



Asix.Evo - Akcje operatorskie

*Dok. Nr PLP7E004
Wersja: 2014-03-25*

ASKOM® i **Asix®** to zastrzeżone znaki firmy **ASKOM Sp. z o. o., Gliwice**. Inne występujące w tekście znaki firmowe bądź towarowe są zastrzeżonymi znakami ich właścicieli.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną lub inną powoduje naruszenie praw autorskich niniejszej publikacji.

ASKOM Sp. z o. o. nie bierze żadnej odpowiedzialności za jakiegokolwiek szkody wynikłe z wykorzystywania zawartych w publikacji treści.

Copyright © 2014, ASKOM Sp. z o. o., Gliwice

ASKOM

ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,
tel. +48 32 3018100, fax +48 32 3018101,

<http://www.askom.com.pl>, e-mail: biuro@askom.com.pl

Spis treści

1 Użycie akcji operatorskich	4
2 Akcje złożone	6
2.1 Parametry akcji złożonych	6
3 Lista akcji operatorskich	8
4 Opisy akcji operatorskich	12
4.1 AcceptAlarm	12
4.2 AcceptAlarmsOnTable	12
4.3 Actions	13
4.4 ActionSet	14
4.5 AddNote	15
4.6 AsbaseAdd	15
4.7 AsbaseClose	16
4.8 AsbaseDelete	17
4.9 AsbaseLoad	17
4.10 AsbaseLoadByValue	19
4.11 AsbaseNavigate	20
4.12 AsbaseOpen	22
4.13 AsbasePrint	23
4.14 AsbaseShow	24
4.15 AsbaseUpdate	25
4.16 Ask	26
4.17 AsreportDisplay	27
4.18 AstrendAsix6	28
4.19 AstrendDisplay	30
4.20 AstrendPrint	32
4.21 Break	33
4.22 CancelControls	34
4.23 ChangePassword	35
4.24 ClearSelection	36
4.25 CloseWindow	36
4.26 ConfirmRole	37
4.27 EditPatterns	39
4.28 EndAlarm	39

4.29 ExcludeAlarm.....	40
4.30 Execute.....	41
4.31 GetVarNames.....	42
4.32 IncludeAlarm.....	42
4.33 Login.....	43
4.34 Logout.....	43
4.35 Message.....	44
4.36 MinimizeWindow.....	45
4.37 NextDiagram.....	46
4.38 Nothing.....	46
4.39 ObjectAction.....	47
4.40 OpenDiagram.....	49
4.41 OpenWindow.....	52
4.42 Perform.....	55
4.43 PlaySound.....	56
4.44 PlaySoundLooping.....	57
4.45 PreviousDiagram.....	58
4.46 PrintScreen.....	58
4.47 RefreshVariable.....	60
4.48 ResetGlobalProperty.....	60
4.49 Run.....	61
4.50 Script.....	61
4.51 ScriptClose.....	62
4.52 ScriptRestart.....	63
4.53 SecurityBrowser.....	64
4.54 SecurityManager.....	65
4.55 SendControls.....	65
4.56 SetBounds.....	66
4.57 SetGlobalProperty.....	67
4.58 SetParameter.....	68
4.59 SetParameters.....	68
4.60 SetPosition.....	69
4.61 SetProperty.....	70
4.62 SetSize.....	71

4.63 SetVariable	72
4.64 SetWindowSize.....	73
4.65 ShowControlPanel	74
4.66 ShowKeyboard	74
4.67 ShowMenu	75
4.68 ShowNumericKeyboard.....	76
4.69 StartAlarm	76
4.70 StopSound	77
4.71 SwitchLanguage.....	78
4.72 SwitchWorkMode.....	78
4.73 SydelShow	79
4.74 Terminate	80
4.75 ToggleBits	80
4.76 VariableInfo	81
4.77 Zoom.....	82
4.78 ZoomIn.....	83
4.79 ZoomOut.....	84

1 Użycie akcji operatorskich

Akcje operatorskie są podstawowym środkiem sterowania pracą aplikacji przez operatora. Akcje są udostępniane operatorowi przez projektanta systemu przy pomocy poniższych mechanizmów:

- Obsługa zdarzeń obiektów i diagramów synoptycznych
- Menu użytkownika
- Terminarz
- Klawisze globalne

Ogólna postać akcji jest następująca:

nazwa_akcji (parametr1, parametr2, ...)

Liczba i znaczenie parametrów zależy od nazwy akcji. W najprostszym przypadku wszystkie parametry mogą być podane bezpośrednio, np.:

```
OpenWindow(OknoP1,PanelGlowny,Pokoj121,"")
```

Powyższa akcja otwiera okno o nazwie *OknoP1*, a w panelu *PanelGlowny* umieszczony zostanie diagram *Pokoj121*.

Każdy z parametrów akcji może być również dowolnym wyrażeniem. W akcjach obowiązują takie same zasady budowania wyrażeń jak we właściwościach obiektów synoptycznych. W momencie wykonania akcji wartości wszystkich użytych wyrażeń są wyliczane i dopiero tak skonstruowana treść akcji podlega wykonaniu, np.

```
OpenWindow(OknoP1,PanelGlowny,Pokoj +Variable(PokNr) , "")
```

Parametr nazwy diagramu jest dynamicznie tworzony na podstawie wartości zmiennej *PokNr*. Jeżeli w momencie wykonania akcji wartość tej zmiennej będzie równa 101, to otwarty zostanie diagram o nazwie *Pokoj101*.

Istotne znaczenie dla wykonania akcji ma również kontekst jej użycia. Akcja wykonywana w obsłudze zdarzenia ma kontekstowy dostęp do właściwości i parametrów obiektów lub diagramów, dla których zdarzenie zostało aktywowane. Akcje wykonywane z terminarza lub klawisza globalnego takiej możliwości nie mają. Akcje wykonywane za pośrednictwem menu użytkownika mogą mieć dostęp do właściwości i parametrów obiektów i diagramów, o ile menu zostało otwarte poprzez akcję *ShowMenu* użytą w obsłudze zdarzenia – w takim przypadku następuje dziedziczenie kontekstu użycia.

```
OpenWindow(OknoP1,PanelGlowny,Pokoj +Variable() , "")
```

W powyższym przypadku nazwa diagramu jest tworzona na podstawie wartości zmiennej głównej obiektu, dla którego zdarzenie zostało aktywowane – bezparametrowa funkcja *Variable* pobiera wartość kontekstowej zmiennej głównej. Jeżeli w momencie wykonania akcji wartość tej zmiennej będzie równa 101, to otwarty zostanie diagram o nazwie *Pokoj101*.

```
OpenWindow(OknoP1,PanelGlowny,Pokoj +Parameter(PokNr) ,"" )
```

W powyższym przypadku nazwa diagramu jest tworzona na podstawie wartości parametru o nazwie *PokNr*. Parametr ten jest parametrem diagramu lub wzorca (też zależnie od kontekstu użycia). Jeżeli w momencie wykonania akcji parametr będzie miał wartość równą 101, to otwarty zostanie diagram o nazwie *Pokoj101*.



Zobacz też

- *AsixEvo_Wyrażenia (PDF/CHM)*

- *AsixEvo_Elementy_wizualizacji (PDF/CHM): 6.3. Standardowe zdarzenia obiektów*

4.2. Zdarzenia diagramów

5. Menu

- *Asix.Evo_Parametryzacja_aplikacji (PDF/CHM): 3. Konfiguracja Terminarza*

2 Akcje złożone

W przypadku , w którym zachodzi potrzeba wykonania sekwencji operacji można stosować akcje złożone. Są to po prostu zestawy zwykłych akcji operatorskich, które są kolejno wykonywane.

Akcje złożone są definiowane w panelu roboczym *Akcje złożone* otwieranym za pośrednictwem drzewa elementów panelu *Eksplorator aplikacji*. Akcja złożona jest uruchamiana przy pomocy akcji operatorskiej *ActionSet*.

Standardowo wykonanie akcji złożonej polega na sekwencyjnym wykonaniu wszystkich akcji składowych. Wykonanie akcji złożonej może być warunkowo przerwane przy pomocy akcji operatorskiej *Break*, np.:

```
Break(Variable(PokNr)==0)
```

Jeżeli wartość zmiennej *PokNr* będzie równa 0, to akcje składowe następujące po akcji *Break* nie zostaną wykonane

Akcje operatorskie wchodzące w skład akcji złożonej posiadają wszystkie cechy zwykłych akcji operatorskich. W szczególności, wartości wyrażeń użytych do zdefiniowania parametrów są wyliczane dopiero w momencie wykonania. Zachodzi też standardowe dziedziczenie kontekstu, jeżeli akcja *ActionSet* została uruchomiona w kontekście obsługi zdarzenia, to w tym samym kontekście zostaną uruchomione wszystkie akcje składowe.

UWAGA:

W prostych przypadkach zamiast akcji złożonej można alternatywnie stosować akcję prostą *Actions*, która także pozwala sekwencyjnie wykonać kilka akcji składowych.

2.1 Parametry akcji złożonych

Cechą specyficzną akcji złożonych są parametry akcji. Nie należy ich mylić z parametrami diagramów i wzorców dostępnymi poprzez funkcję *Parameter*. W trakcie definiowania akcji złożonej można zdefiniować parametry akcji. Parametry identyfikowane są poprzez nazwę. Następnie w dowolnym miejscu akcji składowych można odwołać się do parametru poprzez wstawienie do treści akcji nazwy parametru poprzedzonej znakiem @.

Proces wykonania akcji złożonej wyjaśnia poniższy przykład.

Warunki wstępne:

- Akcja złożona *A1* ma zadeklarowane użycie parametrów o nazwie *Pokoj* i *Nastawa*
- Jedna z akcji składowych jest następującej postaci:

SetVariable(Nastawy+@Pokoj,@Nastawa)

- Akcja złożona jest uruchomiona poprzez akcję wykonaną w kontekście obsługi zdarzenia:

ActionSet(A1,PokNr+Parameter(PokNumer),"10+1")

Wykonanie:

- W momencie wykonania akcji *ActionSet* wyliczana jest wartość drugiego parametru akcji poprzez połączenie tekstu „PokNr” i wartości parametru (diagramu lub wzorca) o nazwie *PokNumer*. Wartość wyliczonego parametru jest przekazywana w postaci tekstowej do akcji złożonej jako parametr *Pokoj*. Załóżmy, że wyliczona wartość jest równa „PokNr99”. Trzeci parametr akcji *ActionSet* przekazywany jest w niezmięnionej formie „10+1” jako parametr *Nastawa*.
- W momencie wykonania akcji składowej *SetVariable* nazwa zmiennej wyliczana jest poprzez złożenie tekstu „Nastawy” i wartości parametru akcji złożonej *Pokoj* (uzyskana zostanie nazwa *NastawyPokNr99*). Wartość sterowana jest bezpośrednio wartością parametru *Nastawa* i będzie równa tekstowi „10+1”.
- W efekcie nastąpi ustawienie zmiennej *NastawyPokNr99* na wartość tekstową „10+1”.

Jeżeli celem było ustawienie wartości 11, to należało użyć akcję *ActionSet* w postaci:

ActionSet(A1,PokNr+Parameter(PokNumer),10+1)

W tym przypadku wyliczenie wartości wyrażenia *10+1* nastąpiłoby przed wykonaniem akcji złożonej.

3 Lista akcji operatorskich

Zarządzanie oknami synoptycznymi

CloseWindow – zamykanie okien synoptycznych

MinimizeWindow - minimalizacja okien synoptycznych

NextDiagram – przełączenie na następny diagram zgodnie zapamiętaną historią wyświetlanych diagramów

OpenDiagram – otwarcie diagramu synoptycznego w dynamicznie tworzonym oknie

OpenWindow – otwarcie predefiniowanego okna synoptycznego lub podmiana diagramu w panelu okna wcześniej otwartego

PreviousDiagram – przełączenie na poprzednio wyświetlany diagram

SetWindowSize – zmiana rozmiaru okien synoptycznych

Zoom – bezwzględna zmiana skalowania diagramu

ZoomIn – względne powiększenie skalowanego diagramu

ZoomOut – względne pomniejszenie skalowanego diagramu

Dostęp do zmiennych procesowych i kontrola operacji sterujących

CancelControls – anulowanie oczekujących operacji sterujących

RefreshVariable – wymuszenie odświeżenia wartości zmiennej procesowej

SendControls – wykonanie oczekujących operacji sterujących

SetVariable – ustawienie wartości zmiennej procesowej

ToggleBits – zmiana bitów w wartości zmiennej procesowej

VariableInfo – wyświetlenie okna z informacjami dotyczącymi wybranej zmiennej procesowej

Sterowanie pracą aplikacji

ShowControlPanel – otwarcie okna panelu kontrolnego

SwitchLanguage – zmiana aktualnego języka pracy aplikacji

SwitchWorkMode – przełączenie trybu pracy aplikacji

Terminate – zakończenie pracy aplikacji

Sterowanie wykonaniem akcji operatorskich

Actions – wykonanie sekwencji akcji składowych

ActionSet – uruchomienie akcji złożonej

Ask – wyświetlenie okienka z zapytaniem, akcja złożona jest przerywana w przypadku negatywnej odpowiedzi użytkownika

Break – warunkowe przerywanie wykonywania akcji złożonej

Execute – wykonanie akcji, której treść jest konstruowana dynamicznie

Nothing – akcja pusta

Perform – warunkowe wykonanie akcji

Sygnalizacja dźwiękowa

PlaySound – odtworzenie pliku dźwiękowego

PlaySoundLooping - odtworzenie pliku dźwiękowego z powtarzaniem

StopSound – przerywanie odtwarzania dźwięku

Współpraca z systemem alarmów

AcceptAlarm – potwierdzenie alarmu

AcceptAlarmsOnTable - potwierdzenie alarmów pokazywanych w tabeli alarmów

EndAlarm – zakończenie alarmu

ExcludeAlarm – wykluczenie alarmu z obsługi

IncludeAlarm – ponowne włączenie alarmu do obsługi

StartAlarm – rozpoczęcie alarmu

Współpraca z systemem uprawnień

AddNote - wprowadzenie notatki operatora

ChangePassword – zmiana hasła użytkownika

ConfirmRole – weryfikacja przynależności użytkownika do roli

Login – zalogowanie użytkownika

Logout – wylogowanie użytkownika

SecurityBrowser – otwarcie okna przeglądania logów gromadzonych przez system zabezpieczeń

SecurityManager – otwarcie okna zarządzania uprawnieniami użytkowników

Uruchomienie składników aplikacji

AsreportDisplay – uruchomienie okna przeglądarki raportów opartych na technologii Reporting Services

Run – uruchomienie zewnętrznego programu lub pliku

Script – uruchomienie skryptu użytkownika

ScriptClose – zakończenie pracy skryptu

ScriptRestart – zamknięcie i ponowne uruchomienie skryptu użytkownika

ShowMenu – otwarcie menu kontekstowego

SydelShow – przełączenie wyświetlanego węzła w programie SydelEko

Sterowanie pracą programu AsTrend

AstrendAsix6 – sterowanie programem AsTrend. Akcja używana tylko przy automatycznej konwersji aplikacji z systemu Asix6.

AstrendDisplay – wyświetlenie trendu przy pomocy program AsTrend

AstrendPrint – wydrukowanie trendu przy pomocy program AsTrend

Komunikacja z programem AsBase

AsbaseAdd – dodanie nowego rekordu w ramach połączenia

AsbaseClose - zamknięcie połączenia pomiędzy aplikacją Asix.Evo a programem AsBase

AsbaseDelete – usunięcie bieżącego rekordu w ramach połączenia

AsbaseLoad – załadowanie wartości z wybranej receptury do zestawu zmiennych procesowych. Wybór receptury wykonywany jest na podstawie jej nazwy.

AsbaseLoadByValue – załadowanie wartości z wybranej receptury do zestawu zmiennych procesowych. Wybór receptury wykonywany jest na podstawie wartości dowolnego pola.

AsbaseNavigate – sterowanie przeglądaniem i zarządzanie danymi w ramach połączenia.

AsbaseOpen – utworzenie połączenia pomiędzy aplikacją Asix.Evo a programem AsBase

AsbasePrint - wydrukowanie wskazanych danych z archiwów programu AsBase

AsbaseShow – wyświetlenie okna programu AsBase z przełączeniem na żądany zestaw danych

AsbaseUpdate – zmiana wartości pól bieżącego rekordu w ramach połączenia

Pomocnicze

ClearSelection – usunięcie stanu selekcji obiektu diagramu

EditPatterns – otwarcie okna edytora krzywych wzorcowych

GetVarNames – zapisuje do schowka nazwy zmiennych używanych przez obiekty znajdujące się pod kursorem myszki

Message – zapisanie informacji do dziennika zdarzeń

ObjectAction – wykonanie operacji na wybranych obiektach

PrintScreen – zapisanie obrazu aktualnego okna, diagramu lub ekranu w schowku systemowym lub w pliku

ResetGlobalProperty – usuwa wykonane wcześniej modyfikacje wartości właściwości globalnej

SetBounds – zmiana położenia i rozmiaru obiektu, wzorca lub grupy obiektów

SetGlobalProperty – zmienia wartość właściwości globalnej

SetParameter – zmiana wartości pojedynczego parametru diagramu

SetParameters – zmiana wartości parametrów diagramu

SetPosition – zmiana położenia obiektu, wzorca lub grupy obiektów

SetProperty – ustawienie właściwości obiektu lub diagramu synoptycznego

SetSize – zmiana rozmiaru obiektu lub wzorca

ShowKeyboard – wyświetlenie okna klawiatury ekranowej

ShowNumericKeyboard – wyświetlenie okna numerycznej klawiatury ekranowej

4 Opisy akcji operatorskich

4.1 AcceptAlarm

Przeznaczenie

Akcja służy do potwierdzenia alarmu przez użytkownika. Alternatywną metodą programowego potwierdzenia alarmu jest użycie w funkcji *Accept* interfejsu *IAlarm* w kodzie skryptu użytkownika. Alarmy mogą być także potwierdzane z wykorzystaniem obiektu *Tabela alarmów aktywnych*.

Składnia

AcceptAlarm(*nazwa_domeny*, *identyfikator_alarmu*)

Parametry

nazwa_domeny

Nazwa domeny alarmów, której dotyczy akcja. Jeżeli nazwa nie jest podana, to akcja dotyczy domeny domyślnej (pierwszej).

identyfikator_alarmu

Identyfikator alarmu, który ma zostać potwierdzony.



Zobacz też

- System alarmów (*AsixEvo_System_alarmów.PDF/CHM*)
- obiekt *Tabela alarmów aktywnych* (*AsixEvo_Objekty.PDF/CHM*)
- Interfejs *IAlarm* (*AsixEvo_Skrypty.PDF/CHM*)

4.2 AcceptAlarmsOnTable

Przeznaczenie

Akcja służy do potwierdzenia alarmów pokazywanych aktualnie w otwartych tabelach alarmów aktywnych. Jest to alternatywa do mechanizmów potwierdzania wbudowanych w interfejs użytkownika tabel alarmów. W parametrze akcji można określić, których tabel akcja dotyczy.

Składnia

AcceptAlarmsOnTable(*nazwa_tabeli*)

Parametry***nazwa_tabeli***

Nazwa tabeli alarmów aktywnych, dla której należy potwierdzić alarmy. Można stosować znaki specjalne * i ?. W szczególności, nazwa * oznacza potwierdzenie wszystkich alarmów pokazywanych przez wszystkie otwarte tabele.

**Zobacz też**

- System alarmów (*AsixEvo_System_alarmów.PDF/CHM*)
- obiekt *Tabela alarmów aktywnych* (*AsixEvo_Objekty.PDF/CHM*)
- akcja [AcceptAlarm](#)

4.3 Actions**Przeznaczenie**

Akcja służy do sekwencyjnego wykonania zestawu akcji składowych. W swoim działaniu jest zbliżona do sposobu działania akcji złożonych. Można ją stosować w prostszych przypadkach, gdy ilość akcji składowych i ich złożoność nie są zbyt duże.

Składnia

Actions(akcja1, akcja2, ...)

Parametry

akcja1, ..., akcjaN

Akcje składowe, które zostaną kolejno wykonane.

Przykłady

Actions(SetVariable(v1,1), SetVariable(v2,1))

Akcja uruchamia kolejno dwie akcji składowe, które ustawiają wartości dwóch zmiennych procesowych.



Zobacz też

[2. Akcje złożone](#)

4.4 ActionSet

Przeznaczenie

Akcja służy do uruchomienia akcji złożonej.

Składnia

ActionSet(*nazwa* , *parametr1*, *parametr2*, ...)

Parametry

nazwa

Nazwa akcji złożonej.

parametr1, ..., parametrN

Wartości parametrów akcji złożonej. Liczba parametrów użytych w wywołaniu musi być zgodna z liczbą parametrów w definicji akcji złożonej. Kolejne parametry wywołania zostaną podstawione do parametrów akcji złożonej zgodnie z kolejnością ich występowania w definicji akcji złożonej.

Przykłady

ActionSet(wyłącz_agregat, agr100, agr101)

Akcja uruchamia akcję złożoną o nazwie *wyłącz_agregat*. Zakładając, że akcja *wyłącz_agregat* posiada dwa parametry o nazwach *zmienna1* i *zmienna2* zdefiniowane w podanej kolejności, to w momencie wykonania akcji parametr *zmienna1* będzie równy *agr100*, a parametr *zmienna2* będzie równy *agr101*.



Zobacz też

[2. Akcje złożone](#)

4.5 AddNote

Przeznaczenie

Akcja powoduje otwarcie okna dialogowego, w którym operator systemu może wprowadzić notatkę tekstową.

Składnia

AddNote (*identyfikator_segmentu*)

Parametry

Identyfikator_segmentu

W parametrze można podać identyfikator segmentu, z którym wprowadzona notatka będzie skojarzona. Wartość parametru może być pusta.



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 5. System uprawnień

4.6 AsbaseAdd

Przeznaczenie

Akcja służy do dopisania nowego rekordu w połączeniu utworzonym wcześniej przy pomocy akcji *AsbaseOpen*. Wartości pól nowego rekordu wynikają z wartości zmiennych procesowych należących do zestawu zmiennych użytego w połączeniu.

Składnia

AsbaseAdd (*identyfikator_połączenia*)

Parametry

Identyfikator_połączenia

W parametrze należy podać identyfikator, który został wcześniej użyty w akcji *AsbaseOpen*.



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*
- akcja [AsbaseOpen](#)

4.7 AsbaseClose

Przeznaczenie

Akcja służy utworzenia zamknięcia połączenia utworzonego wcześniej przy pomocy akcji *AsbaseOpen*.

Akcję *AsbaseClose* należy użyć w obsłudze zdarzenia *Zamknięcie diagramu* diagramu, dla którego połączenie zostało otwarte.

Składnia

AsbaseClose (identyfikator_połączenia)

Parametry

Identyfikator_połączenia

W parametrze należy podać identyfikator, który został wcześniej użyty w akcji *AsbaseOpen*.



Zobacz też

- [akcja AsbaseOpen](#)
- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*

4.8 AsbaseDelete

Przeznaczenie

Akcja służy do usunięcia bieżącego rekordu w połączeniu utworzonym wcześniej przy pomocy akcji *AsbaseOpen*.

Składnia

AsbaseDelete (*identyfikator_połączenia*)

Parametry

Identyfikator_połączenia

W parametrze należy podać identyfikator, który został wcześniej użyty w akcji *AsbaseOpen*.



Zobacz też

- [akcja *AsbaseOpen*](#)
- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*

4.9 AsbaseLoad

Przeznaczenie

Akcja pozwala na załadowanie nastaw pochodzących z rekordu w wybranej grupie receptur do zmiennych procesowych skojarzonych z tą grupą w programie AsBase. Rekord nastaw może być wybierany poprzez bezpośrednie podanie nazwy receptury, lub może być wykonana interaktywna selekcja nazwy.

W przypadku korzystania z trybu połączeniowego jako akcję ładującą recepturę należy zawsze stosować akcję *AsbaseNavigate* z opcją *\$Load*.

Składnia

AsbaseLoad(*identyfikator_grupy_receptur*, *identyfikator_zestawu_zmiennych*, *nazwa_receptury*)

Parametry

identyfikator_grupy_receptur

Identyfikator grupy receptur skąd pobierane są wartości pól receptury

identyfikator_zestawu_zmiennych

Identyfikator zestawu zmiennych, do którego zmiennych zostaną wysłane wartości pochodzące z wybranej receptury.

nazwa_receptury

Nazwa receptury, która ma zostać załadowana. Można podać nazwę dokładną lub wzorzec nazwy zawierający znaki specjalne * i ?. Jeżeli nazwa receptury nie zostanie podana lub zawiera znaki specjalne, to wyświetlone zostanie okno z listą receptur (wszystkich lub zgodnych z wzorcem nazwy) zdefiniowanych w grupie receptur, z której użytkownik będzie mógł wybrać recepturę, która ma być załadowana.

Przykłady

```
AsbaseLoad( Nastawy,Zbiornik1 ,"" )
```

Akcja spowoduje wyświetlenie okienka z listą nazw wszystkich receptur w grupie Nastawy. Użytkownik będzie mógł wybrać recepturę, której wartości pól mają być załadowane do zmiennych procesowych z zestawu *Zbiornik1*.



Zobacz też

- [akcja *AsbaseNavigate*](#), akcja [AsbaseLoadByValue](#)

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*

4.10 AsbaseLoadByValue

Przeznaczenie

Akcja pozwala na załadowanie nastaw pochodzących z rekordu w wybranej grupie receptur do zmiennych procesowych skojarzonych z tą grupą w programie AsBase. Rekord nastaw jest wybierany poprzez podanie wartości referencyjnej dla dowolnego pola receptury.

Składnia

AsbaseLoadByValue(*identyfikator_grupy_receptur*, *identyfikator_zestawu_zmiennych*, *identyfikator_pola*, *wartość_pola*)

Parametry

identyfikator_grupy_receptur

Identyfikator grupy receptur, skąd pobierane są wartości pól receptury.

identyfikator_zestawu_zmiennych

Identyfikator zestawu zmiennych, do którego zmiennych zostaną wysłane wartości pochodzące z wybranej receptury.

identyfikator_pola

Identyfikator pola receptury, dla którego zostanie wykonany test zgodności wartości. Oprócz identyfikatorów pól zdefiniowanych przez użytkownika, można użyć identyfikator *Name*, oznaczający test zgodności nazwy receptury.

wartość_pola

Wartość referencyjna dla pola podanego w parametrze *identyfikator_pola*. Załadowana zostanie receptura, dla której znajdzie zgodność wartości referencyjnej z wartością zadeklarowanego pola.

UWAGA:

Istotną różnicą w sposobie wykonania akcji *AsbaseLoadByValue* w stosunku do akcji *AsbaseLoad* jest to, że fakt załadowania receptury do zmiennych nie jest rejestrowany w tabeli wykonanych ładowań dla grupy receptur. Akcja *AsbaseLoadByValue* z użyciem pola *Name* działa identycznie jak akcja *AsbaseLoad*, za wyjątkiem zarejestrowania faktu wykonania operacji.

Przykłady

```
AsbaseLoadByValue( Nastawy,Preview ,ProdId,Variable(PID) )
```

Do zmiennych zestawu *Preview* zostaną załadowane wartości z rekordu grup receptur *Nastawy*, dla którego wartość pola *ProdId* będzie zgodna z wartością zmiennej *PID*.



Zobacz też

- akcja [AsbaseNavigate](#), akcja [AsbaseLoad](#)

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. Współpraca z programem AsBase

4.11 AsbaseNavigate

Przeznaczenie

– sterowanie przeglądaniem i zarządzanie danymi w ramach połączenia.

Akcja służy do sterowania przeglądaniem i zarządzania danymi w ramach połączenia utworzonego wcześniej przy pomocy akcji *AsbaseOpen*. Akcja pozwala na zmianę bieżącego rekordu w połączeniu, wykonanie operacji załadowania receptury i zmianę parametrów połączenia.

Składnia

AsbaseNavigate (identyfikator_połączenia, typ_operacji, parametry_operacji)

Parametry

Identyfikator_połączenia

W parametrze należy podać identyfikator, który został wcześniej użyty w akcji *AsbaseOpen*.

typ_operacji

Parametr określa, jaka operacja ma być wykonana w ramach połączenia. Możliwe wartości to jedna z poniższych stałych:

\$Count – powoduje przeliczenie ilości rekordów widocznych w połączeniu zgodnie z aktualnym filtrem i umieszczenie tej liczby w odpowiedniej zmiennej zestawu zmiennych.

\$Filter – powoduje zdefiniowanie nowego filtra oraz przejście do ostatniego rekordu połączenia po zastosowaniu filtra. Szczegóły dotyczące składni filtra znajdują się w rozdziale Współpraca z programem AsBase.

\$First – powoduje przejście do pierwszego rekordu w ramach połączenia.

\$Download – wysłanie zawartości bieżącego rekordu do zmiennych z zestawu zmiennych podanego jawnie w parametrze *parametry_operacji*.

\$Last– powoduje przejście do ostatniego rekordu w ramach połączenia.

\$Load – powoduje przejście do rekordu wskazanego nazwą receptury lub wybranego interaktywnie przez użytkownika. Operacja dostępna tylko dla połączeń dotyczących grupy receptur.

\$Next– powoduje przejście do następnego rekordu w ramach połączenia.

\$Previous– powoduje przejście do poprzedniego rekordu w ramach połączenia.

parametry_operacji

W przypadku operacji *\$Filter* parametr określa nową treść filtra. W przypadku operacji *\$Download* parametr określa nazwę zestawu zmiennych, do którego należy wysłać wartości z bieżącego rekordu.

W przypadku operacji *\$Load* w parametrze należy podać nazwę receptury. Można podać nazwę dokładną lub wzorzec nazwy zawierający znaki specjalne * i ?. Jeżeli nazwa receptury nie zostanie podana lub zawiera znaki specjalne, to zostanie wyświetlona lista nazw receptur (wszystkich lub zgodnych z wzorcem nazwy), z której użytkownik będzie mógł wybrać recepturę.

Przykłady

AsbaseNavigate(conn1, \$Next, null)

Akcja powoduje przejście do następnego rekordu w ramach danych wynikających z połączenia *conn1*. Do zmiennych należących do zestawu zmiennych połączenia zostaną wpisane wartości pobrane z nowego bieżącego rekordu.

AsbaseLoad(conn1, \$Load, RC_123)

Akcja powoduje przejście do receptury o nazwie *RC_123* w ramach grupy receptur wynikającej z połączenia *conn1*. Do zmiennych należących do zestawu zmiennych połączenia zostaną wpisane wartości pobrane z nowego bieżącego rekordu.



Zobacz też

- [akcja AsbaseOpen](#)

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. Współpraca z programem AsBase

4.12 AsbaseOpen

Przeznaczenie

Akcja służy utworzenia połączenia służącego do wymiany danych pomiędzy aplikacją Asix.Evo a programem AsBase. W parametrach akcji określa się, które dane będą wymieniane i przez jakie zmienne procesowe. Po wykonaniu akcji dane z bieżącego rekordu połączenia będą zapisywane w zmiennych interfejsowych. Użycie akcji wymaga wcześniejszego sparametryzowania aplikacji AsBase.

Akcję *AsbaseOpen* należy użyć w obsłudze zdarzenia *Otwarcie diagramu* dla diagramu który musi pozostać otwarty przez cały czas trwania wymiany danych. W zdarzeniu *Zamknięcie diagramu* w celu zakończenia połączenia należy użyć akcję *AsbaseClose*.

Składnia

AsbaseOpen (identyfikator_połączenia, typ_danych, identyfikator_źródła, identyfikator_zestawu_zmiennych, identyfikator_podrzednego_źródła, identyfikator_podrzednego_zestawu_zmiennych, filtr)

Parametry

Identyfikator_połączenia

W parametrze należy podać identyfikator, który będzie jednoznacznie identyfikował połączenie z określonym źródłem danych i zestawem zmiennych. Identyfikator wykorzystywany jest później w innych akcjach z rodziny *AsbaseXxx*.

typ_danych

Parametr określa typ źródła danych. Możliwe wartości to jedna z poniższych stałych:

\$Archive – połączenie dotyczy zestawu rejestracji

\$Recipes – połączenie dotyczy grupy receptur

\$Loads – połączenie dotyczy historii ładowań receptur

identyfikator_źródła

Identyfikator źródła danych, w zależności od wartości parametru *typ_danych*, jest to identyfikator zestawu rejestracji lub grupy receptur.

identyfikator_zestawu_zmiennych

Identyfikator zestawu zmiennych, za pośrednictwem którego będzie wykonywana komunikacja pomiędzy aplikacją Asix.Evo a programem AsBase.

identyfikator_podrzednego_zrodla

Identyfikator podrzędnego źródła danych, czyli źródła, które w parametryzacji programu AsBase zostało zadeklarowane jako pozostające w relacji ze źródłem podstawowym.

identyfikator_podrzednego_zestawu_zmiennych

Identyfikator zestawu zmiennych, za pośrednictwem którego będzie wykonywana komunikacja pomiędzy aplikacją Asix.Evo a programem AsBase. Zestaw przeznaczony dla komunikacji ze źródłem podrzędnym.

filtr

Wyrażenie filtrujące, które pozwala na ograniczenie zakresu danych dostępnych poprzez połączenie. Szczegóły dotyczące składni filtra znajdują się w rozdziale *Współpraca z programem AsBase*.

Przykłady

```
AsbaseOpen(conn1, $Archive, Farby, FarbyDgr, null, null, "V_Akt=1" )
```

Akcja tworzy połączenie o nazwie *conn1*. W ramach tego połączenie dostępne będą rekordy danych z zestawu rejestracji o identyfikatorze *Farby*, spełniające warunek mówiący o tym, że wartość pola zestawu o identyfikatorze *Akt* będzie równa 1. Dane będą wymieniane poprzez zestaw zmiennych o identyfikatorze *FarbyDgr*. W zmiennych należących do tego zestawu będą zapisywane wartości aktywnego rekordu połączenia.

**Zobacz też**

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*
- [akcja AsbaseClose](#), [akcja AsbaseAdd](#), [akcja AsbaseDelete](#), [akcja AsbaseNavigate](#), [akcja AsbaseUpdate](#)

4.13 AsbasePrint**Przeznaczenie**

Akcja powoduje wydrukowanie przez program AsBase zawartości wybranych tabel.

Składnia

AsbasePrint (typ_danych, identyfikator_zrodla , identyfikator_widoku)

Parametry

typ_danych

Parametr określa typ źródła danych. Możliwe wartości to jedna z poniższych stałych:

\$Archive – połączenie dotyczy zestawu rejestracji

\$Recipes – połączenie dotyczy grupy receptur

\$Loads – połączenie dotyczy historii ładowań receptur

identyfikator_źródła

Identyfikator źródła danych, w zależności od wartości parametru *typ_danych*, jest to identyfikator zestawu rejestracji lub grupy receptur.

identyfikator_widoku

Identyfikator widoku skojarzonego w programie Asbase ze źródłem danych. Widok określa sposób wydrukowania rekordów, może też zawierać informacje służące do filtrowania danych.

Przykłady

```
AsbasePrint( $Recipes, Nastawy, "" )
```

Akcja spowoduje wydrukowanie wszystkich rekordów z grupy receptur *Nastawy* w układzie domyślnym.



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*

4.14 AsbaseShow

Przeznaczenie

Akcja powoduje uaktywnienie okna programu AsBase (wyciągnięcie go na wierzch). W parametrach akcji można określić, które dane mają zostać pokazane.

Składnia

AsbaseShow (*typ_danych*, *identyfikator_źródła*, *identyfikator_zestawu_zmiennych*, *identyfikator_widoku*)

Parametry

typ_danych

Parametr określa typ źródła danych. Możliwe wartości to jedna z poniższych stałych:

\$Archive – połączenie dotyczy zestawu rejestracji

\$Recipes – połączenie dotyczy grupy receptur

\$Loads – połączenie dotyczy historii ładowań receptur

identyfikator_źródła

Identyfikator źródła danych - w zależności od wartości parametru *typ_danych* jest to identyfikator zestawu rejestracji lub grupy receptur.

identyfikator_zestawu_zmiennych

Identyfikator zestawu zmiennych. Ustawiany tylko wtedy, gdy chce się wyświetlić okno podglądu wartości zmiennych dla zestawu zmiennych.

identyfikator_widoku

Identyfikator widoku skojarzonego w programie AsBase ze źródłem danych. Widok określa sposób wyświetlania rekordów, może też zawierać informacje służące do filtrowania danych.

Przykłady

```
AsbaseShow( $Recipes, Nastawy, "", V1 )
```

Akcja spowoduje wyświetlenie w oknie programu AsBase danych należących do grupy receptur *Nastawy*. Sposób wyświetlenia oraz zakres danych jest określony przez zdefiniowany w aplikacji AsBase widok o identyfikatorze *V1*.



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*

4.15 AsbaseUpdate

Przeznaczenie

Akcja służy do zmiany wartości pól bieżącego rekordu w połączeniu utworzonym wcześniej przy pomocy akcji *AsbaseOpen*. Nowe wartości pól rekordu wynikają z wartości zmiennych procesowych należących do zestawu zmiennych użytego w połączeniu.

Składnia

AsbaseUpdate (*identyfikator_połączenia*)

Parametry

Identyfikator_połączenia

W parametrze należy podać identyfikator, który został wcześniej użyty w akcji *AsbaseOpen*.



Zobacz też

- [akcja *AsbaseOpen*](#)

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 8. *Współpraca z programem AsBase*

4.16 Ask

Przeznaczenie

Akcja przeznaczona do użytku w akcjach złożonych. Wyświetla okienko komunikatów z zapytaniem podanym w parametrze akcji oraz dwoma przyciskami: *OK* i *Anuluj*. Jeżeli użytkownik wybierze przycisk *Anuluj*, to dalsze wykonanie akcji złożonej jest przerywane.

Składnia

Ask (*pytanie*)

Ask (*pytanie, czas_oczekiwania, wartość_domyślna*)

Parametry

pytanie

Treść zapytania wyświetlonego w okienku komunikatu.

czas_oczekiwania

Okres czasu wyrażony w sekundach, po którym okno zapytania zostanie zamknięte, jeżeli użytkownik nie wybierze wcześniej żadnej odpowiedzi. W wariancie akcji bez podania czasu oczekiwania, okienko pozostaje otwarte bezterminowo.

wartość_domyślna

Domyślny rezultat zapytania (*true/false*), który zostanie użyty, gdy użytkownik nie wybierze żadnej odpowiedzi.

Przykłady

```
Actions( Ask("Włączyć pompę?"),SetVariable(v1,1))
```

Operacja ustawienia zmiennej v1, zostanie wykonana tylko wtedy, gdy użytkownik odpowie twierdząco na zapytanie.



Zobacz też

[Akcje złożone](#), [akcja Perform](#), [akcja Actions](#), [akcja Break](#)

4.17 AsreportDisplay

Przeznaczenie

Akcja służy do otwarcia okna przeglądarki raportów utworzonych w technologii Reporting Services z wykorzystaniem modułu AsRaport pakietu Asix. Okno pozwala na generowanie i przeglądanie wybranych raportów. Parametry generacji mogą być zadawane w akcji lub określane w sposób interaktywny.

Użycie akcji wymaga ustawienia parametrów dostępu do serwera usług raportowych. Parametry te podawane są w panelu roboczym *Ustawienia stanowisk* w zakładce *Źródła danych*.

Oprócz parametrów podanych bezpośrednio w treści akcji, na dostępność niektórych funkcji okna przeglądarki mają wpływ także uprawnienia(rola) aktualnie zalogowanego użytkownika.

Składnia

AsreportDisplay (nazwa_raportu, parametry,pokazuj_parametry, współrzędna_X, współrzędna_Y, katalog_raportów)

Parametry***nazwa_raportu***

Parametr służy do podania nazwy raportu, który należy wyświetlić w oknie przeglądarki. Nazwa musi być zgodna z nazwą raportu opublikowanego na serwerze i może zawierać

ścieżkę względną do katalogu raportu. Parametr może być pusty – użytkownik będzie mógł wybrać raport interaktywnie.

Parametry

Parametr służy do określenia wartości parametrów generacji raportu. Zestaw możliwych parametrów wynika z definicji raportu. Parametr ma postać tekstu złożonego z par postaci *@nazwa_parametry=wartość_parametru* rozdzielonych znakiem spacji, np. „@month=12 @section=A”.

pokazuj_raporty

Parametr typu logicznego *true/false*, który określa, czy wyświetlać panel edycji parametrów raportu.

współrzędna_X

Parametr określa współrzędną X okna przeglądarki raportów (lewego górnego narożnika).

współrzędna_Y

Parametr określa współrzędną Y okna przeglądarki raportów (lewego górnego narożnika).

katalog_raportów

Nazwa podkatalogu (lub ścieżka względna podkatalogu) na serwerze raportów, którego zawartość (nazwy raportów oraz podkatalogi) należy umieścić w polu wyboru raportu okna przeglądarki. Wartość * oznacza, że należy pokazać wszystkie raporty, a wartość pusta oznacza, że pole wyboru raportu nie będzie pokazywane.

4.18 AstrendAsix6

Przeznaczenie

Akcja służy do sterowania pracą programu AsTrend. Jest do dokładny odpowiednik akcji *AsTrend* systemu Asix6. Akcja jest używana w przypadku automatycznej konwersji aplikacji systemu Asix6. W aplikacjach tworzonych od podstaw należy używać akcji *AstrendDisplay* i *AstrendPrint*.

Składnia

AstrendAsix6 (nazwa_trendu, czas_początku, współrzędna_X, współrzędna_Y, monitor, parametry_dodatkowe)

Parametry

nazwa_trendu

Nazwa pliku z definicją trendu. Wpisanie znaku "?" powoduje otwarcie okna, w którym użytkownik może wybrać trend do wyświetlenia.

Po nazwie pliku trendu można podać nazwy zmiennych, które zostaną dodane do wykresu zmiennych zadeklarowanych w definicji trendu. Jeżeli nazwy zmiennych będą rozdzielone znakami #, to zmienne zostaną dodane do trendu. Jeżeli nazwy zmiennych będą rozdzielone znakami <, to zmienne zastąpią zmienne znajdujące się w definicji trendu.

czas_początku

Określenie czasu początku wyświetlanego trendu. Parametr może być czasem bezwzględnym w postaci *yyyymmddhhnss*, czasem w formacie OPC lub nazwą zmiennej procesowej. W sytuacji, gdy parametrem jest nazwą zmiennej, to jej wartość interpretowana jest jako ilość sekund od 1.1.1970 dla zmiennych o wartościach liczbowych, a dla zmiennych tekstowych jako czas w postaci *yyyymmddhhnss* lub w formacie OPC. Możliwe jest też podanie czasów początku i końca trendu, rozdzielając je znakiem średnika.

współrzędna_X

Parametr określa współrzędną X okna programu AsTrend (lewego górnego narożnika). Wartość -1 oznacza pobranie odpowiedniej współrzędnej z definicji trendu.

współrzędna_Y

Parametr określa współrzędną Y okna programu AsTrend (lewego górnego narożnika). Wartość -1 oznacza pobranie odpowiedniej współrzędnej z definicji trendu.

monitor

W parametrze należy podać numer monitora, na którym powinno otworzyć się okno programu AsTrend. Wartość 0 oznacza monitor, na którym aktualnie znajduje się kursor myszki, -1 oznacza monitor wynikający z przyjętych współrzędnych x, y.

parametry_dodatkowe

Dodatkowe parametry akcji zgodne z opisem akcji *AsTrend* systemu Asix6.



Zobacz też

- [akcja *AsTrendDisplay*](#), [akcja *AsTrendPrint*](#)

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 7. *Współpraca z programem AsTrend*

4.19 AstrendDisplay

Przeznaczenie

Akcja służy do otwarcia okna programu AsTrend z wyświetlonym wybranym trendem. W akcji można podać nazwę wzorca trendu i czas początku trendu. Opcjonalnie można podać nazwy zmiennych wyświetlonych w trendzie.

Składnia

AstrendDisplay (nazwa_trendu, czas_początku, współrzędna_X, współrzędna_Y, monitor)

AstrendDisplay (nazwa_trendu, czas_początku, czas_końca, współrzędna_X, współrzędna_Y, monitor)

Parametry

nazwa_trendu

Nazwa pliku z definicją trendu. Wpisanie znaku "?" powoduje otwarcie okna, w którym użytkownik może wybrać trend do wyświetlenia. Brak nazwy spowoduje otwarcie okna programu AsTrend z zachowaniem poprzednio używanego trendu.

Po nazwie pliku trendu można podać nazwy zmiennych, które zostaną dodane do wykresu zmiennych zadeklarowanych w definicji trendu. Jeżeli nazwy zmiennych będą rozdzielone znakami #, to zmienne zostaną dodane do trendu. Jeżeli nazwy zmiennych będą rozdzielone znakami <, to zmienne zastąpią zmienne znajdujące się w definicji trendu.

czas_początku

Określenie czasu początku wyświetlanego trendu. Parametr może być wartością typu *DateTime*, czasem bezwzględnym w postaci *yyyymmddhhnnss*, czasem w formacie OPC lub nazwą zmiennej procesowej. W sytuacji, gdy parametrem jest nazwą zmiennej, to jej wartość interpretowana jest jako ilość sekund od 1.1.1970 dla zmiennych o wartościach liczbowych, a dla zmiennych tekstowych jako czas w postaci *yyyymmddhhnnss* lub w formacie OPC.

czas_końca

Określnie czasu końca wyświetlanego trendu. Podawany w sposób analogiczny jak dla czasu początku. Jeżeli parametr nie zostanie użyty, to długość okresu trendu wynika z zakresu zapisanego w pliku definicji trendu.

współrzędna_X

Parametr określa współrzędną X okna programu AsTrend (lewego górnego narożnika). Wartość -1 oznacza pobranie odpowiedniej współrzędnej z definicji trendu.

współrzędna_Y

Parametr określa współrzędną Y okna programu AsTrend (lewego górnego narożnika). Wartość -1 oznacza pobranie odpowiedniej współrzędnej z definicji trendu.

monitor

W parametrze należy podać numer monitora, na którym powinno otworzyć się okno programu AsTrend. Wartość 0 oznacza monitor, na którym aktualnie znajduje się kursor myszki, -1 oznacza monitor wynikający z przyjętych współrzędnych x, y.

Przykłady

AstrendDisplay ("tr1.trnx#z1#z2", zmienna_startu, "", "", "")

Akcja wyświetla trend *tr1.trnx*. Do zmiennych zadeklarowanych w pliku trendu zostaną dodane zmienne *z1* i *z2*. Czas początku trendu zostanie ustalony na podstawie aktualnej wartości zmiennej *zmienna_startu*.

**Zobacz też**

- [akcja AsTrendPrint](#)

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 7. Współpraca z programem AsTrend

4.20 AstrendPrint

Przeznaczenie

Akcja służy do wydrukowania wybranego trendu przy pomocy programu AsTrend z wyświetlonym wybranym trendem. W akcji można podać nazwę wzorca trendu i czas początku trendu. Opcjonalnie można podać nazwy zmiennych wyświetlonych w trendzie.

Składnia

AstrendPrint (*nazwa_trendu*, *czas_początku*, *tytuł*, *nagłówek*, *stopka*)

AstrendPrint (*nazwa_trendu*, *czas_początku*, *czas_końca*, *tytuł*, *nagłówek*, *stopka*)

Parametry

nazwa_trendu

Nazwa pliku z definicją trendu. Wpisanie znaku "?" powoduje otwarcie okna, w którym użytkownik może wybrać trend do wyświetlenia.

Po nazwie pliku trendu można podać nazwy zmiennych, które zostaną dodane do wykresu zmiennych zadeklarowanych w definicji trendu. Jeżeli nazwy zmiennych będą rozdzielone znakami #, to zmienne zostaną dodane do trendu. Jeżeli nazwy zmiennych będą rozdzielone znakami <, to zmienne zastąpią zmienne znajdujące się w definicji trendu.

czas_początku

Określenie czasu początku wyświetlanego trendu. Parametr może być wartością typu *DateTime*, czasem bezwzględnym w postaci *yyyymmddhhnnss*, czasem w formacie OPC lub nazwą zmiennej procesowej. W sytuacji, gdy parametrem jest nazwą zmiennej, to jej wartość interpretowana jest jako ilość sekund od 1.1.1970 dla zmiennych o wartościach liczbowych, a dla zmiennych tekstowych jako czas w postaci *yyyymmddhhnnss* lub w formacie OPC.

czas_końca

Określenie czasu końca wyświetlanego trendu. Podawany w sposób analogiczny jak dla czasu początku. Jeżeli parametr nie zostanie użyty, to długość okresu trendu wynika z zakresu zapisanego w pliku definicji trendu.

tytuł

Parametr określa nowy tytuł wydruku, zastępujący tytuł zdefiniowany w pliku trendu.

nagłówek

Parametr określa nowy nagłówek wydruku, zastępujący nagłówek zdefiniowany w pliku trendu.

stopka

Parametr określa nowy tekst stopki wydruku, zastępujący stopkę zdefiniowaną w pliku trendu.

Przykłady

```
AstrendPrint (tr1.trnx, "NOW-1H" ,"" ,"" ,"" )
```

Akcja drukuje trend *tr1.trnx* z domyślnymi ustawieniami tytułu, nagłówka i stopki. Drukowany jest trend z ostatniej godziny.

**Zobacz też**

- [akcja *AsTrendDisplay*](#)
- *Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 7. Współpraca z programem AsTrend*

4.21 Break

Przeznaczenie

Akcja pozwala na warunkowe przerywanie wykonywania akcji złożonej. Gdy testowane wyrażenie jest równe *true*, to akcja zostanie przerwana. Akcję można też używać jako jeden z elementów akcji *Actions*.

Składnia

Break (*warunek*)

Parametry***warunek***

Warunek logiczny, którego wartość decyduje o tym, czy należy przerwać wykonywanie akcji złożonej.

Przykłady

Break (Variable(v3)==0)

SetVariable (v2, 0)

SetVariable (v1, 0)

Operacje sterujące zostaną wykonane tylko wtedy, gdy aktualna wartość zmiennej v3 będzie różna od 0.



Zobacz też

[Akcje złożone](#), [akcja Perform](#), [akcja Actions](#), [akcja Ask](#)

4.22 CancelControls

Przeznaczenie

Akcja pozwala na anulowanie wybranych oczekujących operacji sterujących.

Niektóre obiekty na diagramach synoptycznych mogą pracować w trybie opóźnionego sterowania. Akcja *CancelControls* powoduje, że te obiekty wrócą do trybu normalnego działania, a operacje sterujące nie zostaną wykonane.

Składnia

CancelControls(*nazwa_obiektu*)

Parametry

nazwa_obiektu

Parametr określa nazwę obiektów, dla których sterowania mają zostać anulowane. W nazwie można stosować znaki specjalne wzorców nazw '*' i '?' .

Parametr może być też jedną z poniższych stałych:

\$All – akcja dotyczy wszystkich obiektów we wszystkich oknach i diagramach.

\$AllVisible – akcja dotyczy wszystkich obiektów we wszystkich widocznych oknach. Okno jest widoczne wtedy, gdy nie jest przesłonięte w żaden sposób przez jakiegokolwiek inne okno.

\$Window – akcja dotyczy wszystkich obiektów we wszystkich diagramach bieżącego okna (wynikającego z kontekstu użycia akcji).

\$Diagram – akcja dotyczy wszystkich obiektów bieżącego diagramu (wynikającego z kontekstu użycia akcji).

Przykłady

CancelControls(\$Diagram)

Wszystkie oczekujące operacje sterujące na aktywnym (wynikającym z kontekstu użycia) diagramie zostaną zaniechane.



Zobacz też

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, 6. *Organizacja operacji sterujących*

- [akcja SendControls](#)

4.23 ChangePassword

Przeznaczenie

Akcja powoduje wyświetlenie okna służącego do zmiany hasła aktualnie zalogowanego użytkownika. Hasło można też zmienić w obiekcie typu *Pole autoryzacji* osadzonym na diagramie synoptycznym.

Składnia

ChangePassword()



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 5. *System uprawnień*

- AsixEvo_Objekty.PDF/CHM, obiekt *Pole autoryzacji*

4.24 ClearSelection

Przeznaczenie

Akcja powoduje usunięcie selekcji z aktualnie wybranego obiektu diagramu.

Składnia

ClearSelection()

4.25 CloseWindow

Przeznaczenie

Akcja służy do zamykania otwartych okien synoptycznych. Możliwe jest grupowe zamykanie okien na podstawie zgodności ze wzorcem nazwy. Zamykanie okien może być ograniczone do wybranego monitora.

Składnia

CloseWindow (*nazwa_okna, wszystkie*)
CloseWindow (*nazwa_okna, wszystkie, monitor*)

Parametry

nazwa_okna

Parametr określa nazwę okien, których akcja dotyczy. Można podać nazwę dokładną lub wzorzec nazwy zawierający znaki specjalne * i ?. Stosowane są następujące zasady wyszukiwania okien:

- gdy nazwa okna jest pusta, to zamykane jest tylko okno wynikające z kontekstu użycia – okno z którego została wywołana akcja *CloseWindow*.
- gdy nazwa jest podana, to przeszukiwanie okien wykonywane jest w poniższej kolejności:
 - 1 - okno kontekstowe
 - 2 - okno aktywne
 - 3 - pozostałe otwarte okna
- okna startowe aplikacji mogą być zamknięte tylko wtedy, gdy podana została dokładna nazwa okna.

wszystkie

Gdy parametr ma wartość *false*, to zamykana jest tylko pierwsze okno spełniające warunki określone parametrami *nazwa_okna* i *monitor*. Jeżeli parametr ma wartość *true*, to zamykane są wszystkie okna spełniające warunki.

monitor

Parametr określa, na którym monitorze okna mają zostać zamknięte. Parametr może przyjmować poniższe wartości:

-1 – akcja dotyczy wszystkich monitorów.

0 – zamykane są okna na monitorze, na którym znajduje się kursor myszki.

>0 – numer monitora zgodny z numeracją systemową – akcja dotyczy tylko podanego monitora.

Przykłady

CloseWindow ("A*", true, 0)

Akcja zamyka wszystkie okna, których nazwy zaczynają się od znaku A i znajdują się na monitorze wskazywanym przez położenie kursora myszki.

**Zobacz też**

- [akcja OpenWindow](#), [akcja OpenDiagram](#), [akcja MinimizeWindow](#)

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, 5. Zasady otwierania i zamykania okien synoptycznych

4.26 ConfirmRole

Przeznaczenie

Akcja służy do sprawdzenia, czy użytkownik jest przypisany do sprawdzanej roli. Akcja jest przeznaczona do użycia w akcjach złożonych. W zależności od trybu sprawdzane są uprawnienia aktualnie zalogowanego użytkownika lub wyświetlane jest okno logowania, które wymusza identyfikację użytkownika w momencie wykonania akcji.

Wariant z wymuszoną autoryzacją pozwala wykonać akcję złożoną na prawach innego użytkownika niż aktualny. Akcja nigdy nie powoduje zmiany aktualnego użytkownika.

W przypadku błędnej autentyfikacji lub gdy użytkownik nie posiada wskazanej roli wykonywanie kolejnych akcji w akcji złożonej jest przerywane. Akcję można też używać jako jeden z elementów akcji *Actions*.

Składnia

ConfirmRole(*nazwa_rol*, *tryb_weryfikacji*)

Parametry

nazwa_rol

Nazwa roli, przynależność do której jest weryfikowana.

tryb_weryfikacji

Parametr określający sposób weryfikacji uprawnień użytkownika. Możliwe są następujące warianty:

\$Always - potwierdzenie tożsamości wymagane jest za każdym razem. Zapytanie o identyfikator i hasło nastąpi nawet wtedy, gdy aktualnie zalogowany użytkownik jest przypisany do sprawdzanej roli.

\$Never – potwierdzenie tożsamości nigdy nie jest wykonywane. Akcja sprawdza jedynie, czy bieżący użytkownik jest przypisany do sprawdzanej roli.

\$IfRequired – Potwierdzenie tożsamości wykonane będzie tylko wtedy, gdy aktualnie zalogowany użytkownik nie jest przypisany do danej roli.

Przykłady

```
ConfirmRole ("SuperOperator", $Never)
```

```
SetVariable ( v2, 0)
```

```
SetVariable ( v1, 0)
```

W akcji złożonej operacje sterujące zostaną wykonane tylko wtedy, gdy aktualnie zalogowany użytkownik jest przypisany do roli *SuperOperator*. Okno logowania nie będzie wyświetlone.

```
ConfirmRole ("SuperOperator", $Always)
```

```
SetVariable ( v2, 0)
```

```
SetVariable ( v1, 0)
```

Niezależnie od uprawnień aktualnie zalogowanego użytkownika, nastąpi weryfikacja użytkownika. Żeby dalsza część akcji została wykonana, należy podać identyfikator i hasło użytkownika, który jest przypisany do roli *SuperOperator*.

**Zobacz też**

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 5. System uprawnień
- Asix.Evo_Techniki_budowy_diagramów.PDF/CHM), 6.5. Kontrola uprawnień

4.27 EditPatterns

Przeznaczenie

Uruchamia edytor przebiegów wzorcowych. Edytor krzywych wzorcowych uruchomiony przy pomocy akcji *EditPatterns* pozwala na edycję tylko krzywych lokalnych.

Składnia

EditPatterns()

4.28 EndAlarm

Przeznaczenie

Akcja służy do zgłoszenia zdarzenia zakończenia alarmu. Alarm będący obiektem akcji musi należeć do strategii zewnętrznej lub niezdefiniowanej. Alternatywną metodą zgłoszenia zakończenia alarmu jest użycie w funkcji *Cancel* interfejsu *IAlarm* w kodzie skryptu użytkownika.

Składnia

EndAlarm(*nazwa_domeny*, *identyfikator_alarmu*)

EndAlarm(*identyfikator_alarmu*, *identyfikator_alarmu*, *parametr_1*, ...)

Parametry***nazwa_domeny***

Nazwa domeny alarmów, której dotyczy akcja. Jeżeli nazwa nie jest podana, to akcja dotyczy domeny domyślnej (pierwszej).

identyfikator_alarmu

Identyfikator alarmu, który ma zostać zakończony.

parametr_1, ..., parametr_n

Dodatkowe parametry zgłoszenie rejestrowane w logu alarmów.



Zobacz też

- AsixEvo_System_alarmów.PDF/CHM, *Strategie wykrywania alarmów*
- AsixEvo_Skrypty.PDF/CHM, *Interfejs IAlarm*
- [akcja StartAlarm](#)
- Asix.Evo_System_alarmów.PDF/CHM

4.29 ExcludeAlarm

Przeznaczenie

Akcja służy do wykluczenia alarmu z obsługi. Jeżeli alarm jest wykluczony, to ignorowane są wszystkie kolejne zdarzenia początku i końca alarmu. Alarmy mogą być także wykluczane z wykorzystaniem obiektu *Tabela alarmów aktywnych*.

Składnia

`ExcludeAlarm(nazwa_domeny, identyfikator_alarmu)`

Parametry

nazwa_domeny

Nazwa domeny alarmów, której dotyczy akcja. Jeżeli nazwa jest pusta, to akcja dotyczy domeny domyślnej (pierwszej).

identyfikator_alarmu

Identyfikator alarmu, który ma zostać wykluczony.



Zobacz też

- AsixEvo_System_alarmów.PDF/CHM
- AsixEvo_Objekty.PDF/CHM, obiekt *Tabela alarmów aktywnych*
- [akcja IncludeAlarm](#)

4.30 Execute

Przeznaczenie

Akcja wykonuje inną akcję podaną w parametrze wywołania. Akcja pozwala na wykonywanie akcji definiowanych w bazie definicji zmiennych lub budowanych w zależności od stanu aplikacji. W tym drugim przypadku akcję *Execute* należy stosować tylko wtedy, gdy typ akcji nie jest z góry znany.

Składnia

Execute(*akcja*)

Parametry

akcja

Dynamicznie tworzona treść akcji, którą należy wykonać.

Przykłady

```
Execute( Attribute(Akcja) )
```

W powyższym przykładzie treść akcji pobierana jest z atrybutu *Akcja* zmiennej kontekstowej (zmiennej głównej obiektu, dla którego akcja *Execute* została wykonana w reakcji na jakieś zdarzenie).

```
Execute ( "SetVariable (v1," + Variable(v1)==0?1:0 )
```

W tym przypadku tworzona jest dynamicznie akcja *SetVariable*, o treści zależnej od wartości zmiennej *v1*. Chociaż konstrukcja jest poprawna, to zalecane jest identycznie działająca prostsza forma:

```
SetVariable (v1,Variable(v1)==0?1:0)
```

```
Execute (Variable(v1)==0? "SetVariable (v1,1)":"Nothing()")
```

W zależności od wartości zmiennej *v1* wykonywana jest akcja *SetVariable* lub *Nothing*. Identyczny efekt można uzyskać przy pomocy akcji *Perform*:

```
Perform (Variable(v1)==0, SetVariable (v1,1), Nothing())
```



Zobacz też

[akcja *Nothing*](#), [akcja *Perform*](#)

4.31 GetVarNames

Przeznaczenie

Akcja zapisuje do schowka nazwy zmiennych używanych w obiektach położonych pod aktualną pozycją kursora myszki. Nazwy zmiennych ze schowka mogą być następnie użyte w programie AsTrend jako źródło wyboru zmiennych.

Wykrywane są nazwy zmiennych podane we właściwościach o charakterze nazwy zmiennej oraz zmienne, do których odwołanie nastąpiło poprzez prefiksy właściwości # i &.

Akcja jest typowo używana jako akcja wykonawcza globalnego skrótu klawiszowego.

Składnia

GetVarNames ()

4.32 IncludeAlarm

Przeznaczenie

Akcja służy do ponownego włączenia do obsługi alarmu wcześniej wykluczonego. Alarmy mogą być także włączane od obsługi z wykorzystaniem obiektu *Tabela alarmów aktywnych*.

Składnia

IncludeAlarm(*nazwa_domeny*, *identyfikator_alarmu*)

Parametry

nazwa_domeny

Nazwa domeny alarmów, której dotyczy akcja. Jeżeli nazwa jest pusta, to akcja dotyczy domeny domyślnej (pierwszej).

identyfikator_alarmu

Identyfikator alarmu, który ma zostać włączony do obsługi.



Zobacz też

- AsixEvo_System_alarmów.PDF/CHM

- AsixEvo_Obiekty.PDF/CHM, obiekt *Tabela alarmów aktywnych*
- [akcja ExcludeAlarm](#)

4.33 Login

Przeznaczenie

Akcja powoduje wyświetlenie okna służącego do zalogowania użytkownika. Inne dostępne metody logowania to:

- Okno panelu kontrolnego, które można w każdej aplikacji otworzyć przy pomocy kombinacji klawiszy *Ctrl-Shift-F1*
- W obiekcie typu *Pole autoryzacji* osadzonym na diagramie synoptycznym

Składnia

Login()



Zobacz też

- AsixEvo_Obiekty.PDF/CHM, obiekt *Pole autoryzacji*
- Asix.Evo_Parametryzacja_aplikacji, 5. *System uprawnień*

4.34 Logout

Przeznaczenie

Akcja wylogowuje aktualnego użytkownika. Inne dostępne metody wylogowania to:

- Okno panelu kontrolnego, które można w każdej aplikacji otworzyć przy pomocy kombinacji klawiszy *Ctrl-Shift-F1*
- W obiekcie typu *Pole autoryzacji* osadzonym na diagramie synoptycznym

Składnia

Logout()



Zobacz też

- AsixEvo_Objekty.PDF/CHM, obiekt *Pole autoryzacji*
- Asix.Evo_Parametryzacja_aplikacji, 5. System uprawnień

4.35 Message

Przeznaczenie

Akcja pozwala na zapisanie w dzienniku zdarzeń aplikacji dowolnego komunikatu. Zarejestrowane komunikaty są widoczne w panelu komunikatów i w obiektach typu *Dziennik zdarzeń*.

Składnia

Message(*typ_komunikatu*, *źródło_komunikatu*, *treść_komunikatu*)

Parametry

typ_komunikatu

Parametr określa ważność komunikatu. Możliwe są poniższe wartości:

\$Info – Komunikat informacyjny

\$Error – Komunikat informujący o wystąpieniu błędu

\$Warning – Komunikat ostrzegawczy

źródło_komunikatu

Krótki tekst identyfikujący źródło, które utworzyło komunikat.

treść_komunikatu

Dowolny tekst stanowiący treść komunikatu.

Przykłady

```
Message($Info, "Test", "Akcja wykonana dla użytkownika " + Variable(CurrentUser))
```

Akcja zapisuje do dziennika zdarzeń komunikat informacyjny, w którym przekazuje nazwę aktualnie zalogowanego użytkownika.



Zobacz też

- AsixEvo_Objekty.PDF/CHM, Obiekt *Dziennik zdarzeń*

4.36 MinimizeWindow

Przeznaczenie

Akcja służy do zminimalizowania otwartych okien synoptycznych. Możliwa jest grupowa minimalizacja okien na podstawie zgodności z wzorcem nazwy.

Składnia

MinimizeWindow(*nazwa_okna*)

Parametry

nazwa_okna

Parametr określa nazwę okien, których akcja dotyczy. Można podać nazwę dokładną lub wzorzec nazwy zawierający znaki specjalne * i ?. Gdy nazwa okna jest pusta, to minimalizowane jest tylko okno wynikające z kontekstu użycia – okno z którego została wywołana akcja *MinimizeWindow*.

Przykłady

```
MinimizeWindow("")
```

Akcja minimalizuje okno w kontekście, którego została użyta.



Zobacz też

- [akcja OpenWindow](#), [akcja CloseWindow](#), [akcja OpenDiagram](#)

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, *Zasady otwierania i zamykania okien synoptycznych*

4.37 NextDiagram

Przeznaczenie

Akcja służy od przełączenia diagramu w wybranym panelu okna synoptycznego. Przełączenie bazuje na zapamiętanej historii wcześniej używanych diagramów. Akcja ma sens tylko w przypadku wcześniejszego użycia akcji *PreviousDiagram* dla wybranego panelu. Akcja *NextDiagram* wyświetla następny diagram z historii, odwracając działanie akcji *PreviousDiagram*.

Składnia

PreviousDiagram (nazwa_okna, nazwa_panelu)

Parametry

nazwa_okna

Nazwa okna, którego akcja dotyczy. Może być pusta, a wtedy okno będące obiektem akcji ustalane jest na podstawie kontekstu użycia.

nazwa_panelu

Nazwa panelu, w którym diagram ma zostać przełączony. Może być pusta, a wtedy panel będący obiektem akcji ustalany jest na podstawie kontekstu użycia.



Zobacz też

- akcja [OpenWindow](#), akcja [PreviousDiagram](#)

4.38 Nothing

Przeznaczenie

Akcja nie wykonuje żadnych czynności. Przeznaczona jako pusta alternatywa w konstrukcjach wykorzystujących akcje *Perform* i *Execute*.

Składnia

Nothing()

Przykłady

Perform (HasRole("SuperOperator"), SetVariable (v, 0), Nothing())

Jeżeli zalogowany użytkownik jest przypisany do roli *SuperOperator*, to zostanie wykonana operacja sterująca. Jeżeli nie, to nic nie zostanie wykonane.

Execute(Attribute(Akcja))

W powyższym przykładzie treść akcji pobierana jest z atrybutu *Akcja* zmiennej kontekstowej. Jeżeli dla niektórych zmiennych nie należy wykonywać żadnych operacji, to w bazie zmiennych trzeba użyć akcji *Nothing*.



Zobacz też

[akcja Execute](#), [akcja Perform](#)

4.39 ObjectAction

Przeznaczenie

Akcja pozwala na wykonanie operacji na wybranych obiektach diagramu. Rodzaje dostępnych operacji zależą od typu wybranych obiektów.

Składnia

ObjectAction (nazwa_obiektu, rodzaj_operacji, parametr1, parametr2, ...)

Parametry

nazwa_obiektu

Nazwa obiektów, na których należy wykonać żądaną operację. W parametrze można używać znaków '*' i '?'.

W przypadku użycia akcji w kontekście obiektu będącego częścią wzorca wyszukiwanie obiektów zgodnych z podaną nazwą dotyczy standardowo tylko obiektów tego wzorca. Odwołanie do obiektu nie będącego częścią wzorca wymaga dodania prefiksu '!' do nazwy obiektu.

rodzaj_operacji

Stała określająca rodzaj operacji wykonywanej na wskazanych obiektach. Dozwolony zestaw operacji zależy od typu obiektów.

Poniższa tabela opisuje dostępne rodzaje operacji.

Przeglądarka	
\$Reload	Przeładowanie strony. Operacja nie ma dodatkowych parametrów.
\$Print	Otwarcie okienka wydruku strony. Operacja nie ma dodatkowych parametrów.
Tabela Asbase	
\$First	Przejdźcie do pierwszego wiersza. Operacja nie ma dodatkowych parametrów.
\$Last	Przejdźcie do ostatniego wiersza. Operacja nie ma dodatkowych parametrów.
\$Next	Przejdźcie do następnego wiersza. Operacja nie ma dodatkowych parametrów.
\$NextPage	Przejdźcie do następnej strony. Operacja nie ma dodatkowych parametrów.
\$Previous	Przejdźcie do poprzedniego wiersza. Operacja nie ma dodatkowych parametrów.
\$PreviousPage	Przejdźcie do poprzedniej strony. Operacja nie ma dodatkowych parametrów.
\$Download	Załadowanie wybranej receptury do zestawu zmiennych określonego we właściwości <i>Identyfikator zestawu zmiennych ładowań</i> obiektu. Operacja nie ma dodatkowych parametrów. Operacja dozwolona tylko dla receptur.
\$Reload	Ponowne odczytanie rekordów z bazy danych. Operacja nie ma dodatkowych parametrów.

parametr1, ..., parametrN

Wartości parametrów operacji wykonywanej na obiektach. Liczba i postać parametrów użytych w wywołaniu musi być zgodna z wybranym rodzajem operacji.

4.40 OpenDiagram

Przeznaczenie

Akcja pozwala na otwarcie diagramu w dynamicznie utworzonym nowym oknie. Jeżeli diagram o podanej nazwie i identycznym zestawie parametrów jest już otwarty, to nastąpi jedynie aktywizacja wcześniej utworzonego okna.

Składnia

OpenDiagram (*nazwa_diagramu*, *parametry*, *tytuł*, *tryb_otwarcia*, *współrzędna_X*, *współrzędna_Y*, *atrybuty*, *tryb_wymiany*)

Parametry

nazwa_diagramu

Parametr określa nazwę diagramu, który ma zostać wyświetlony.

parametry

Parametr służy do przekazania do otwieranego diagramu wartości jego parametrów. Zestaw dozwolonych parametrów zależy od definicji diagramu. Parametry są podawane jako sekwencja par *nazwa_parametru=wartość_parametru* rozdzielonych znakiem średnika, np.

Kolor=Red;Zmienna=v1

tytuł

Parametr określa tekst, który ma zostać wyświetlony na belce tytułowej okna.

tryb_otwarcia

Parametr określa sposób zachowania się okna po otwarciu. Możliwe wartości to jedna z poniższych stałych:

\$Normal – okno zwykłe

\$Dialog – okno dialogowe, nie będzie możliwości przejścia do innych otwartych okien aplikacji dopóki okno dialogowe nie zostanie zamknięte

\$Temp – okno tymczasowe, w momencie przejścia do dowolnego innego okna nastąpi automatyczne zamknięcie okna otwartego w trybie tymczasowym

współrzędna_X

Parametr określa współrzędną X tworzonego okna (lewego górnego narożnika).
Parametr może być bezpośrednią współrzędną liczbową lub jedną z poniższych stałych:

&CursorLocation – współrzędną X określa aktualne położenie kursora myszki.
Współrzędna może zostać zmodyfikowana, aby zapobiec częściowemu wyjściu otwieranego okna poza obszar monitora.

&ActiveLocation – współrzędną X otwieranego jest identyczna jak współrzędna X okna, które było aktywne w momencie wykonania akcji.

współrzędna_Y

Parametr określa współrzędną Y tworzonego okna (lewego górnego narożnika).
Zasady podawania parametru są identyczne jak dla parametru *współrzędna_X*.

atrybuty

Parametr określa wygląd tworzonego okna. Wartością parametru jest łańcuch tekstowy złożony z poniższych nazw atrybutów rozdzielonych znakiem przecinka.

NoFrame – okno nie będzie posiadało ramki.

NoTitleBar – okno nie będzie posiadało belki tytułowej.

NoButtons – na belce tytułowej okna nie będą wyświetlane przyciski systemowe.

NoMinButton – na belce tytułowej nie będzie wyświetlany przycisk minimalizacji.

NoMaxButton – na belce tytułowej nie będzie wyświetlany przycisk maksymalizacji.

FixedSize – nie będzie można zmienić rozmiaru okna.

ToolWindow – okno będzie wyświetlone w stylu systemowego okna narzędziowego (m.in. mniejsza wysokość belki tytułowej).

TopMost – okno będzie wyświetlane w trybie „zawsze na wierzchu”, czyli zawsze będzie przykrywało okna zwykłe.

NoTaskBar – okno nie będzie pokazywane na systemowym pasku zadań.

Maximized – okno zostanie wyświetlone na całym obszarze monitora wynikającego ze współrzędnych x i y.

Center – okno zostanie wyśrodkowane na monitorze wynikającym ze współrzędnych x i y.

KeepAspectRatio – zapewnia zachowanie proporcji okna przy zmianie jego wielkości przez użytkownika.

FitToScreen – zapewnia dopasowanie okna do wielkości ekranu. Jeżeli okno jest większe od ekranu, to zostanie odpowiednio zmniejszone. Podobnie, jeżeli wystaje poza krawędź ekranu, to zostanie przesunięte. Opcja jest przydatna, gdy aplikacja jest uruchamiana na komputerach o różnych rozmiarach ekranów.

tryb_wymiany

Parametr określa jakie dodatkowe akcje są podejmowane po otwarciu okna. Możliwe wartości to jedna z poniższych stałych:

\$None – żadne dodatkowe operacje nie są wykonywane.

\$Closable – zamknięte zostaną wszystkie otwarte okna, które nie posiadają blokady zamykania i znajdują się na monitorze nowo otwartego okna.

\$ExceptCurrent – zamknięte zostaną wszystkie otwarte okna, które nie posiadają blokady zamykania i znajdują się na monitorze nowo otwartego okna za wyjątkiem okna aktywnego w momencie wykonania akcji.

\$CurrentOnly – zamknięte zostanie tylko okno aktywne w momencie wykonania akcji.

Przykłady

```
OpenDiagram ( diag1,"pokoj=11", null, $Dialog, $CursorLocation, $CursorLocation,
"NoTitleBar,FixedSize", $None )
```

Akcja otwiera w nowym oknie diagram *diag1*, przekazując mu wartość parametru *pokoj*. Okno jest typu dialogowego, a pozycja wynika z aktualnego położenia kursora myszki. Belka tytułowa nie jest pokazywana, a okno nie można zmienić swojego rozmiaru. Nie są zamykane żadne inne okna aplikacji.



Zobacz też

- [akcja *OpenWindow*](#), [akcja *CloseWindow*](#), [akcja *MinimizeWindow*](#)

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, 5. *Zasady otwierania i zamykania okien synoptycznych*

4.41 OpenWindow

Przeznaczenie

Akcja służy do otwierania predefiniowanych okien synoptycznych lub do wymiany diagramów wyświetlanych w panelach okna. Jest to podstawowa akcja służąca do stworzenia schematu nawigacji w aplikacji.

Składnia

OpenWindow (*nazwa_okna, nazwa_panelu, nazwa_diagramu, parametry, nowe_okno, współrzędna_X, współrzędna_Y, tryb_wymiany*)

OpenWindow (*nazwa_okna, nazwa_panelu, nazwa_diagramu, parametry, nowe_okno*)

OpenWindow (*nazwa_okna, nazwa_panelu, nazwa_diagramu, parametry*)

Parametry

nazwa_okna

Nazwa okna, którego akcja dotyczy. Może być pusta, a wtedy okno będące obiektem akcji ustalane jest na podstawie kontekstu użycia. Szczegóły użycia w sekcji *Algorytm*.

nazwa_panelu

Nazwa panelu, w którym diagram ma zostać otwarty. Może być pusta, a wtedy panel będący obiektem akcji ustalany jest na podstawie kontekstu użycia. Szczegóły użycia w sekcji *Algorytm*.

nazwa_diagramu

Nazwa diagramu, którego akcja dotyczy. Może być pusta, a wtedy akcja powoduje jedynie otwarcie okna bez zmiany diagramu lub z otwarciem domyślnych diagramów paneli. Szczegóły użycia w sekcji *Algorytm*.

parametry

Parametr służy do przekazania do otwieranego diagramu wartości jego parametrów. Zestaw dozwolonych parametrów zależy od definicji diagramu. Parametry są podawane jako sekwencja par *nazwa_parametru=wartość_parametru* rozdzielonych znakiem średnika, np.

Kolor=Red;Zmienna=v1

Jeżeli w oknie będącym obiektem akcji jest używane menu, to parametry akcji są także przekazywane do tego menu.

nowe_okno

Parametr typu logicznego *true/false*, który określa, czy należy utworzyć nowe okno. Szczegóły użycia w sekcji *Algorytm*.

współrzędna_X

Parametr określa współrzędną X otwieranego okna (lewego górnego narożnika). Parametr może być bezpośrednią współrzędną liczbową lub jedną z poniższych stałych:

\$Default – współrzędna X wynika z definicji okna lub (dla okna już otwartego) jest to bieżąca pozycja. Jest to wartość domyślna w przypadku użycia wariantu akcji bez parametru *współrzędna_X*.

\$CursorLocation – współrzędną X określa aktualne położenie kursora myszki. Współrzędna może zostać zmodyfikowana, aby zapobiec częściowemu wyjściu otwieranego okna poza obszar monitora.

\$ActiveLocation – współrzędna X otwieranego okna jest identyczna jak współrzędna X okna, które było aktywne w momencie wykonania akcji.

współrzędna_Y

Parametr określa współrzędną Y otwieranego okna (lewego górnego narożnika). Zasady podawania parametru są identyczne jak dla parametru *współrzędna_X*.

tryb_wymiany

Parametr określa dodatkowe akcje, które są podejmowane po otwarciu okna. Możliwe wartości to jedna z poniższych stałych:

\$None – żadne dodatkowe operacje nie są wykonywane (wartość domyślna).

\$Closable – zamknięte zostaną wszystkie otwarte okna, które nie posiadają blokady zamykania i znajdują się na monitorze otwieranego okna.

\$ExceptCurrent – zamknięte zostaną wszystkie otwarte okna, które nie posiadają blokady zamykania i znajdują się na monitorze otwieranego okna za wyjątkiem okna aktywnego w momencie wykonania akcji.

\$CurrentOnly – zamknięte zostanie tylko okno aktywne w momencie wykonania akcji.

\$ReplaceCurrent – zamknięte zostanie tylko okno aktywne w momencie wykonania akcji. Współrzędne i rozmiar okna otwieranego zostaną ustawione identycznie jak dla okna zamykanego (parametry *współrzędna_X* i *współrzędna_Y* są ignorowane).

Algorytm

1. Jeżeli nazwa diagramu nie jest podana:
 - Jeżeli nazwa okna też nie została podana, to obiektem akcji jest okno wynikające z kontekstu użycia lub okno bieżące.
 - Jeżeli nazwa okna została podana i parametr *nowe_okno* jest równy *true*, to otwierane jest nowe okno o podanej nazwie z domyślnymi diagramami.
 - Jeżeli nazwa okna została podana i parametr *nowe_okno* jest równy *false*, to wyszukiwane jest wśród otwartych okien okno o podanej nazwie (zaczynając od okna aktywnego). Jeżeli takie okno zostanie znalezione, to jest ono obiektem akcji. Jeżeli nie, to otwierane jest nowe okno o podanej nazwie z domyślnymi diagramami.
2. Jeżeli nazwa diagramu została podana:
 - Jeżeli nazwa okna nie została podana, to w oknie kontekstowym (wynikającym z miejsca użycia akcji *OpenWindow*) i w podanym panelu (lub w panelu kontekstowym) otwierany jest podany diagram.
 - Jeżeli nazwa okna została podana, to sprawdzane jest, czy wśród otwartych okien o podanej nazwie istnieje okno, które ma już otwarty diagram określony w wywołaniu akcji wraz z identycznym zestawem parametrów (w kolejności wyszukiwania: okno kontekstowe, okno bieżące, pozostałe okna). Jeżeli takie okno zostanie znalezione, to zostanie ono oknem aktywnym.
 - Jeżeli w poprzednim punkcie właściwego okna nie znaleziono, a parametr *nowe_okno* jest równy *false*, to wyszukiwane jest wśród otwartych okien okno o podanej nazwie (w kolejności: okno kontekstowe, okno bieżące, pozostałe okna). Jeżeli takie okno zostanie znalezione, to w panelu podanym w treści akcji otwierany jest podany diagram. Jeżeli okna nie znaleziono lub parametr *nowe_okno* jest równy *true*, to otwierane jest nowe okno o podanej nazwie. W panelu podanym w treści akcji otwierany jest podany diagram, a w pozostałych panelach otwierane są diagramy domyślne.
3. W każdym przypadku, jeżeli udało się określić okno będące obiektem akcji, to okno to staje się oknem aktywnym, wykonywana jest w razie konieczności odpowiednia korekta pozycji okna i zamykane są okna wynikające z parametru *tryb_wymiany*.

Przykłady

OpenWindow (start, schemat, silnik, zmienna=s1)

Jeżeli okno o nazwie *start* jest już otwarte, to w panelu *schemat* zostanie otwarty diagram *silnik* z parametrami *zmienna=s1*. Jeżeli taki diagram z identycznymi parametrami był już otwarty, to nastąpi jedynie aktywizacja okna.

Jeżeli okno o nazwie *start* nie jest jeszcze otwarte, to zostanie teraz otworzone. W panelu *schemat* zostanie otwarty diagram *silnik* z parametrami *zmienna=s1*. W pozostałych panelach zostaną otwarte diagramy domyślne.

OpenWindow (null, schemat, silnik, zmienna=s1, false, \$Default, \$Default, \$Closable)

W oknie kontekstowym, w panelu *schemat* zostanie otwarty diagram *silnik* z parametrami *zmienna=s1*. Pozycja okna nie zostanie zmieniona. Dodatkowo zostaną zamknięte wszystkie okna bez blokady zamknięcia.



Zobacz też

- [akcja OpenDiagram](#), [akcja CloseWindow](#), [akcja MinimizeWindow](#)

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, 5. *Zasady otwierania i zamykania okien synoptycznych*

4.42 Perform

Przeznaczenie

Akcja pozwala na sprawdzenie warunku logicznego i w zależności od jego wartości wykonanie jednej z dwóch akcji. Akcja *Perform* w niektórych przypadkach może zastąpić akcje złożone z konstrukcją *Break*.

Składnia

Perform(warunek, akcja_gdy_true, akcja_gdy_false)

Parametry

warunek

Wyrażenie warunkowe, którego wartość decyduje, która z akcji składowych zostanie wykonana.

akcja_gdy_true

Akcja, która zostanie wykonana, gdy parametr *warunek* będzie miał wartość *true*.

akcja_gdy_false

Akcja, która zostanie wykonana, gdy parametr *warunek* będzie miał wartość *false*.

Przykłady

```
Perform ( Variable(AktywnaStrefa)==1, SetVariable(Strefa1,0), PlaySound(NieWolno) )
```

Jeżeli wartość zmiennej *AktywnaStrefa* jest równa 1, to zostanie ustawiona zmienna *Strefa1* na 0. W przeciwnym wypadku zostanie odtworzony dźwięk o nazwie *NieWolno*.



Zobacz też

- [Akcje złożone](#), [akcja Break](#)

4.43 PlaySound

Przeznaczenie

Akcja odtwarza jednokrotnie wybrany sygnał dźwiękowy. Sygnał dźwiękowy musi być wcześniej dodany do puli dźwięków aplikacji.

Składnia

PlaySound(*nazwa_dźwięku*)

Parametry

nazwa_dźwięku

Nazwa sygnału dźwiękowego, który należy odtworzyć.

Przykłady

PlaySound(alert)

Akcja odtwarza jednokrotnie plik dźwiękowy, który został zadeklarowany w puli dźwięków pod nazwą *alert*.

**Zobacz też**

- AsixEvo_System_alarmów.PDF/CHM, *Sygnalizacja dźwiękowa*

- [akcja PlaySoundLooping](#), [akcja StopSound](#)

4.44 PlaySoundLooping**Przeznaczenie**

Akcja odtwarza wielokrotnie wybrany sygnał dźwiękowy. Odtwarzanie jest powtarzane do momentu wykonania akcji *StopSound* lub do rozpoczęcia odtwarzania innego sygnału dźwiękowego.

Sygnał dźwiękowy musi być wcześniej dodany do puli dźwięków aplikacji.

Składnia

PlaySoundLooping(*nazwa_dźwięku*)

Parametry

nazwa_dźwięku

Nazwa sygnału dźwiękowego, który należy odtworzyć.

Przykłady

PlaySoundLooping(alert)

Akcja odtwarza wielokrotnie plik dźwiękowy, który został zadeklarowany w puli dźwięków pod nazwą *alert*.

**Zobacz też**

- AsixEvo_System_alarmów.PDF/CHM, *Sygnalizacja dźwiękowa*

- [akcja PlaySound](#), [akcja StopSound](#)

4.45 PreviousDiagram

Przeznaczenie

Akcja służy od przełączenia diagramu w wybranym panelu okna synoptycznego. Przełączenie bazuje na zapamiętanej historii wcześniej używanych diagramów. Wybierany jest poprzednio wyświetlany diagram. Kolejne wykonania akcji powodują coraz głębsze cofanie się w historii diagramów.

Składnia

PreviousDiagram (nazwa_okna, nazwa_panelu)

Parametry

nazwa_okna

Nazwa okna, którego akcja dotyczy. Może być pusta, a wtedy okno będące obiektem akcji ustalane jest na podstawie kontekstu użycia.

nazwa_panelu

Nazwa panelu, w którym diagram ma zostać przełączony. Może być pusta, a wtedy panel będący obiektem akcji ustalany jest na podstawie kontekstu użycia.



Zobacz też

- akcja [OpenWindow](#), akcja [NextDiagram](#)

4.46 PrintScreen

Przeznaczenie

Akcja rejestruje obraz wybranego obszaru ekranu. Możliwa jest rejestracja aktywnego okna lub diagramu. Można też zarejestrować kompletny obraz ekranu monitora. Zarejestrowany obraz jest wstawiany do schowka systemowego, drukowany lub zapisywany w pliku. W przypadku zapisu do pliku obraz jest zapisywany w formacie *png* w podkatalogu *Screenshots* katalogu roboczego aplikacji. Nazwa pliku jest tworzona automatycznie na podstawie aktualnej godziny i rodzaju rejestrowanego obrazu.

Składnia

PrintScreen (obszar, tryb_wydruku)

PrintScreen (obszar, tryb_wydruku, tytuł_wydruku)

Parametry

obszar

Parametr określa obszar ekranu, który należy zarejestrować. Możliwe jest użycie poniższych wartości:

\$Diagram – zarejestrowany zostanie obszar aktywnego diagramu.

\$Screen – zarejestrowany zostanie obszar całego ekranu monitora, na którym znajduje się kursor myszki.

\$Window – zarejestrowany zostanie obszar aktywnego okna.

nazwa_obiektu – nazwa obiektu umieszczonego na aktywnym diagramie. Zarejestrowany zostanie tylko obszar tego obiektu.

tryb_wydruku

Parametr określa sposób wykonania akcji. Możliwe jest użycie poniższych stałych:

\$File – obraz jest zapisywany w pliku.

\$Clipboard – obraz jest umieszczany w schowku systemowym.

\$Printer – obraz jest drukowany na drukarce. Używana jest systemowa drukarka domyślna. W ustawieniach stanowiska można zdefiniować dodatkowe opcje wydruku.

tytuł_wydruku

Parametr określa tytuł umieszczany w nagłówku wydruku.

Przykłady

```
PrintScreen($Window, $File)
```

Obszar aktywnego okna zostanie zarejestrowany w pliku umieszczonym w podkatalogu *Screenshots* katalogu roboczego aplikacji.

4.47 RefreshVariable

Przeznaczenie

Akcja pozwala na wymuszenie odczytu wartości zmiennej poza okresem jej planowanego cyklicznego odświeżania wynikającego z definicji.

Składnia

RefreshVariable(*nazwa zmiennej*)

Parametry

nazwa_zmiennej

Nazwa zmiennej procesowej, której wartość należy odświeżyć. Pusta nazwa oznacza użycie zmiennej głównej obiektu wynikającego z kontekstu użycia akcji. Możliwa jest notacja sufiksowa nazwy zmiennej względem zmiennej głównej obiektu kontekstowego.

Przykłady

RefreshVariable ("")

Akcja wymusza odświeżenie wartości zmiennej głównej obiektu na konto, którego akcja jest wykonywana.

4.48 ResetGlobalProperty

Przeznaczenie

Akcja ustawia wartość właściwości globalnej zgodnie z jej oryginalną definicją. Usuwa w ten sposób skutki wcześniejszego użycia akcji SetGlobalProperty.

Składnia

ResetGlobalProperty(*nazwa_właściwości*)

Parametry

nazwa_właściwości

Nazwa właściwości globalnej, której wartość należy zresetować.

4.49 Run

Przeznaczenie

Akcja służy do uruchomienia zewnętrznego programu. Można uruchamiać program bezpośrednio podany w parametrach lub wykorzystać systemowe kojarzenie rozszerzeń nazw plików z programami.

Składnia

Run(*nazwa_pliku*)

Run(*nazwa_pliku*, *argumenty*)

Parametry

nazwa_pliku

Parametr określa program, który należy uruchomić. Może to być nazwa pliku typu exe lub nazwa pliku z danymi. W drugim przypadku zostanie uruchomiony program skojarzony w systemie operacyjnym z typem pliku danych.

W poszukiwaniu programu typu exe uwzględniana jest systemowa ścieżka programów.

argumenty

Tekstowy łańcuch argumentów przekazywany do uruchamianego programu. Format i znaczenie parametru jest specyficzne dla programu.

Przykłady

```
Run("plikipdf\plan.pdf")
```

Zostanie uruchomiona zadeklarowana w systemie przeglądarka plików pdf oraz automatycznie otwarty plik *plan.pdf* z podkatalogu *plikipdf* katalogu aplikacji.

4.50 Script

Przeznaczenie

Akcja służy do uruchomienia skryptu użytkownika.

Składnia

Script(*nazwa* , *parametr1*, *parametr2*, ...)

Parametry

nazwa

Nazwa skryptu użytkownika.

parametr1*, ..., *parametrN

Wartości parametrów skryptu. Liczba i postać parametrów użytych w wywołaniu musi być zgodna z wymaganiami skryptu.

Przykłady

Script(ObliczZuzycie, v1)

Akcja uruchamia skrypt o nazwie *ObliczZuzycie*, przekazując do niego pojedynczy parametr o treści *v1*.



Zobacz też

- AsixEvo_Skrypty.PDF/CHM

4.51 ScriptClose

Przeznaczenie

Akcja służy do zakończenia pracy uruchomionego skryptu użytkownika. Skrypt, który należy zakończyć jest identyfikowany przez nazwę oraz zestaw parametrów z jakimi został uruchomiony. Akcja kończy pracę wszystkich skryptów pasujących do parametrów wywołania.

Składnia

ScriptClose(*nazwa* , *parametr1*, *parametr2*, ...)

Parametry

nazwa

Nazwa skryptu użytkownika.

parametr1, ..., parametrN

Wartości parametrów skryptu. Liczba i postać parametrów użytych w wywołaniu. Musi być zgodna z parametrami użytymi w momencie uruchomienia skryptu.

Przykłady

```
Perform(IsScriptWorking(S1),ScriptClose(S1),Nothing())
```

Akcja kończy pracę skryptu o nazwie S1 uruchomionego wcześniej bez podania parametrów. Dodatkowo pokazane jest użycie funkcji *IsScriptWorking*, która pozwala sprawdzić, czy taki skrypt jest uruchomiony.



Zobacz też

- AsixEvo_Skrypty.PDF/CHM

4.52 ScriptRestart

Przeznaczenie

Akcja służy do restartu (zakończenia i ponownego uruchomienia) pracy uruchomionego wcześniej skryptu użytkownika. Skrypt, który należy zrestartować jest identyfikowany przez nazwę oraz zestaw parametrów z jakimi został uruchomiony. Akcja dotyczy wszystkich skryptów pasujących do parametrów wywołania.

Składnia

ScriptRestart(*nazwa* , *parametr1* , *parametr2* , ...)

Parametry

nazwa

Nazwa skryptu użytkownika.

parametr1, ..., parametrN

Wartości parametrów skryptu. Liczba i postać parametrów użytych w wywołaniu. Musi być zgodna z parametrami użytymi w momencie uruchomienia skryptu.



Zobacz też

- AsixEvo_Skrypty.PDF/CHM

4.53 SecurityBrowser

Przeznaczenie

Akcja powoduje otwarcie okna dialogowego służącego do przeglądania logów gromadzonych przez system zabezpieczeń. Użycie akcji ma sens tylko wtedy, gdy system zabezpieczeń pracuje w trybie centralnym. Dostęp do logów systemu zabezpieczeń może być ograniczony w zależności od uprawnień użytkownika oraz posiadanej licencji systemu Asix.

Składnia

SecurityBrowser(*typ_logu*)

SecurityBrowser(*typ_logu*, *filtr*)

Parametry

typ_logu

Parametr określa typ logu, którego zawartość należy pokazać w momencie otwarcia okna. Użytkownik będzie mógł przełączyć się interaktywnie na inne typy logów.

Parametr może być też jedną z poniższych stałych:

\$Messages – przełączenie na log komunikatów systemu zabezpieczeń

\$Controls – przełączenie na log wykonanych operacji sterujących

\$Actions – przełączenie na log działań wykonanych przez operatora

\$Notes – przełączenie na log notatek operatora

filtr

Parametr określa filtr wyświetlanych danych. Dla logu operacji sterujących jest to nazwa zmiennej, a dla logu notatek identyfikator segmentu. W treści parametru można używać znaków specjalnych * i ?.

W przypadku logu działań operatora należy podać rodzaj operacji. Dopuszczalne rodzaje operacji operatora to:

OpenDiagram – otwarcie diagramu

CloseDiagram – zamknięcie diagramu

OpenTrend – otwarcie trendu

CloseTrend – zamknięcie trendu

RunAsReport – wykonanie akcji AsreportDisplay

RunFile – wykonanie akcji Run

ChangeMode – zmiana trybu pracy



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 5. System uprawnień

4.54 SecurityManager

Przeznaczenie

Akcja otwiera okno dialogowe służące do zarządzania uprawnieniami użytkowników w trakcie wykonywania aplikacji.

Składnia

SecurityManager()



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 5. System uprawnień

4.55 SendControls

Przeznaczenie

Niektóre obiekty na diagramach synoptycznych mogą pracować w trybie opóźnionego sterowania. Akcja *SendControls* powoduje, że te obiekty wykonają oczekujące operacje sterujące.

Składnia

SendControls(*nazwa_obiektu*)

Parametry

nazwa_obiektu

Parametr określa nazwę obiektów, dla których sterowania mają zostać wykonane. W nazwie można stosować znaki specjalne wzorców nazw '*' i '?' .

Parametr może być też jedną z poniższych stałych:

\$All – akcja dotyczy wszystkich obiektów we wszystkich oknach i diagramach.

\$AllVisible – akcja dotyczy wszystkich obiektów we wszystkich widocznych oknach. Okno jest widoczne wtedy, gdy nie jest przesłonięte w żaden sposób przez jakiegokolwiek inne okno.

\$Window – akcja dotyczy wszystkich obiektów we wszystkich diagramach bieżącego okna (wynikającego z kontekstu użycia akcji).

\$Diagram – akcja dotyczy wszystkich obiektów bieżącego diagramu (wynikającego z kontekstu użycia akcji).

Przykłady

SendControls(\$Window)

Wszystkie oczekujące operacje sterujące na aktywnym (wynikającym z kontekstu użycia) oknie zostaną wykonane.



Zobacz też

- Asix.Evo – Techniki budowy diagramów.PDF/CHM, *Organizacja operacji sterujących*

- [akcja CancelControls](#)

4.56 SetBounds

Przeznaczenie

Akcja pozwala na zmianę położenia i rozmiaru wybranego elementu diagramu. Służy do realizacji animacji ruchu i rozmiaru obiektów na diagramie. Dodatkowe informacje i przykład użycia znajdują się w opisie akcji *SetPosition*.

Składnia

SetBounds (nazwa_obiektu, współrzędna_X, współrzędna_Y, szerokość, wysokość)

SetBounds (współrzędna_X, współrzędna_Y, szerokość, wysokość)

Parametry

nazwa_obiektu

Nazwa elementu, którego położenie i rozmiar należy zmienić. Wariant akcji bez podania nazwy dotyczy zmiany położenia i rozmiaru elementu kontekstowego (obiektu, wzorca lub grupy). W parametrze można używać znaków '*' i '?'.

W przypadku użycia akcji w kontekście obiektu będącego częścią wzorca wyszukiwanie obiektów zgodnych z podaną nazwą dotyczy standardowo tylko obiektów tego wzorca. Odwołanie do obiektu nie będącego częścią wzorca wymaga dodania prefiksu '!' do nazwy obiektu.

współrzędna_X

Nowa wartość współrzędnej X kontrolowanego elementu w jednostkach bezwzględnych.

współrzędna_Y

Nowa wartość współrzędnej Y kontrolowanego elementu w jednostkach bezwzględnych.

szerokość

Nowa szerokość kontrolowanego elementu w jednostkach bezwzględnych.

wysokość

Nowa wysokość kontrolowanego elementu w jednostkach bezwzględnych.



Zobacz też

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, *Animacja ruchu i zmiana rozmiaru obiektów*

- akcje [SetPosition](#), akcja [SetSize](#)

4.57 SetGlobalProperty

Przeznaczenie

Akcja pozwala na zmianę wartości właściwości globalnej. Zmiana dotyczy tylko lokalnego stanowiska. Zmiana zostanie zapamiętana - w przypadku restartu aplikacji właściwość będzie w dalszym ciągu posiadała zmienioną wartość. Powrót do oryginalnej definicji wymaga użycia akcji *ResetGlobalProperty*.

Składnia

SetGlobalProperty(*nazwa_właściwości*, *wartość*)

Parametry

nazwa_właściwości

Nazwa właściwości globalnej, której wartość należy zmienić.

wartość

Nowa wartość właściwości globalnej.

4.58 SetProperty

Przeznaczenie

Akcja pozwala na zmianę wartości wskazanego parametru diagramu kontekstowego.

Składnia

SetProperty (*nazwa_parametru*, *wartość_parametru*)

Parametry

nazwa_parametru

Nazwa parametru diagramu, którego wartość należy zmienić.

wartość_parametru

Nowa wartość parametru.

4.59 SetProperty

Przeznaczenie

Akcja pozwala na zmianę wartości parametrów diagramu kontekstowego.

Składnia

SetProperty (*parametry*)

Parametry

parametry

Parametr określa nowe wartości parametrów. Parametry są podawane jako sekwencja par *nazwa_parametru*=*wartość_parametru* rozdzielonych znakiem średnika, np.

Kolor=Red;Zmienna=v1

4.60 SetPosition

Przeznaczenie

Akcja pozwala na zmianę położenia i rozmiaru wybranego elementu diagramu. Służy do realizacji animacji ruchu obiektów na diagramie. Akcja jest typowo używana w obsłudze zdarzenia *Animacja* obiektu, wzorca lub grupy. Parametry położenia przekazywane do funkcji muszą być wyrażone we współrzędnych bezwzględnych (od 0 do 1 000 000, współrzędna bezwzględna 1 000 000 odpowiada prawej krawędzi diagramu). W przypadku postępowania się współrzędnymi względnymi (pikselowymi) należy wykonać odpowiednie przeliczenia używając funkcji *RelToAbsX* i *RelToAbsY*.

Składnia

SetPosition (*nazwa_obiektu*, *współrzędna_X*, *współrzędna_Y*)

SetPosition (*współrzędna_X*, *współrzędna_Y*)

Parametry

nazwa_obiektu

Nazwa elementu, którego położenie należy zmienić. Wariant akcji bez podania nazwy dotyczy zmiany położenia elementu kontekstowego (obiektu, wzorca lub grupy). W parametrze można używać znaków '*' i '?'.
 W przypadku użycia akcji w kontekście obiektu będącego częścią wzorca wyszukiwanie obiektów zgodnych z podaną nazwą dotyczy standardowo tylko obiektów tego wzorca. Odwołanie do obiektu nie będącego częścią wzorca wymaga dodania prefiksu '!' do nazwy obiektu.

współrzędna_X

Nowa wartość współrzędnej X kontrolowanego elementu w jednostkach bezwzględnych.

współrzędna_Y

Nowa wartość współrzędnej Y kontrolowanego elementu w jednostkach bezwzględnych.

Przykład

```
SetPosition( RelToAbsX(100+Variable(pos)), "" )
```

Modyfikowana jest tylko współrzędna X elementu diagramu. Zmienna *pos* przekazuje położenie we współrzędnych pikselowych. Po dodaniu offsetu położenia wykonywana jest konwersja na współrzędne bezwzględne.



Zobacz też

- Asix.Evo – Techniki budowy diagramów.PDF/CHM, *Animacja ruchu i zmiana rozmiaru obiektów*
- [akcja SetBounds](#), [akcja SetSize](#)
- AsixEvo_Wyrazenia_i_funkcje.PDF/CHM, funkcja *RelToAbsX*
- AsixEvo_Wyrazenia_i_funkcje.PDF/CHM, funkcja *RelToAbsY*

4.61 SetProperty

Przeznaczenie

Akcja pozwala na ustawienie dowolnej właściwości dla wskazanego obiektu lub diagramu. Pozwala to na sterowanie wyglądem/zachowaniem aplikacji.

Składnia

SetProperty (*nazwa_obiektu*, *nazwa_właściwości*, *wartość*)

SetProperty (*nazwa_właściwości*, *wartość*)

Parametry

nazwa_obiektu

Nazwa obiektu lub wzorca, którego właściwość należy zmienić. Jeżeli nazwa nie jest podana, to akcja dotyczy zmiany właściwości elementu kontekstowego (obiektu, wzorca, grupy lub diagramu). W parametrze można używać znaków '*' i '?'.

W przypadku użycia akcji w kontekście obiektu będącego częścią wzorca wyszukiwanie obiektów zgodnych z podaną nazwą dotyczy standardowo tylko obiektów tego wzorca. Odwołanie do obiektu nie będącego częścią wzorca wymaga dodania prefiksu '!' do nazwy obiektu.

nazwa_właściwości

Nazwa właściwości, którą należy zmienić. Należy używać nazw wewnętrznych właściwości. Najprostszą metodą wstawienia do treści akcji poprawnej nazwy właściwości jest skorzystanie z komendy *Właściwości lokalne* w menu edytora akcji.

wartość

Nowa wartość zmienianej właściwości.

Przykłady

```
 SetProperty(Temp,MainVar,blok1Temp)
```

Akcja zmienia główną zmienną obiektu o nazwie *Temp* na wartość *blok1Temp*. Skutkuje to przełączeniem obiektu na wyświetlanie stanu nowej zmiennej procesowej.

4.62 SetSize**Przeznaczenie**

Akcja pozwala na zmianę rozmiaru wybranego elementu diagramu. Służy do realizacji animacji rozmiaru obiektów na diagramie. Dodatkowe informacje i przykład użycia znajdują się w opisie akcji *SetPosition*.

Składnia

SetSize (*nazwa_obiektu*, *szerokość*, *wysokość*)

SetSize(*szerokość*, *wysokość*)

Parametry***nazwa_obiektu***

Nazwa elementu, którego rozmiar należy zmienić. Wariant akcji bez podania nazwy dotyczy zmiany rozmiaru elementu kontekstowego (obektu, wzorca lub grupy). W parametrze można używać znaków '*' i '?'.
 W przypadku użycia akcji w kontekście obiektu będącego częścią wzorca wyszukiwanie obiektów zgodnych z podaną nazwą dotyczy standardowo tylko obiektów tego wzorca. Odwołanie do obiektu nie będącego częścią wzorca wymaga dodania prefiksu '!' do nazwy obiektu.

szerokość

Nowa szerokość kontrolowanego elementu w jednostkach bezwzględnych.

wysokość

Nowa wysokość kontrolowanego elementu w jednostkach bezwzględnych.



Zobacz też

- Asix.Evo – Techniki budowy diagramów.PDF/CHM, *Animacja ruchu i zmiana rozmiaru obiektów*

- [akcja SetPosition](#), [akcja SetBounds](#)

4.63 SetVariable

Przeznaczenie

Akcja pozwala na bezpośrednie wysłanie do wskazanej zmiennej procesowej wartości sterującej.

Wykonanie akcji jest kontrolowane przez system uprawnień.

Składnia

SetVariable(*nazwa zmiennej, wartość*)

SetVariable(*nazwa zmiennej, wartość, maska_wartości*)

Parametry

nazwa_zmiennej

Nazwa zmiennej procesowej, do której należy wysłać sterowanie. Pusta nazwa oznacza użycie zmiennej sterowanej obiektu wynikającego z kontekstu użycia akcji. Możliwa jest notacja sufiksowa nazwy zmiennej względem zmiennej głównej obiektu wynikającego z kontekstu.

wartość

Wartość sterowania wysyłana do zmiennej. W wariacie akcji bez parametru *maska_wartości* podana wartość jest wysyłana bez żadnej modyfikacji. W drugim wariacie wartość jest modyfikowana zgodnie z algorytmem podanym w opisie parametru *maska_wartości*.

maska_wartości

Jeżeli parametr jest użyty, to w trakcie wykonywania akcji:

- odczytywana jest bieżąca wartość zmiennej;
- w odczytanej wartości zmieniane są tylko te bity, którym w *masce_wartości* odpowiadają bity o wartości 1;
- tak zmodyfikowana wartość wysyłana jest do zmiennej.

Przykłady

```
SetVariable("",1)
```

Zmienna sterowana obiektu, na konto którego wykonywana jest akcja, zostanie ustawiona na wartość 1.

```
SetVariable("#_S",8,0xc)
```

Ustawiona zostanie zmienna, której nazwa powstanie z połączenia nazwy głównej obiektu, na konto którego wykonywana jest akcja, oraz sufiksu „_S”. Bit 3 zmiennej zostanie ustawiony na 1 a bit 2 na 0. Pozostałe bity pozostaną niezmienione. Na przykład, jeżeli wartość oryginalna zmiennej była równa 0xff, to po wykonaniu sterowania będzie równa 0xfb.

**Zobacz też**

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, *Organizacja operacji sterujących*

4.64 SetWindowSize**Przeznaczenie**

Akcja służy do zmiany rozmiaru okien synoptycznych. Możliwa jest grupowa zmiana rozmiaru okien na podstawie zgodności ze wzorcem nazwy.

Akcja jest alternatywą do ręcznej zmiany rozmiaru okna przez użytkownika. Użytkownik może zmienić rozmiar okna, które nie ma założonej blokady i posiada ramkę. Należy pamiętać, że zmiana rozmiaru okna spowoduje zmianę rozmiarów paneli wchodzących w skład okna.

Może też nastąpić przeskalowanie diagramów do nowych rozmiarów.

Składnia

```
SetWindowSize(nazwa_okna, szerokość, wysokość)
```

Parametry***nazwa_okna***

Parametr określa nazwę okien, których akcja dotyczy. Można podać nazwę dokładną lub wzorzec nazwy zawierający znaki specjalne * i ?. Gdy nazwa okna jest pusta, to zmieniany jest rozmiar okna wynikającego z kontekstu użycia – okno z którego została wywołana akcja *SetWindowSize*.

Szerokość

Nowa szerokość okna.

Wysokość

Nowa wysokość okna.

Przykłady

```
SetWindowsSize (null,400,200)
```

Akcja zmienia rozmiar okna kontekstowego.



Zobacz też

- AsixEvo_Techniki_budowy_diagramów.PDF/CHM, 4.2. *Kontrola położenia i rozmiaru okien*

4.65 ShowControlPanel

Przeznaczenie

Akcja służy do otwarcia okna panelu kontrolnego. Panel kontrolny to predefiniowane okno pozwalające na wykonanie podstawowych czynności administracyjnych.

Okno panelu można w każdej aplikacji otworzyć także przy pomocy kombinacji klawiszy *Ctrl-Shift-F1*.

Składnia

```
ShowControlPanel()
```

4.66 ShowKeyboard

Przeznaczenie

Akcja służy do wyświetlenia okna alfanumerycznej klawiatury ekranowej.

Składnia

```
ShowKeyboard()
```



Zobacz też

- [Akcja ShowNumericKeyboard](#)

4.67 ShowMenu

Przeznaczenie

Akcja służy do wyświetlenia wskazanego menu kontekstowego. Menu jest otwierane w miejscu, w którym znajduje się kursor myszki. Alternatywą do menu kontekstowego są menu podłączone na stałe do okien synoptycznych.

Składnia

ShowMenu(*nazwa_menu*, *parametry*)

Parametry

nazwa_menu

Nazwa menu zdefiniowanego w aplikacji.

parametry

Parametr służy do przekazania do otwieranego menu wartości jego parametrów. Zestaw dozwolonych parametrów zależy od definicji menu. Parametry są podawane jako sekwencja par *nazwa_parametru=wartość_parametru* rozdzielonych znakiem średnika, np.

Kolor=Red;Zmienna=v1



Zobacz też

- [AsixEvo_Elementy_wizualizacji.PDF/CHM](#), 5. Menu

4.68 ShowNumericKeyboard

Przeznaczenie

Akcja służy do wyświetlenia okna numerycznej klawiatury ekranowej.

Składnia

ShowNumericKeyboard()



Zobacz też

[Akcja ShowKeyboard](#)

4.69 StartAlarm

Przeznaczenie

Akcja służy do zgłoszenia zdarzenia rozpoczęcia alarmu. Alarm będący obiektem akcji musi należeć do strategii zewnętrznej lub niezdefiniowanej. Alternatywną metodą zgłoszenia początku alarmu jest użycie w funkcji *Raise* interfejsu *Alarm* w kodzie skryptu użytkownika.

Składnia

StartAlarm(*nazwa_domeny*, *identyfikator_alarmu*)

StartAlarm(*nazwa_domeny*, *identyfikator_alarmu*, *parametr_1*,...)

Parametry

nazwa_domeny

Nazwa domeny alarmów, której dotyczy akcja. Jeżeli nazwa nie jest podana, to akcja dotyczy domeny domyślnej (pierwszej).

identyfikator_alarmu

Identyfikator alarmu, który ma zostać rozpoczęty.

parametr_1*, ..., *parametr_n

Dodatkowe parametry rejestrowane w logu alarmów.



Zobacz też

- Asix.Evo_System_alarmów.PDF/CHM
- Asix.Evo_System_alarmów.PDF/CHM, *Strategie wykrywania alarmów*
- AsixEvo_Skrypty.PDF/CHM, *Interfejs IAlarm*
- [akcja EndAlarm](#)

4.70 StopSound

Przeznaczenie

Akcja przerywa aktualnie odtwarzany sygnał dźwiękowy. W wersji wywołania z podanym parametrem nazwy domeny alarmowej, akcja powoduje wyciszenie sygnału dźwiękowego sygnalizacji alarmu na stanowisku wykonania akcji oraz na innych stanowiskach uczestniczących w pracy domeny.

Składnia

StopSound()
StopSound(*nazwa_domeny*)

Parametry

nazwa_domeny

Nazwa domeny alarmów, której dotyczy akcja. Można podać nazwę '*', co oznacza wyłączenie sygnału alarmowego dla wszystkich domen alarmów.



Zobacz też

- AsixEvo_System_alarmów.PDF/CHM, *Sygnalizacja dźwiękowa*
- [akcja PlaySound](#), [akcja PlaySoundLooping](#)

4.71 SwitchLanguage

Przeznaczenie

Akcja pozwala na przełączenie języka pracy. Przełączany jest aktualny język aplikacji i skojarzony z nim język programu. Aplikacja musi być skonfigurowana do pracy wielojęzycznej.

Język aplikacji można zmienić także poprzez użycie okna panelu kontrolnego. Okno panelu można w każdej aplikacji otworzyć przy pomocy kombinacji klawiszy *Ctrl-Shift-F1*.

Składnia

SwitchLanguage(*kod_języka*)

Parametry

kod_języka

Dwuliterowy kod języka.

Przykłady

```
SwitchLanguage("en")
```

Akcja przełącza aplikację na pracę w języku angielskim.



Zobacz też

- [AsixEvo_Parametryzacja_aplikacji.PDF/CHM](#), 4. *Parametryzacja aplikacji wielojęzycznych*

4.72 SwitchWorkMode

Przeznaczenie

Akcja pozwala na przejście z trybu wykonywania aplikacji do wybranego trybu edycyjnego.

Wykonanie akcji jest kontrolowane przez system uprawnień. Aby przejść do trybu edycji, trzeba się zalogować oraz posiadać uprawnienie do właściwego trybu edycyjnego.

Tryb pracy można zmienić także poprzez użycie okna panelu kontrolnego. Okno panelu można w każdej aplikacji otworzyć przy pomocy kombinacji klawiszy *Ctrl-Shift-F1*.

Składnia

SwitchWorkMode(*tryb_edycji*)

Parametry

tryb_edycji

Parametr określa żądany tryb edycji. Parametr musi być jedną z poniższych stałych:

\$Architect – pełny tryb edycyjny z otwarciem okna Architekta.

\$WindowEdit – uproszczony tryb edycji okien synoptycznych.

\$DiagramEdit – uproszczony tryb edycji diagramów synoptycznych.

Przykłady

SwitchWorkMode(\$DiagramEdit)

Akcja przełącza aplikację do trybu uproszczonej edycji diagramów synoptycznych.

**Zobacz też**

- AsixEvo_Jak_to_zrobić.PDF/CHM, *Uruchamianie programu i zmiana trybów pracy/Przejście z trybu wykonawczego do trybu edycyjnego*
- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, *5. System uprawnień*

4.73 SydelShow**Przeznaczenie**

Akcja służy do sterowania pracą programu SydelEko firmy Technicon. Akcja pozwala na przełączanie informacji wyświetlanych w oknie programu SydelEKO.

Składnia

SydelShow(*opis_węzła*)

Parametry

opis_węzła

Parametr służy do identyfikacji węzła, którego okno należy wyświetlić. Należy podać pełną nazwę opisową węzła lub jego numeryczny identyfikator.

4.74 Terminate

Przeznaczenie

Akcja służy do zakończenia wykonywania aplikacji. Działa tylko wtedy, gdy aplikacja została uruchomiona od początku w trybie wykonawczym.

Wykonanie akcji jest kontrolowane przez system uprawnień. Aby zakończyć pracę aplikacji, trzeba się zalogować oraz posiadać uprawnienie do zamknięcia aplikacji.

Możliwe są alternatywne metody kończenia pracy aplikacji:

- Zamknięcie wszystkich otwartych okien.
- Użycie przycisku końca pracy w oknie panelu kontrolnego. Okno panelu można w każdej aplikacji otworzyć przy pomocy kombinacji klawiszy *Ctrl-Shift-F1*.

Składnia

Terminate()



Zobacz też

- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 5. System uprawnień

4.75 ToggleBits

Przeznaczenie

Akcja pozwala na zmianę wartości wybranych bitów zmiennej procesowej na przeciwną.

Wykonanie akcji jest kontrolowane przez system uprawnień.

Składnia

ToggleBits(*nazwa zmiennej, maska_bitów*)

Parametry

nazwa_zmiennej

Nazwa zmiennej procesowej, do której należy wysłać sterowanie. Pusta nazwa oznacza użycie zmiennej sterowanej obiektu wynikającego z kontekstu użycia akcji. Możliwa jest

notacja sufiksowa nazwy zmiennej względem zmiennej głównej obiektu wynikającego z kontekstu.

maska_bitów

Parametr określa bity zmiennej, które ulegną zmianie. Zmieniane są tylko te bity, którym w masce_bitów odpowiadają bity o wartości 1.

Przykłady

ToggleBits ("",0x1)

Wartość najmłodszego bitu zmiennej sterowanej obiektu, na konto którego wykonywana jest akcja, zostanie zmieniona na przeciwną.



Zobacz też

- Asix.Evo_Techniki_budowy_diagramów.PDF/CHM, 6. *Organizacja operacji sterujących*
- Asix.Evo_Parametryzacja_aplikacji.PDF/CHM, 5. *System uprawnień*

4.76 VariableInfo

Przeznaczenie

Akcja pozwala na wyświetlenie okienka informacyjnego dotyczącego wskazanej zmiennej. Możliwe jest określenie nazw atrybutów, których wartości zostaną pokazane.

Składnia

VariableInfo(nazwa zmiennej, lista_atrybutów)

Parametry

nazwa_zmiennej

Nazwa zmiennej procesowej, dla której należy wyświetlić okienko informacyjne. Pusta nazwa oznacza użycie zmiennej głównej obiektu wynikającego z kontekstu użycia akcji. Możliwa jest notacja sufiksowa nazwy zmiennej względem zmiennej głównej obiektu wynikającego z kontekstu.

lista_atrybutów

Lista nazw atrybutów zmiennej rozdzielonych znakami przecinka. Parametr określa, które atrybuty zmiennej należy wyświetlić w okienku informacyjnym. Pusta lista atrybutów oznacza wyświetlenie wszystkich atrybutów.

Preferowaną metodą określenia listy atrybutów jest wykorzystanie właściwości globalnej typu *Zestaw atrybutów*. Pozwala to jednokrotnie zdefiniować listę, a następnie zastosować ją we wszystkich wywołaniach akcji *VariableInfo*, używając funkcji *Property*.

Przykłady

```
VariableInfo ("", Property(InfoAttrs))
```

Wyświetlone zostanie okno informacyjne dla kontekstowej zmiennej głównej obiektu. Nazwy wyświetlanych atrybutów zostaną pobrane z definicji właściwości globalnej *InfoAttrs*.

4.77 Zoom

Przeznaczenie

Akcja ustawia stopień skalowania diagramu na wybraną wartość procentową względem wielkości początkowej diagramu.

Składnia

Zoom (*nazwa_okna*, *nazwa_panelu*, *stopień_przeskalowania*)

Parametry

nazwa_okna

Nazwa okna, którego akcja dotyczy. Może być pusta, a wtedy okno będące obiektem akcji ustalane jest na podstawie kontekstu użycia.

nazwa_panelu

Nazwa panelu, w którym aktualnie otwarty diagram zostanie przeskalowany. Może być pusta, a wtedy panel będący obiektem akcji ustalany jest na podstawie kontekstu użycia.

Stopień_przeskalowania

Bezwzględna procentowa wartość przeskalowania diagramu w stosunku do jego wielkości początkowej. Wartość 100 oznacza powrót do początkowej wielkości diagramu.

Przykłady

```
Zoom ( null, schemat, 150)
```

Diagram znajdujący się w panelu *Schemat* bieżącego okna zostanie przeskalowany do 150% jest wielkości początkowej.



Zobacz też

- akcja [ZoomIn](#), akcja [ZoomOut](#)

4.78 ZoomIn

Przeznaczenie

Akcja powoduje powiększenie skalowanego diagramu o zadaną wartość procentową względem aktualnego stanu.

Składnia

ZoomIn (nazwa_okna, nazwa_panelu, stopień_przeskalowania)

Parametry

nazwa_okna

Nazwa okna, którego akcja dotyczy. Może być pusta, a wtedy okno będące obiektem akcji ustalane jest na podstawie kontekstu użycia.

nazwa_panelu

Nazwa panelu, w którym aktualnie otwarty diagram zostanie przeskalowany. Może być pusta, a wtedy panel będący obiektem akcji ustalany jest na podstawie kontekstu użycia.

stopień_przeskalowania

Procentowa wartość powiększenia diagramu w stosunku do jego wielkości aktualnej, np. wartość 10 oznacza powiększenie diagramu o 10%.



Zobacz też

- akcja [Zoom](#), akcja [ZoomOut](#)

4.79 ZoomOut

Przeznaczenie

Akcja powoduje pomniejszenie skalowanego diagramu o zadaną wartość procentową względem aktualnego stanu.

Składnia

ZoomOut (*nazwa_okna, nazwa_panelu, stopień_przeskalowania*)

Parametry

nazwa_okna

Nazwa okna, którego akcja dotyczy. Może być pusta, a wtedy okno będące obiektem akcji ustalane jest na podstawie kontekstu użycia.

nazwa_panelu

Nazwa panelu, w którym aktualnie otwarty diagram zostanie przeskalowany. Może być pusta, a wtedy panel będący obiektem akcji ustalany jest na podstawie kontekstu użycia.

stopień_przeskalowania

Procentowa wartość pomniejszenia diagramu w stosunku do jego wielkości aktualnej, np. wartość 10 oznacza pomniejszenie diagramu o 10%.



Zobacz też

- akcja [Zoom](#), akcja [ZoomIn](#)