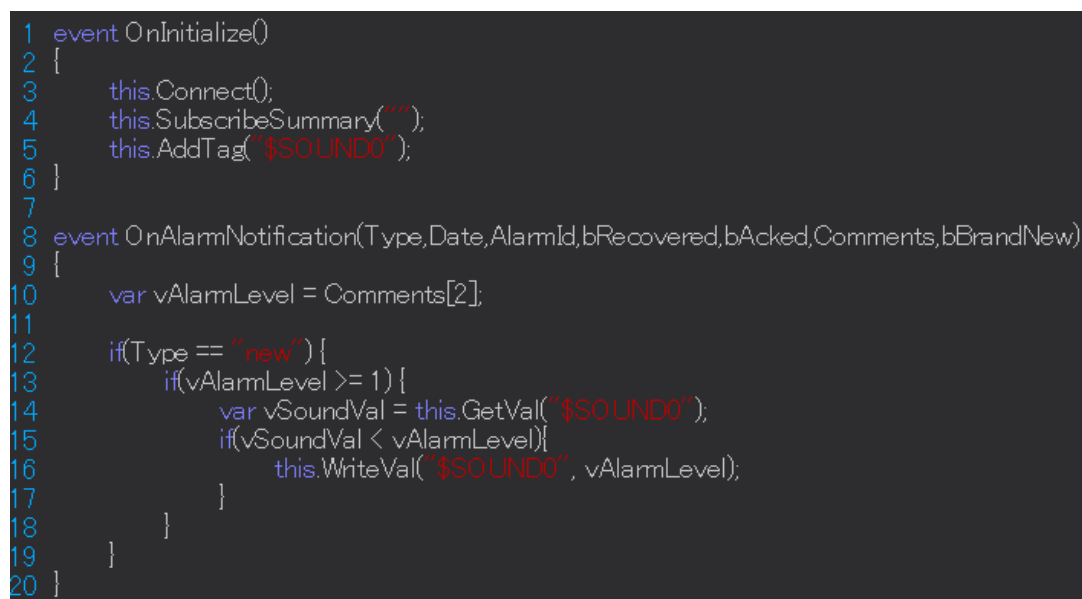
OnAlarmNotification イベントの「{」と「}」の間に、以下のコードを記述してください。

尚、このイベントは、アラームの発生、復帰、確認時など、アラームの状態が変化した際に都度呼び出されます。

```
event OnAlarmNotification(Type, Date, AlarmId, bRecovered, bAcked, Comments, bBrandNew)
{
    var vAlarmLevel = Comments[2];

    if(Type == "new") {
        if(vAlarmLevel >= 1) {
            var vSoundVal = this.GetVal("$SOUND0");
            if(vSoundVal < vAlarmLevel){
                this.WriteVal("$SOUND0", vAlarmLevel);
            }
        }
    }
}
```

(記述例)



```
1 event OnInitialize()
2 {
3     this.Connect();
4     this.SubscribeSummary("");
5     this.AddTag("$SOUND0");
6 }
7
8 event OnAlarmNotification(Type, Date, AlarmId, bRecovered, bAcked, Comments, bBrandNew)
9 {
10     var vAlarmLevel = Comments[2];
11
12     if(Type == "new") {
13         if(vAlarmLevel >= 1) {
14             var vSoundVal = this.GetVal("$SOUND0");
15             if(vSoundVal < vAlarmLevel){
16                 this.WriteVal("$SOUND0", vAlarmLevel);
17             }
18         }
19     }
20 }
```

上記の2つのイベントの処理記述により、警報発生時に警報音の呼び出しが行われるようになります。

警報音操作ボタンの実装

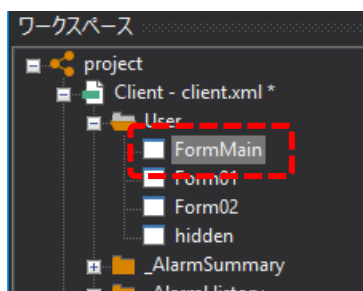
ここまでの手順で実装した処理内容には、アラーム発生時に警報音を鳴らすのみで、警報音を停止する仕組みがありません。そこで、警報音の停止と、警報音の有効／無効を切り替えるための操作ボタンを追加します。

部品ライブラリには、サウンドセレクト部品の操作を行うためのボタン部品が用意されています。今回追加する警報音の操作処理についても、部品ライブラリの部品を利用することとします。

今回は例として、画面の左側のメニュー部分にボタンを追加します。

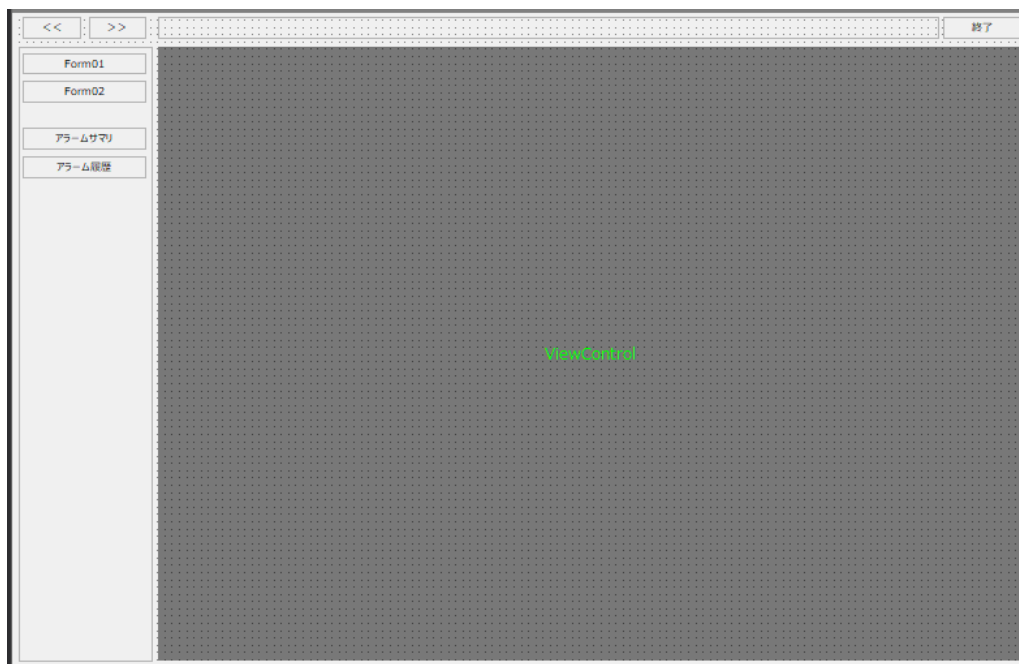
1. メインフォームを選択する

Panel Editor のワークスペースのツリーから、User フォルダの下にある「FormMain」を選択してください。



教材サンプルでは、起動時に FormMain フォームが自動的に呼び出されるように設定されています。また、画面の左側に配置されているボタン群をクリックすると、フォーム中央部分に張り付けられているビューコントロール上に対称のフォームが表示される仕組みとなっています。

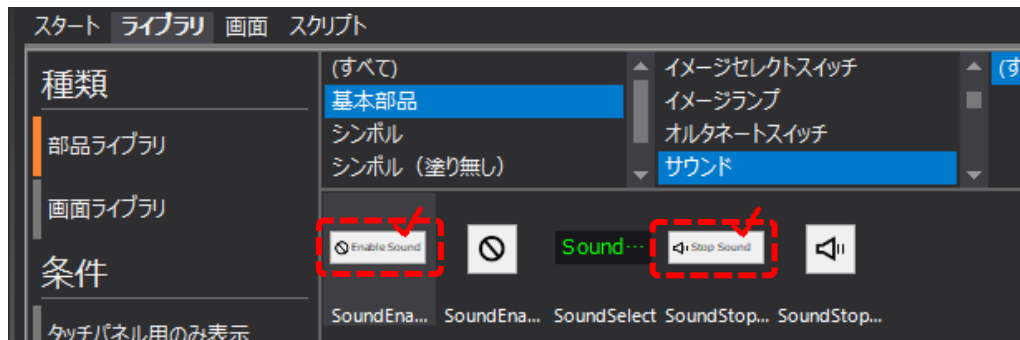
(FormMain フォーム)



このフォームの左側のメニュー部分に、警報音を操作するための部品ライブラリを張り付けます。

2. 部品ライブラリのサウンド操作部品を張り付ける

ライブラリタブから、部品ライブラリの SoundEnable101、SoundStop101 の2つの部品を選択して、FormMain フォームに張り付けてください。



ここでは例として、左下隅の近辺に以下のように配置しました。



尚、左記では、ボタンのサイズ、テキストの表示内容やアイコンの表示位置を調整しています。

部品ライブラリの部品を張り付けた後に、自由に調整してみてください。

(参考) 編集したプロパティ：

_Text
Width
Height
TextCenterX
ImageLeft
ImageRight
ImageTop
ImageBottom

以上で設定作業は完了です。

動作確認

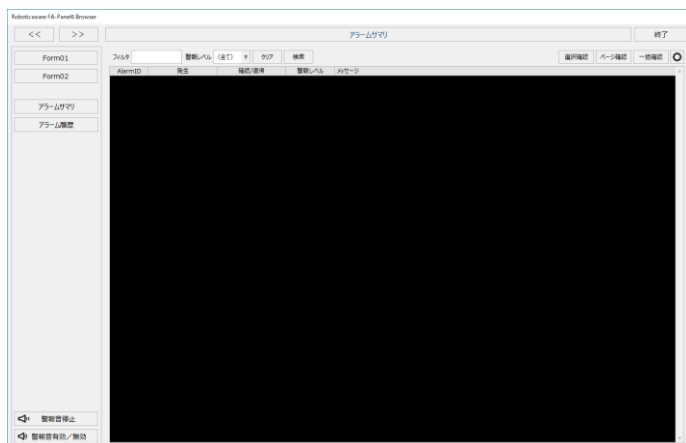
Panel Server の画面上部にあるオンライン（黄矢印）をクリックしてください。



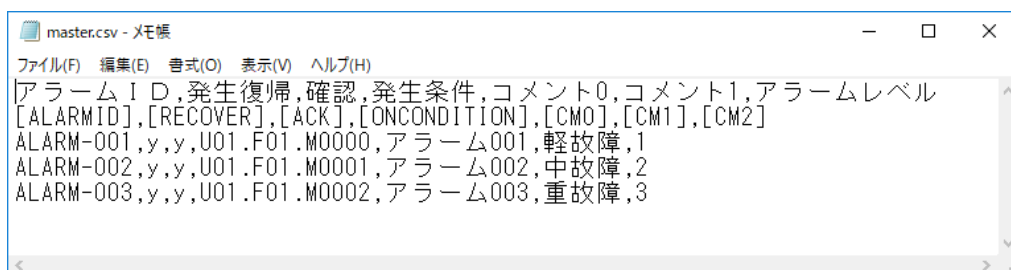
Panel Editor の画面上部にあるブラウザ実行アイコン（緑矢印）をクリックし、Panel Browser を呼び出します。



Panel Browser が起動しました。



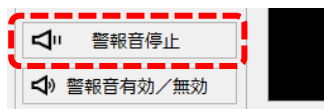
尚、教材サンプルのアラームマスタには、あらかじめ以下のようなアラームが設定されており、U01.F01.M0000 から M0002 までのタグがアラームとして登録されています。



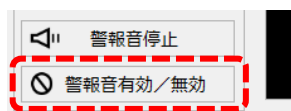
Panel Server から、U01.F01.M0000、M0001、M0002 のうち、任意のタグに True を書き込んでください。すると、アラームサマリにアラームが表示されるとともに、対象のレベルの警報音が鳴動する事を確認してください。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-001	2018/05/24 11:40:35		軽故障	アラーム001

画面左下に配置した警報音停止ボタンをクリックし、警報音が停止することを確認してください。



続いて、警報音の有効/無効の切り替えを試してみます。画面下部に配置した切り替えボタンをクリックし、以下の状態にしてください（※アイコンが丸に斜線の状態が、無効の状態です）。



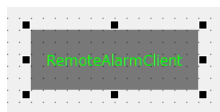
アラームマスタに登録されているタグのうち、任意のタグに True を書き込んでください。すると、アラームサマリにアラームが表示され、警報音は鳴動しない事を確認してください。

AlarmID	発生	確認/復帰	警報レベル	メッセージ
ALARM-002	2018/05/24 11:45:52		中故障	アラーム002

※動作確認が終わったら、画面から確認操作を行い、Panel Server からタグの値を False に戻しておいてください。

■リモートアラームクライアントコントロールについて

リモートアラームクライアントコントロールは、リモートアラームグリッドから表形式の表示機能を除外した部品です。



このコントロールはバックグラウンドでアラームの監視を行いたい場合に使用します。

リモートアラームクライアントは、OnInitialize イベントで Connect メソッドを呼び出すようにします。

```
event OnInitialize()
{
    this.Connect();
    this.SubscribeSummary("");
    this.AddTag("$SOUND0");
}
```

上記で this.Connect() として呼び出すことにより、AlarmServer プロパティで指定された接続先アクション（今回の例では「Alarm」）に対して接続を行います。

Connect の呼び出し後、SubscribeSummary メソッドを呼び出すことにより、アラームサーバ側で検出されたアラームの状態を OnAlarmNotification イベント経由で受け取ることができるようになります。

そして、OnAlarmNotification イベントには以下の処理を記述しました。

```
event OnAlarmNotification(Type,Date,AlarmId,bRecovered,bAcked,Comments,bBrandNew)
{
    var vAlarmLevel = Comments[2];

    if(Type == "new") {
        if(vAlarmLevel >= 1) {
            var vSoundVal = this.GetVal("$SOUND0");
            if(vSoundVal < vAlarmLevel){
                this.WriteVal("$SOUND0", vAlarmLevel);
            }
        }
    }
}
```

OnAlarmNotification イベントでは、呼び出されるタイミングでいくつかのパラメータを受け取ることができます。今回の処理で使用したのは、「Type」と「Comments」です。

Type には、検出されたアラームの状態が渡されます。発生時は「new」という文字列が来るため、上記では Type が new かどうかを条件判定しています。

また、Comments には、アラームマスタで定義されているコメントの内容が配列で渡されます。

```
Comments[0]・・・CM0
Comments[1]・・・CM1
Comments[2]・・・CM2
：　　：　　：
```

今回の処理では、アラームレベル毎にサウンドセレクトの音色を変えるために、CM2 に定義されているアラームレベルを参照しています。

尚、OnInitialize イベントで記述されている AddTag や、本イベントの GetVal、WriteVal などのメソッドは、スクリプトからタグにアクセスするためのメソッドです。アラーム発生時にアラームレベルを受け取り、サウンドセレクト部品をコントロールするための内部タグ（\$SOUND0）に対して目的の値を書き込む事により、サウンドが鳴る仕組みとなっています。



リモートアラームクライアントコントロールに関する詳細は、オンラインマニュアルの「コントロールリファレンス」「データ」「リモートアラームクライアントコントロール」に記述されています。



スクリプトからのタグへのアクセスに関する詳細は、オンラインマニュアルの「画面作成ガイド」「スクリプト」「スクリプトの記述例」「スクリプトからのタグの読み書き」に記述されています。

■サウンドセレクト部品について

サウンドセレクト部品は、サウンドファイルを再生するためのライブラリ部品です。



この部品は、内部タグ「\$SOUND0」と連動して動作します。

\$SOUND0 はサウンドの停止とサウンド選択に使用します。\$SOUND0 に 0 を書き込むとサウンドを停止します。\$SOUND0 に 1 から 10 の整数を書き込むと指定されたサウンド番号のサウンドファイルを再生します。今回の例では、軽故障は 1、中故障は 2、重故障は 3 となっています。



警報音を他の音色に変更したい場合は、サウンドセレクト部品に設定されているサウンドファイルを変更することで対応可能です。サウンドセレクト部品に関する詳細は、オンラインマニュアルの「ライブラリリファレンス」「部品ライブラリ」「基本部品」「サウンドセレクト」に記述されています。



標準フレームワークを使用した標準プロジェクトの場合、アラームの発生に応じて警報音が鳴動する機能が標準で備わっています。そして、標準フレームワークでは、警報音を鳴らすためのロジックは共通アラームサマリのスクリプトに実装されています（※リモートアラームクライアントコントロールは使用していません）。

（共通アラームサマリ表示エリア）



共通アラームサマリ表示エリアは、標準フレームワークのウィザード画面の設定によって非表示にすることができますが、共通アラームサマリを非表示とすると、警報音を鳴らす仕組みが組み込まれないため注意が必要です。従って、標準フレームワークであっても、画面下部の共通アラームサマリ表示を使用しない画面レイアウトでアラーム発生時の警報音を鳴らしたいような場合には、本セクションで紹介した方法で警報音を鳴らす処理を独自に実装する必要があります。

尚、標準フレームワークの場合は、バックグラウンドフォームとして「hidden」フォームがあらかじめ登録されており、hidden フォーム上にはサウンドセレクト部品が張り付けられた状態となっています。従って、標準フレームワークを用いた標準プロジェクトで独自に警報音を鳴らす処理を作り込む場合は、それらの部品を流用して構築するようにしてください。

「FA-Panel6 FC01 機能別コース／アラーム」 Copyright © 2018 Roboticsware, Inc. All rights reserved.

2018 年 5 月 24 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。